

Distributed projected–reflected–gradient algorithms for stochastic generalized Nash equilibrium problems

Original

Distributed projected–reflected–gradient algorithms for stochastic generalized Nash equilibrium problems / Franci, B., Grammatico, S.. - (2021), pp. 369-374. (2021 European Control Conference, ECC 2021 Delft (Netherlands) 29 June 2021 - 02 July 2021) [10.23919/ecc54610.2021.9655217].

Availability:

This version is available at: 11583/3003651 since: 2025-10-06T07:06:36Z

Publisher:

Institute of Electrical and Electronics Engineers - IEEE

Published

DOI:10.23919/ecc54610.2021.9655217

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Distributed projected–reflected–gradient algorithms for stochastic generalized Nash equilibrium problems

Barbara Franci and Sergio Grammatico

Abstract—We consider the stochastic generalized Nash equilibrium problem (SGNEP) with joint feasibility constraints and expected–value cost functions. We propose a distributed stochastic projected reflected gradient algorithm and show its almost sure convergence when the pseudogradient mapping is monotone and the solution is unique. The algorithm is based on monotone operator splitting methods tailored for SGNEPs when the expected–value pseudogradient mapping is approximated at each iteration via an increasing number of samples of the random variable. Finally, we show that a preconditioned variant of our proposed algorithm has convergence guarantees when the pseudogradient mapping is cocoercive.

I. INTRODUCTION

Stochastic generalized Nash equilibrium problems (SGNEPs) have received some attention from the system and control community [1], [2], [3]. In a SGNEP, a set of agents interacts with the aim of minimizing their expected–value cost functions while subject to some joint feasibility constraints. Moreover, both the cost function and the constraints depend on the strategies of the other agents. SGNEPs arise when there is some uncertainty, modeled via a random variable with an unknown distribution. The interest on the topic is related to their possible applications in networked Cournot game with market capacity constraints and uncertain demand [4] as transportation systems, where the drivers perception of travel time is uncertain [5], and electricity markets where companies schedule their production without knowing the demand [6].

When the random variable is known, the expected–value formulation can be solved via a standard technique for deterministic GNEPs [7], [8]. In fact, one possible approach is to recast the problem as a stochastic variational inequality (SVI) through the use of the Karush–Kuhn–Tucker conditions. Then, the problem can be rewritten as a monotone inclusion and solved via operator splitting techniques. The difficulty in the stochastic case is that the pseudogradient is usually not directly accessible, for instance because the expected value is hard to compute. For this reason, in many situations, the search for a solution of a SVI relies on samples of the random variable. One of the most used approximation schemes in the literature is the stochastic approximation (SA). In this case, the approximation is done by using only one (or a finite number of) realization of the random variable. SA was first presented in [9], it is not computationally expensive

even if sometimes it requires strong assumptions on the pseudogradient mapping and on the parameters [1], [10].

Depending on the monotonicity assumptions on the pseudogradient mapping or the affordable computational complexity, there are different algorithms that can be used to find a SGNE. Among others, one can consider the stochastic preconditioned forward–backward (SpFB) algorithm [3] which is guaranteed to converge under cocoercivity of the pseudogradient and by demanding one projection step per iteration. However, cocoercivity is not the weakest possible assumption, hence one would like to have an algorithm that converges under mere monotonicity. Under this assumption, the stochastic forward–backward–forward (SFBF) algorithm involves only one projection step per iteration but two costly evaluation of the pseudogradient mapping [11]. Another alternative is the stochastic extragradient (SEG) algorithm whose iterates are characterized by two projection steps and two evaluation of the pseudogradient mapping [12], [13]. To summarize, having weaker assumptions comes at the price of implementing computationally expensive algorithms.

In this paper, we propose two special instances of the stochastic projected reflected gradient (SPRG) algorithm for SGNEPs. The SPRG involves only one projection step and one evaluation of the pseudogradient, computed in the reflection of the second–last iterate on the last one, i.e., it uses the previous two iterates for the updates. Specifically, our contributions are the following.

- We exploit two splitting techniques to obtain two distributed algorithms, with and without preconditioning, that are instances of the SPRG scheme.
- We prove convergence of the first algorithm to a SGNE if the pseudogradient mapping is monotone and the solution is unique (Section V), differently from [13] where it is assumed to be monotone and “weak sharp”.
- The weak sharpness property, discussed in Section IV, is hard to check on the problem data but it is guaranteed to hold on a cocoercive operator. Therefore, we prove convergence of the second, preconditioned algorithm when the pseudogradient mapping is cocoercive (Section VI).

We note that PRG was first presented for deterministic VI in [14] and for SVI in [10], [13]. Nevertheless, this is the first time that it is designed for SGNEPs, i.e. a non–trivial instance of SVI with coupling constraints between the agents. Remarkably, under uniqueness of the solution, our algorithm has convergence guarantees also for merely monotone pseudogradient mappings. This is a significant advantage compared to the SpFB which may not converge

The authors are with the Delft Center for System and Control, TU Delft, The Netherlands {b.franci-1, s.grammatico}@tudelft.nl

This work was partially supported by NWO under research projects OMEGA (613.001.702) and P2P-TALES (647.003.003), and by the ERC under research project COSMOS (802348).

in that case.

Notation and preliminaries: We use the same notation as [3], [15], [16] and we recall that for all $x, y \in C$, a mapping $F : C \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ is [15], [16]: *pseudomonotone* if $(x - y)^T F(y) \geq 0 \Rightarrow (x - y)^T F(x) \geq 0$; *monotone* if $(F(x) - F(y))^T (x - y) \geq 0$; *maximally monotone* if there exists no monotone operator $G : C \rightarrow \mathbb{R}^n$ such that the graph of G properly contains the graph of F ; β -*cocoercive* if for $\beta > 0$ $(F(x) - F(y))^T (x - y) \geq \beta \|F(x) - F(y)\|^2$; *L-Lipschitz continuous* if for $L > 0$ $\|F(x) - F(y)\| \leq L \|x - y\|$.

