On Variance-Reduced Extragradient Methods for Stochastic Generalized Nash Equilibrium Problems

On variance-reduced Extragradient methods for stochastic generalized Nash equilibrium problems

Barbara Franci

***Abstract*— We study variance reduction schemes for stochastic generalized Nash equilibrium problems. Specifically, we consider two instances of the extragradient algorithm to find a Nash equilibrium and show their convergence under weaker assumptions than the literature. In the particular case where we can write the cost function as a finite sum, we also propose a novel approximation scheme that sensibly lowers the computational burden. Numerical simulations suggest that the performance of the new approximation scheme can improve the computations also in the fully stochastic (infinite) case.**

***Index Terms*— Game Theory, stochastic generalized Nash equilibrium problems, variance reduction.**

I. Introduction

A generalized Nash equilibrium problem (GNEP) involves a set of agents solving a collection of optimization problems, coupled with the other participants' decision variables via the cost functions and feasibility constraints. This type of problems is widely studied [1]–[4] thanks to their possible applications [5]–[7]. When the cost functions are expected-valued and affected by a random variable, we talk about a stochastic generalized Nash equilibrium problem (SGNEP).

The available Nash equilibrium (NE) seeking algorithms rely on the fact that GNEPs can be re-written as a variational inequality (VI), hence opening to a variety of possible iterative schemes. The challenges are however twofold: VIs are not supposed to be split among the participants and heavy computations of the game mapping are required. In other words, the agents cannot find a NE without information on the other participants and they have to perform computationally expensive steps in any iterative scheme. To overcome the first issue, the NE are characterized using the Karush–Kuhn–Tucker (KKT) conditions and the problem is re-written as a monotone inclusion. Thanks to operator theory, the iterations can then be split among the agents to obtain distributed iterations, where the participants need to know only their own objective function and their contribution to the coupling constraints [1]–[3].