II. STOCHASTIC GENERALIZED NASH EQUILIBRIUM PROBLEM

We consider a set of agents $\mathcal{I} = \{1, \dots, N\}$, each of them choosing its decision variable $x_i \in \mathbb{R}^{n_i}$ from its local decision set $\Omega_i \subseteq \mathbb{R}^{n_i}$. The aim of each agent is to minimize its local cost function $\mathbb{J}_i : \mathbb{R}^n \rightarrow \mathbb{R}$ defined as

$$\mathbb{J}_i(x_i, \mathbf{x}_{-i}) := \mathbb{E}_\xi [J_i(x_i, \mathbf{x}_{-i}, \xi(\omega))]$$

for some measurable function $J_i : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ and $n = \sum_{i=1}^N n_i$. We note that the cost function depends on the local variable x_i , on the decisions of the other agents $\mathbf{x}_{-i} = \text{col}((x_j)_{j \neq i})$ and on the random variable $\xi : \Xi \rightarrow \mathbb{R}^d$ which expresses the uncertainty in the cost function. Given the associated probability space $(\Xi, \mathcal{F}, \mathbb{P})$, \mathbb{E}_ξ represents the mathematical expectation with respect to the distribution of ξ . We assume that $\mathbb{E}[J_i(\mathbf{x}, \xi)]$ is well defined for all the feasible $\mathbf{x} = \text{col}(x_1, \dots, x_N)$.

The cost function should satisfy some assumptions, postulated next, to make our analysis possible. Such assumptions are standard for (stochastic) Nash equilibrium problems [17].

Assumption 1: For each $i \in \mathcal{I}$ and $\mathbf{x}_{-i} \in \mathcal{X}_{-i}$ $\mathbb{J}_i(\cdot, \mathbf{x}_{-i})$ is convex and continuously differentiable. For each $i \in \mathcal{I}$ and $\xi \in \Xi$, $J_i(\cdot, \mathbf{x}_{-i}, \xi)$ is convex, Lipschitz continuous and continuously differentiable. For each \mathbf{x}_{-i} , $J_i(x_i, \mathbf{x}_{-i}, \cdot)$ is measurable and Lipschitz continuous with constant $\ell_i(\mathbf{x}_{-i}, \xi)$, integrable in ξ . ■

Since we consider a SGNEP, we introduce affine shared constraints of the form $A\mathbf{x} \leq b$, where $A = [A_1, \dots, A_N] \in \mathbb{R}^{m \times n}$, $A_i \in \mathbb{R}^{m \times n_i}$ indicates how agent i is involved in the coupling constraints and $b \in \mathbb{R}^m$. Thus, we denote each agent $i \in \mathcal{I}$ feasible decision set with the set-valued mapping

$$\mathcal{X}_i(\mathbf{x}_{-i}) := \{y_i \in \Omega_i \mid A_i y_i \leq b - \sum_{j \neq i}^N A_j x_j\}, \quad (1)$$

and the collective feasible set as

$$\mathcal{X} = \Omega \cap \{\mathbf{y} \in \mathbb{R}^n \mid A\mathbf{y} - b \leq \mathbf{0}_m\} \quad (2)$$

where $\Omega = \prod_{i=1}^N \Omega_i$. We suppose that there is no uncertainty in the constraints. Standard assumptions for the constraints sets are postulated next [18].

Assumption 2: For each $i \in \mathcal{I}$, Ω_i is nonempty, compact and convex. \mathcal{X} satisfies Slater's constraint qualification. ■

The aim of each agent i , given the decision variables of the other agents \mathbf{x}_{-i} , is to choose a decision x_i , that solves its local optimization problem, i.e.,

$$\forall i \in \mathcal{I} : \begin{cases} \min_{x_i \in \Omega_i} & \mathbb{J}_i(x_i, \mathbf{x}_{-i}) \\ \text{s.t.} & A_i x_i \leq b - \sum_{j \neq i}^N A_j x_j. \end{cases} \quad (3)$$

¹For brevity, we use ξ instead of $\xi(\omega)$, $\omega \in \Xi$, and \mathbb{E} instead of \mathbb{E}_ξ .

From a game-theoretic perspective, we aim at computing a stochastic generalized Nash equilibrium (SGNE) [17].

Definition 1: A collective variable $\mathbf{x}^* \in \mathcal{X}$ is a stochastic generalized Nash equilibrium if, for all $i \in \mathcal{I}$:

$$\mathbb{J}_i(x_i^*, \mathbf{x}_{-i}^*) \leq \inf\{\mathbb{J}_i(y, \mathbf{x}_{-i}^*) \mid y \in \mathcal{X}_i(\mathbf{x}_{-i}^*)\}.$$

When Assumptions 1-2 hold, existence of a SGNE of the game in (3) is guaranteed by [17, §3.1].

Among all the possible Nash equilibria, we focus on those that are also solutions of an associated (stochastic) variational inequality. First let us define the pseudogradient mapping as

$$\mathbb{F}(\mathbf{x}) = \text{col}(\mathbb{E}[\nabla_{x_i} J_i(x_i, \mathbf{x}_{-i}, \xi_i)]_{i \in \mathcal{I}}), \quad (4)$$

where the exchange between the expected value and the gradient is possible because of Assumption 1 [17, Lem. 3.4]. Then, the stochastic variational inequality SVI(\mathcal{X}, \mathbb{F}) is the problem of finding $\mathbf{x}^* \in \mathcal{X}$ such that

$$\langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \quad \text{for all } \mathbf{x} \in \mathcal{X}. \quad (5)$$

with $\mathbb{F}(\mathbf{x})$ as in (4) and \mathcal{X} as in (1). We also note that any solution of SVI(\mathcal{X}, \mathbb{F}) is a SGNE of the game in (3) while the opposite does not hold in general.

Remark 1: Under Assumptions 1-2, the solution set of SVI(\mathcal{X}, \mathbb{F}) is non empty and compact, i.e. $\text{SOL}(\mathcal{X}, \mathbb{F}) \neq \emptyset$ [16, Corollary 2.2.5]. ■

A. Operator-theoretic characterization

In this section, we rewrite the SGNEP as a monotone inclusion, i.e., the problem of finding a zero of a set-valued monotone operator. To this aim, we characterize the SGNE of the game in terms of the Karush–Kuhn–Tucker (KKT) conditions for the coupled optimization problems in (3). Let us denote with $\lambda_i \in \mathbb{R}_{\geq 0}^m$ the dual variable associated with the coupling constraints. Then, the associated KKT optimality conditions reads as [16]

$$\forall i \in \mathcal{I} : \begin{cases} 0 \in \mathbb{E}[\nabla_{x_i} J_i(x_i^*, \mathbf{x}_{-i}^*, \xi)] + N_{\Omega_i}(x_i^*) + A_i^\top \lambda_i \\ 0 \in -(A\mathbf{x}^* - b) + N_{\mathbb{R}_{\geq 0}^m}(\lambda^*). \end{cases} \quad (6)$$

From [18, Th. 3.1], it follows that \mathbf{x}^* is a solution of the SGNEP if the KKT conditions hold with $\lambda_1 = \lambda_2 = \dots = \lambda_N = \lambda^*$, i.e., \mathbf{x}^* is a solution of SVI(\mathcal{X}, \mathbb{F}) in (5). In words, [18, Th. 3.1] says that variational equilibria are those such that the shared constraints have the same dual variable λ^* for all the agents. We conclude this section writing the KKT conditions as

$$0 \in \mathcal{T}(\mathbf{x}, \boldsymbol{\lambda}) := \begin{bmatrix} N_\Omega(\mathbf{x}) + \mathbb{F}(\mathbf{x}) + A^\top \boldsymbol{\lambda} \\ N_{\mathbb{R}_{\geq 0}^m}(\boldsymbol{\lambda}) - (A\mathbf{x} - b) \end{bmatrix}, \quad (7)$$

where $\mathcal{T} : \mathcal{X} \times \mathbb{R}_{\geq 0}^m \Rightarrow \mathbb{R}^n \times \mathbb{R}^m$ is a set-valued mapping. This notation is helpful to obtain the algorithms and for the convergence analysis.

B. Approximation scheme

Since the distribution of the random variable is unknown, we replace the expected value with an approximation. For the convergence analysis, we use the stochastic approximation (SA) scheme. We assume to have access to a pool of i.i.d. samples of the random variable collected, for all

Algorithm 1 Distributed stochastic projected reflected gradient (SPRG)

Initialization: $x_i^0 \in \Omega_i, \lambda_i^0 \in \mathbb{R}_{\geq 0}^m$, and $z_i^0 \in \mathbb{R}^m$.

Iteration k : Agent i

(1) Updates

$$\begin{aligned}\tilde{x}_i^k &= 2x_i^k - x_i^{k-1} \\ \tilde{z}_i^k &= 2z_i^k - z_i^{k-1} \\ \tilde{\lambda}_i^k &= 2\lambda_i^k - \lambda_i^{k-1}\end{aligned}$$

(2) Receives $x_j^k, j \in \mathcal{N}_i^J$ and $z_j^k, \lambda_j^k, j \in \mathcal{N}_i^\lambda$, then updates:

$$\begin{aligned}x_i^{k+1} &= \text{proj}_{\Omega_i} [x_i^k - \alpha_i (F_i^{\text{SA}}(\tilde{x}_i^k, \tilde{\mathbf{x}}_{-i}^k, \xi_i^k) + A_i^\top \tilde{\lambda}_i^k)] \\ z_i^{k+1} &= z_i^k - \nu_i \sum_{j \in \mathcal{N}_i^\lambda} w_{i,j} (\tilde{\lambda}_j^k - \tilde{\lambda}_i^k) \\ \lambda_i^{k+1} &= \text{proj}_{\mathbb{R}_{\geq 0}^m} \{ \lambda_i^k + \tau_i (A_i \tilde{x}_i^k - b_i) \\ &\quad - \tau \sum_{j \in \mathcal{N}_i^\lambda} w_{i,j} [(z_i^k - z_j^k) - (\tilde{\lambda}_i^k - \tilde{\lambda}_j^k)] \}\end{aligned}$$

$k \in \mathbb{N}$ and for each agent $i \in \mathcal{I}$, in the vectors $\bar{\xi}_i^k = \text{col}(\xi_i^{(1)}, \dots, \xi_i^{(N_k)})$. At each time k , the approximation is

$$F_i^{\text{SA}}(x_i^k, \mathbf{x}_{-i}^k, \bar{\xi}_i^k) = \frac{1}{N_k} \sum_{s=1}^{N_k} \nabla_{x_i} J_i(x_i^k, \mathbf{x}_{-i}^k, \xi_i^{(s)}) \quad (8)$$

where N_k is the batch size, i.e., the number of sample to be taken. We define the distance between the expected value and its approximation as

$$\epsilon^k = F^{\text{SA}}(\mathbf{x}^k, \boldsymbol{\xi}^k) - \mathbb{F}(\mathbf{x}^k),$$

where $F^{\text{SA}}(\mathbf{x}, \boldsymbol{\xi}) = \text{col}(F_i^{\text{SA}}(\mathbf{x}, \bar{\xi}_i))$ and $\boldsymbol{\xi} = \text{col}(\bar{\xi}_1, \dots, \bar{\xi}_N)$. From now on, the superscript SA indicates that we use F^{SA} as in (8). Let us introduce the filtration $\mathcal{F} = \{\mathcal{F}_k\}$, i.e., a family of σ -algebras such that $\mathcal{F}_0 = \sigma(X_0)$, for all $k \geq 1$, $\mathcal{F}_k = \sigma(X_0, \xi_1, \xi_2, \dots, \xi_k)$ and $\mathcal{F}_k \subseteq \mathcal{F}_{k+1}$ for all $k \geq 0$. The filtration \mathcal{F} collects the informations that each agent has at the beginning of each iteration k . We note that the process ϵ_k is adapted to \mathcal{F}_k and it satisfies the following assumption [11], [12].

Assumption 3: For all $k \geq 0$, $\mathbb{E}[\epsilon_k | \mathcal{F}_k] = 0$ a.s. ■

Moreover, the stochastic error has a vanishing second moment that depends on the increasing number of samples N_k taken at each iteration.

Assumption 4: There exist $c, k_0, a > 0$ such that, for all $k \in \mathbb{N}$, $N_k \geq c(k + k_0)^{a+1}$. For all k and $C > 0$, the stochastic error is such that

$$\mathbb{E}[\|\epsilon^k\|^2 | \mathcal{F}_k] \leq \frac{C\sigma^2}{N_k} \text{ a.s.} \quad (9)$$

The bound for the stochastic error in (9) can be obtained as a consequence of some milder assumptions and it is called variance reduction; we refer to [12, Lem. 3.12], [3, Lem. 6] for more details. Concerning the batch size, Assumption 4 is standard in the SA literature [12, Eq. 11], [13, Eq. v-SPRG] since it implies that the sequence $(1/N_k)_{k \in \mathbb{N}}$ is summable.

III. DISTRIBUTED STOCHASTIC PROJECTED REFLECTED GRADIENT ALGORITHMS

In this section, we propose two distributed instances of a stochastic projected reflected gradient (SPRG) algorithm for finding a v-SGNE of the game in (3). The iterations are

Algorithm 2 Distributed stochastic preconditioned projected reflected gradient (SpPRG)

Initialization: $x_i^0 \in \Omega_i, \lambda_i^0 \in \mathbb{R}_{\geq 0}^m$, and $z_i^0 \in \mathbb{R}^m$.

Iteration k : Agent i

(1) Updates

$$\begin{aligned}\tilde{x}_i^k &= 2x_i^k - x_i^{k-1} \\ \tilde{\lambda}_i^k &= 2\lambda_i^k - \lambda_i^{k-1}\end{aligned}$$

(2) Receives $\tilde{x}_j^k, j \in \mathcal{N}_i^J$ and $\lambda_j^k, j \in \mathcal{N}_i^\lambda$ then updates:

$$\begin{aligned}x_i^{k+1} &= \text{proj}_{\Omega_i} [x_i^k - \alpha_i (F_i^{\text{SA}}(\tilde{x}_i^k, \tilde{\mathbf{x}}_{-i}^k, \xi_i^k) - A_i^\top \lambda_i^k)] \\ z_i^{k+1} &= z_i^k + \nu_i \sum_{j \in \mathcal{N}_i^\lambda} w_{i,j} (\lambda_i^k - \lambda_j^k)\end{aligned}$$

(3) Receives $\tilde{x}_j^{k+1}, j \in \mathcal{N}_i^J, z_j^{k+1}, \tilde{\lambda}_j^k, j \in \mathcal{N}_i^\lambda$ then updates:

$$\begin{aligned}\lambda_i^{k+1} &= \text{proj}_{\mathbb{R}_{\geq 0}^m} [\lambda_i^k + \sigma_i (A_i (2x_i^{k+1} - x_i^k) - b_i) \\ &\quad + \sigma_i \sum_{j \in \mathcal{N}_i^\lambda} w_{i,j} (2(z_i^{k+1} - z_j^{k+1}) - (z_i^k - z_j^k)) \\ &\quad - \sigma_i \sum_{j \in \mathcal{N}_i^\lambda} w_{i,j} (\tilde{\lambda}_i^k - \tilde{\lambda}_j^k)]\end{aligned}$$

presented in Algorithms 1 and 2 and are inspired by [13], [14]. We note that in Algorithm 1 only one communication round is needed for the update of the local variables x_i^k, z_i^k and λ_i^k at each iteration k , while the preconditioned version requires an extra exchange of information before the update of the dual variable. However, in Algorithm 2 the reflection step is not done for the auxiliary variable. Since we want the algorithm to be distributed, we assume that each agent i only knows its local data Ω_i, A_i and b_i . Moreover, each player is able to compute the approximation $F^{\text{SA}}(\mathbf{x}, \boldsymbol{\xi})$ in (8) of $\mathbb{F}(\mathbf{x})$ in (4), given the collective decision \mathbf{x} . We assume that each agent has access to all the decision variables that affect its pseudogradient (full decision information setup). These information are collected in the sets $\mathcal{N}_i^J, i \in \mathcal{I}$, i.e., the set of agents j whose decision x_j explicitly influences J_i . Since the v-SGNE requires consensus of the dual variables, we introduce an auxiliary variable $z_i \in \mathbb{R}^m, i \in \mathcal{I}$. The role of $\mathbf{z} = \text{col}(z_1, \dots, z_N)$ is to help reaching consensus and, along with a local copy of the dual variable λ_i , it is shared through the graph $\mathcal{G}^\lambda = (\mathcal{I}, \mathcal{E}^\lambda)$. The set of edges \mathcal{E}^λ represents the exchange of the private information on the dual variables: $(i, j) \in \mathcal{E}^\lambda$ if agent i can receive $\{\lambda_j, z_j\}$ from agent j . The set of neighbors of i in \mathcal{G}^λ is indicated with $\mathcal{N}_i^\lambda = \{j | (j, i) \in \mathcal{E}^\lambda\}$ [7], [2]. Since each agent feasible set implicitly depends on all the other agents decisions, to reach consensus of the dual variables, all agents must coordinate and therefore, \mathcal{G}^λ must be connected.

Assumption 5: The dual-variable communication graph \mathcal{G}^λ is undirected and connected. ■

The weighted adjacency matrix of the dual variables graph is indicated with $W \in \mathbb{R}^{N \times N}$. Let $L = D - W \in \mathbb{R}^{N \times N}$ be the Laplacian matrix associated to W , where $D = \text{diag}(d_1, \dots, d_N)$ is the diagonal matrix of the agents degrees $d_i = \sum_{j=1}^N w_{i,j}$. It follows from Assumption 5 that the adjacency matrix W and the Laplacian L are both symmetric, i.e., $W = W^\top$ and $L = L^\top$.

To obtain the distributed iterations presented in Algorithm 1 and 2, we exploit a splitting technique starting from the operator \mathcal{T} in (7). First, let us note that it can be written as $\mathcal{T} = \mathcal{Q} + \mathcal{P} + \mathcal{R}$ where

$$\begin{aligned}\mathcal{Q} : \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbb{F}(\mathbf{x}) \\ \mathbf{b} \end{bmatrix}, \\ \mathcal{R} : \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} 0 & \mathbf{A}^\top \\ -\mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix}, \\ \mathcal{P} : \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbf{N}_\Omega(\mathbf{x}) \\ \mathbf{N}_{\mathbb{R}_{\geq 0}^m}(\boldsymbol{\lambda}) \end{bmatrix},\end{aligned}\quad (10)$$

and $\boldsymbol{\lambda} = \text{col}(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{Nm}$. Moreover, let $\mathbf{L} = L \otimes \mathbf{I}_m \in \mathbb{R}^{Nm \times Nm}$, $\mathbf{A} = \text{diag}\{A_1, \dots, A_N\} \in \mathbb{R}^{Nm \times n}$ and let us define the column vector \mathbf{b} of suitable dimensions.

Following [7], to force consensus on the dual variables, we impose the Laplacian constraint $\mathbf{L}\boldsymbol{\lambda} = 0$. Then, to preserve monotonicity, we augment the operators introducing the auxiliary variable \mathbf{z} . Exploiting the splitting $\mathcal{T} = (\mathcal{Q} + \mathcal{R}) + \mathcal{P}$, we consider the following extended operators:

$$\begin{aligned}\mathcal{A} : \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbb{F}(\mathbf{x}) \\ 0 \\ \mathbf{L}\boldsymbol{\lambda} + \mathbf{b} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \mathbf{A}^\top \\ 0 & 0 & \mathbf{L} \\ -\mathbf{A} & -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} \\ \mathcal{B} : \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbf{N}_\Omega(\mathbf{x}) \\ 0 \\ \mathbf{N}_{\mathbb{R}_{\geq 0}^m}(\boldsymbol{\lambda}) \end{bmatrix}.\end{aligned}\quad (11)$$

Since the distribution of the random variable is unknown, we replace $\mathbb{F}(\mathbf{x})$ in (4) with F^{SA} in (8) and \mathcal{A} with $\hat{\mathcal{A}} = \mathcal{A}^{\text{SA}}$. Then, given $\boldsymbol{\omega} = \text{col}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda})$, Algorithm 1 in compact form reads as the SPRG iteration

$$\boldsymbol{\omega}^{k+1} = (\text{Id} + \Phi^{-1}\hat{\mathcal{B}})^{-1}(\boldsymbol{\omega}^k - \Phi^{-1}\hat{\mathcal{A}}(2\boldsymbol{\omega}^k - \boldsymbol{\omega}^{k-1})), \quad (12)$$

where $\Phi \succ 0$ contains the inverse of step size sequences, i.e., $\Phi = \text{diag}(\alpha^{-1}, \nu^{-1}, \sigma^{-1})$, and α^{-1} , ν^{-1} , σ^{-1} are diagonal matrices.

Another possible splitting of the operator \mathcal{T} in (7) can be considered, namely, $\mathcal{T} = \mathcal{Q} + (\mathcal{P} + \mathcal{R})$. Therefore, we can write a different couple of extended operators as

$$\begin{aligned}\mathcal{C} : \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbb{F}(\mathbf{x}) \\ 0 \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{L}\boldsymbol{\lambda} \end{bmatrix} \\ \mathcal{D} : \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} &\mapsto \begin{bmatrix} \mathbf{N}_\Omega(\mathbf{x}) \\ \mathbf{0}_{mN} \\ \mathbf{N}_{\mathbb{R}_{\geq 0}^m}(\boldsymbol{\lambda}) \end{bmatrix} + \begin{bmatrix} 0 & 0 & \mathbf{A}^\top \\ 0 & 0 & \mathbf{L} \\ -\mathbf{A} & -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix}.\end{aligned}\quad (13)$$

Analogously to $\hat{\mathcal{A}}$, we replace \mathbb{F} in (4) with the approximation F^{SA} in (8) and \mathcal{C} with $\hat{\mathcal{C}} = \mathcal{C}^{\text{SA}}$. Then, we can write Algorithm 2 in compact form as

$$\boldsymbol{\omega}^{k+1} = (\text{Id} + \Psi^{-1}\hat{\mathcal{D}})^{-1}(\boldsymbol{\omega}^k - \Psi^{-1}\hat{\mathcal{C}}(2\boldsymbol{\omega}^k - \boldsymbol{\omega}^{k-1})) \quad (14)$$

where Ψ is the preconditioning matrix. Specifically, let $\alpha^{-1} = \text{diag}\{\alpha_1^{-1}\mathbf{I}_{n_1}, \dots, \alpha_N^{-1}\mathbf{I}_{n_N}\} \in \mathbb{R}^{n \times n}$ and similarly σ^{-1} and ν^{-1} of suitable dimensions. Then, we have

$$\Psi = \begin{bmatrix} \alpha^{-1} & 0 & -\mathbf{A}^\top \\ 0 & \nu^{-1} & -\mathbf{L} \\ -\mathbf{A} & -\mathbf{L} & \sigma^{-1} \end{bmatrix}. \quad (15)$$

By expanding (14) with $\hat{\mathcal{C}}$ and \mathcal{D} as in (13) and Ψ as in (15), we obtain the iterations in Algorithm 2.

We note that in general the extended operators in (11) and (13) have different monotonicity properties. Which one specifically is discussed for the convergence analysis (Section V and VI, respectively) that follows from the considerations in the next section.

IV. TECHNICAL DISCUSSION ON WEAK SHARPNESS AND COCOERCIVITY

The original proof of the SPRG presented in [13] for SVI shows convergence under the assumption of monotonicity and weak sharpness. The weak sharpness property was first introduced to characterize the minima of

$$\min_{x \in \mathcal{X}} f(x) \quad (16)$$

with $f : \mathcal{X} \rightarrow \bar{\mathbb{R}}$ [19]. It was presented as an extension of the concept of strong (or sharp) solution, i.e., for all $x^* \in \mathcal{X}^* = \text{SOL}(f, \mathcal{X})$ $f(x) \geq f(x^*) + \rho \|x - x^*\|$, which holds if there is only one minimum. For generalizing non-unique solutions, the following definition was proposed in [19]: a set \mathcal{X}^* is a set of weak sharp minima for the function f if, for all $x \in \mathcal{X}$ and $x^* \in \mathcal{X}^*$,

$$f(x) \geq f(x^*) + \rho \text{dist}(x, \mathcal{X}^*) \quad (17)$$

where $\text{dist}(x, \mathcal{X}^*) = \inf_{x^* \in \mathcal{X}^*} \|x - x^*\|$. We note that a strong solution is also a weak sharp minimum while the contrary holds only if the solution is unique [19].

The concept was later extended to variational inequalities in [20], using the formal definition

$$-\mathbb{F}(\mathbf{x}^*) \in \text{int} \left(\bigcap_{\mathbf{x} \in \mathcal{X}^*} [\mathbb{T}\boldsymbol{\mathcal{X}}(\mathbf{x}) \cap \mathbf{N}_{\boldsymbol{\mathcal{X}^*}}(\mathbf{x})]^\circ \right), \quad (18)$$

which was already proved to be equivalent to (17) for the problem in (16) when $\mathbb{F}(\mathbf{x}^*) = \nabla f(\mathbf{x}^*)$. Unfortunately, the characterization in (18) is hard to use in a convergence proof. Hence, more practical conditions have been proposed. The first one [20] relies on the gap function G and it reads as

$$G(x) = \max_{y \in \mathcal{X}} \langle \mathbb{F}(y), x - y \rangle \geq \rho \text{dist}(x, \mathcal{X}^*). \quad (19)$$

Another condition, used in the convergence proof of the SPRG [13], was proposed in [21]:

$$\langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq \rho \text{dist}(\mathbf{x}, \boldsymbol{\mathcal{X}}^*), \quad (20)$$

for all $\mathbf{x}^* \in \boldsymbol{\mathcal{X}}^*$ and $\mathbf{x} \in \boldsymbol{\mathcal{X}}$. For the weak sharpness definition in (18) to be equivalent to (19) and (20), the (pseudo)gradient mapping should have the F -unique property, i.e., $\mathbb{F}(\text{SOL}(\mathbb{F}, \boldsymbol{\mathcal{X}}))$ should be at most a singleton [16, Section 2.3.1]. The class of operators that certainly have this property is that of monotone^+ operators, namely, a monotone mapping F such that for all $\mathbf{x}, \mathbf{y} \in \boldsymbol{\mathcal{X}}$ $\langle F(\mathbf{y}) - F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle = 0 \Rightarrow F(\mathbf{y}) = F(\mathbf{x})$. If a mapping is monotone^+ , then (18) is equivalent to (19) and (20) [20, Thm. 4.1]. The monotone^+ property does not necessarily hold for the extended operator \mathcal{C} in (13), even if it holds for \mathbb{F} . However, it holds if the operator is cocoercive [16, Def. 2.3.9]. We conclude this section with some examples showing that the condition in (20) may hold also if the mapping is not monotone^+ and that the domains are relevant for the validity of the assumption. For more details on monotone^+ operators and the weak sharpness property, we refer to [22], [23], [20], [24].

Example 1: [20] Consider the variational inequality in (5) where $F(\mathbf{x}) = \text{col}(-x_2, 2x_1)$ and $\boldsymbol{\mathcal{X}} = [0, 1]^2$. The mapping

F is pseudomonotone but not monotone⁺ on \mathcal{X} . The solution set is $\mathcal{X}^* = \{\mathbf{x} \in \mathcal{X} : x_2 = 0\}$ and it holds that

$$\begin{aligned} G(\mathbf{x}) &= \max_{\mathbf{y} \in \mathcal{X}} \langle F(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle = \max_{\mathbf{y} \in \mathcal{X}} -x_1 y_2 - y_1 y_2 + 2x_2 y_1 \\ &= 2x_2 = 2 \text{dist}(\mathbf{x}, \mathcal{X}^*). \end{aligned} \quad (21)$$

Therefore, \mathcal{X}^* satisfies (19) with $\rho = 2$ but, for any $\mathbf{x}^* \in \mathcal{X}^*$, $[T_{\mathcal{X}}(\mathbf{x}^*) \cap N_{\mathcal{X}^*}(\mathbf{x}^*)]^\circ = \{x_2^* \leq 0\}$ and $-F(\mathbf{x}^*) \notin \bigcap_{\mathbf{x}^* \in \mathcal{X}^*} [T_{\mathcal{X}}(\mathbf{x}^*) \cap N_{\mathcal{X}^*}(\mathbf{x}^*)]^\circ$. Thus, the solution set \mathcal{X}^* is not weakly sharp. ■

Example 2: Consider the mapping $F(\mathbf{x}) = \text{col}(-x_2, x_1)$ and the associate variational inequality in (5) with $\mathcal{X} = [0, 1]^2$. Then the mapping F is monotone but not monotone⁺ on \mathcal{X} . The solution set is $\mathcal{X}^* = \{\mathbf{x} \in \mathcal{X} : x_2 = 0\}$ and, similarly to (21), the conditions (19) and (20) hold.

Now, let $\mathcal{X} = \mathbb{R}^2$. In this case, there is only one solution and $\mathcal{X}^* = \{\mathbf{0}_2\}$. However, (20) reads as $\langle F(\mathbf{0}), \mathbf{x} \rangle = 0 \geq \rho \text{dist}(\mathbf{x}, \mathcal{X}^*) = \|\mathbf{x}\|$, which is false. ■

V. CONVERGENCE UNDER UNIQUENESS OF SOLUTION

In light of the considerations in Section IV, we know that a unique solution is also a weak solution and that (20) may hold even if the mapping is not monotone⁺. Therefore, here we consider the case of merely monotone operators but with unique solution and prove that the proposed (non-preconditioned) Algorithm 1 converges to a v-SGNE.

First, the following lemma ensure that the zeros of $\mathcal{A} + \mathcal{B}$ are v-SGNEs.

Lemma 1: Let Assumptions 1-5 hold. Consider the operators \mathcal{T} in (7) and \mathcal{A} and \mathcal{B} in (11). Then:

- (i) If $\omega^* \in \text{zer}(\mathcal{A} + \mathcal{B})$, then \mathbf{x}^* is a v-SGNE of game in (3), i.e., \mathbf{x}^* solves the SVI(\mathcal{X}, \mathbb{F}) in (5). Moreover $\boldsymbol{\lambda}^* = \mathbf{1}_N \otimes \lambda^*$, and $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ satisfy the KKT condition in (6) i.e., $\text{col}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \in \text{zer}(\mathcal{T})$
- (ii) $\text{zer}(\mathcal{T}) \neq \emptyset$ and $\text{zer}(\mathcal{A} + \mathcal{B}) \neq \emptyset$

Proof: It follows from [7, Th. 2]. ■

To ensure that \mathcal{A} and \mathcal{B} have the properties for the convergence result, we make the following assumption.

Assumption 6: \mathbb{F} as in (4) is monotone and $\ell_{\mathbb{F}}$ -Lipschitz continuous for some $\ell_{\mathbb{F}} > 0$. ■

Then, the following holds.

Lemma 2: Let Assumptions 1 and 6 hold and let $\Phi > 0$. Then, \mathcal{A} and \mathcal{B} in (11) have the following properties.

- (i) \mathcal{A} is monotone and $\ell_{\mathcal{A}}$ -Lipschitz continuous.
- (ii) The operator \mathcal{B} is maximally monotone.
- (iii) $\Phi^{-1}\mathcal{A}$ is monotone and ℓ_{Φ} -Lipschitz continuous.
- (iv) $\Phi^{-1}\mathcal{B}$ is maximally monotone.

Proof: It follows similarly to [7], [3]. ■

To guarantee that the weak sharpness property holds, we assume to have a strong solution.

Assumption 7: The SVI in (5) has a unique solution. ■

We can now state the convergence result.

Theorem 1: Let Assumptions 1-7 hold. Then, the sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ generated by Algorithm 1 with F^{SA} as in (8) converges a.s. to a v-SGNE of the game in (3).

Proof: The iterations of Algorithm 1 are obtained by expanding (12) and solving for \mathbf{x}_k , \mathbf{z}_k and $\boldsymbol{\lambda}_k$. Therefore, Algorithm 1 is a SPRG iteration as in (12). The convergence

of the sequence $(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ to a v-GNE of the game in (3) then follows by [13, Prop. 10] and Lemma 1 since $\Phi^{-1}\mathcal{A}$ is monotone by Lemma 2 and has a unique solution. ■

VI. CONVERGENCE UNDER COCOERCIVITY

We know from Section IV that cocoercive operators have the weak sharpness property. Here we consider this case.

Assumption 8: \mathbb{F} is β -cocoercive for some $\beta > 0$. ■

Remark 2: If a function is β -cocoercive, it is also $1/\beta$ -Lipschitz continuous [15, Remark 4.15]. ■

We note that the operator \mathcal{A} in (11) contains a skew symmetric matrix that is not cocoercive. For this reason we consider the splitting in (13). While Lemma 1 guarantees that the zeros of $\mathcal{C} + \mathcal{D}$ are the same as the zeros of \mathcal{T} in (7), we now show the necessary monotonicity properties of the extended operators \mathcal{C} and \mathcal{D} in (13).

Lemma 3: Let Assumptions 1 and 8 hold and let $\Psi > 0$. Then, \mathcal{C} and \mathcal{D} in (13) have the following properties.

- (i) \mathcal{C} is θ -cocoercive where $0 < \theta \leq \min\{\frac{1}{2d^*}, \beta\}$ and d^* is the maximum weighted degree of \mathcal{G}^λ ;
- (ii) \mathcal{D} is maximally monotone;
- (iii) $\Psi^{-1}\mathcal{C}$ is $\theta\tau$ -cocoercive, with $\tau = \frac{1}{|\Psi^{-1}|}$;
- (iv) $\Psi^{-1}\mathcal{D}$ is maximally monotone.

Proof: It follows from [7, Lem. 5] and [7, Lem. 7]. ■

Furthermore, since the preconditioning matrix must be positive definite, we postulate the following assumption on the step sizes [7].

Assumption 9: Let θ be the cocoercivity constant as in Lemma 3, $\tau = \frac{1}{|\Psi^{-1}|} \in (0, \frac{\theta}{8})$ and the step sizes α , ν and σ satisfy, for all $i \in \mathcal{I}$,

$$\begin{aligned} 0 < \alpha_i &\leq (\max_{j=1, \dots, n_i} \{\sum_{k=1}^m |[A_i^T]_{jk}| \} + \tau)^{-1} \\ 0 < \nu_i &\leq (2d_i + \tau)^{-1} \\ 0 < \sigma_i &\leq (\max_{j=1, \dots, m} \{\sum_{k=1}^{k_i} |[A_i]_{jk}| \} + 2d_i + \tau)^{-1} \end{aligned}$$

where $[A_i^T]_{jk}$ indicates the entry (j, k) of the matrix A_i^T . ■

We are now ready to state our convergence result.

Theorem 2: Let Assumptions 1-5 and 8-9 hold. Then, the sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ generated by Algorithm 2 with F^{SA} as in (8) converges a.s. to a v-SGNE of the game in (3).

Proof: The iterations of Algorithm 2 are obtained by expanding (14) and solving for \mathbf{x}_k , \mathbf{z}_k and $\boldsymbol{\lambda}_k$. Therefore, Algorithm 2 is a SPRG iteration as in (14). The convergence of the sequence $(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ to a v-GNE of the game in (3) then follows by [13, Prop. 10] and Lemma 1 since $\Psi^{-1}\mathcal{C}$ is cocoercive by Lemma 3. ■

VII. NUMERICAL SIMULATIONS

Let us propose some numerical evaluations to validate the analysis with a realistic application to an electricity market with capacity constraints, casted as a network Cournot game with markets capacity constraints [7], [2], [13].

We consider a set of $N = 20$ companies selling energy to $m = 7$ markets. Each generator decides the quantity of energy x_i to deliver to the n_i markets it is connected with. Each company i has a production limit of the form $0 < x_i < \gamma_i$ where each component of γ_i is randomly drawn from $[1, 1.5]$. Each company has a cost of production

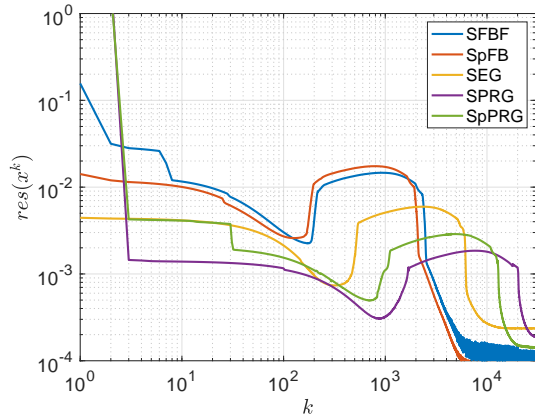


Fig. 1. Relative distance of the primal variable from the solution.

$c_i(x_i) = 1.5x_i + q_i$, where q_i is a given constant. Each market j has a bounded capacity b_j , randomly drawn from $[0.5, 1]$. The collective constraints are then given by $Ax \leq b$ where $A = [A_1, \dots, A_N]$. The prices of the markets are collected in $P : \mathbb{R}^m \times \Xi \rightarrow \mathbb{R}^m$. The uncertainty variable ξ , which represents the demand uncertainty, appears in this functional. P is supposed to be a linear function and reads as $P(\xi) = \bar{P}(\xi) - DAx$. Each component of $\bar{P} = \text{col}(\bar{P}_1(\xi), \dots, \bar{P}_7(\xi))$ is taken with a normal distribution with mean 3 and finite variance. The entries of D are randomly taken in $[0.5, 1]$. The cost function of each agent is given by $\mathbb{J}_i(x_i, x_{-i}, \xi) = c_i(x_i) - \mathbb{E}[P(\xi)^\top (Ax) A_i x_i]$, and it is influenced by the variables of the companies selling in the same market as in [7, Fig. 1]. The dual variables graph is a cycle graph with the addition of the edges (2, 15) and (6, 13) [7].

We simulate the SpFB, the forward-backward-forward (SFBF) and the extragradient (SEG) algorithms to make a comparison with our SPRG and SpPRG, using the SA scheme. The parameters α , ν and σ are taken to be the highest possible that guarantee convergence. As a measure of the distance from the solution, we consider the residual, $\text{res}(x^k) = \|x^k - \text{proj}_C(x^k - F(x^k))\|$, which is equal zero if and only if x is a solution. The plot in Fig. 1 shows how the residual varies in the number of iterations. The performances of SpPRG and SPRG are very similar and the difference in the trajectory is related to the different step sizes which depend on the Lipschitz constant of C in (13) and \mathcal{A} in (11) respectively.

VIII. CONCLUSION

The stochastic projected reflected gradient algorithm is applicable to distributed stochastic generalized Nash equilibrium seeking. To guarantee convergence to a solution and to obtain a distributed algorithm, preconditioning may be used and the pseudogradient mapping should be cocoercive. However, should the equilibrium be unique, the cocoercivity assumption can be reduced to mere monotonicity.

REFERENCES

[1] J. Koshal, A. Nedic, and U. V. Shanbhag, "Regularized iterative stochastic approximation methods for stochastic variational inequality

problems," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 594–609, 2013.

[2] C.-K. Yu, M. Van Der Schaar, and A. H. Sayed, "Distributed learning for stochastic generalized Nash equilibrium problems," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3893–3908, 2017.

[3] B. Franci and S. Grammatico, "A distributed forward-backward algorithm for stochastic generalized Nash equilibrium seeking," *IEEE Transactions on Automatic Control*, 2020.

[4] I. Abada, S. Gabriel, V. Briat, and O. Massol, "A generalized Nash–Cournot model for the northwestern european natural gas markets with a fuel substitution demand function: The gammes model," *Networks and Spatial Economics*, vol. 13, no. 1, pp. 1–42, 2013.

[5] D. Watling, "User equilibrium traffic network assignment with stochastic travel times and late arrival penalty," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1539–1556, 2006.

[6] R. Henrion and W. Römisch, "On m-stationary points for a stochastic equilibrium problem under equilibrium constraints in electricity spot market modeling," *Applications of Mathematics*, vol. 52, no. 6, pp. 473–494, 2007.

[7] P. Yi and L. Pavel, "An operator splitting approach for distributed generalized Nash equilibria computation," *Automatica*, vol. 102, pp. 111–121, 2019.

[8] G. Belgioioso and S. Grammatico, "Projected-gradient algorithms for generalized equilibrium seeking in aggregative games are preconditioned forward-backward methods," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2188–2193.

[9] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[10] S. Cui and U. V. Shanbhag, "On the analysis of reflected gradient and splitting methods for monotone stochastic variational inequality problems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 4510–4515.

[11] R. Bot, P. Mertikopoulos, M. Staudigl, and P. Vuong, "Mini-batch forward-backward-forward methods for solving stochastic variational inequalities," *Stochastic Systems*, 2020.

[12] A. Iusem, A. Jofré, R. I. Oliveira, and P. Thompson, "Extragradient method with variance reduction for stochastic variational inequalities," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 686–724, 2017.

[13] S. Cui and U. V. Shanbhag, "On the analysis of variance-reduced and randomized projection variants of single projection schemes for monotone stochastic variational inequality problems," *arXiv preprint arXiv:1904.11076*, 2019.

[14] Y. Malitsky, "Projected reflected gradient methods for monotone variational inequalities," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 502–520, 2015.

[15] H. H. Bauschke, P. L. Combettes *et al.*, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011, vol. 408.

[16] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[17] U. Ravat and U. V. Shanbhag, "On the characterization of solution sets of smooth and nonsmooth convex stochastic Nash games," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1168–1199, 2011.

[18] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized Nash games and variational inequalities," *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, 2007.

[19] J. V. Burke and M. C. Ferris, "Weak sharp minima in mathematical programming," *SIAM Journal on Control and Optimization*, vol. 31, no. 5, pp. 1340–1359, 1993.

[20] P. Marcotte and D. Zhu, "Weak sharp solutions of variational inequalities," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 179–189, 1998.

[21] F. Yousefian, A. Nedić, and U. V. Shanbhag, "Optimal robust smoothing extragradient algorithms for stochastic variational inequality problems," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 5831–5836.

[22] A. N. Iusem, "On some properties of paramonotone operators," *Journal of Convex Analysis*, vol. 5, pp. 269–278, 1998.

[23] J.-P. Crouzeix, P. Marcotte, and D. Zhu, "Conditions ensuring the applicability of cutting-plane methods for solving variational inequalities," *Mathematical Programming*, vol. 88, no. 3, pp. 521–539, 2000.

[24] Y. Hu and W. Song, "Weak sharp solutions for variational inequalities in banach spaces," *Journal of mathematical analysis and applications*, vol. 374, no. 1, pp. 118–132, 2011.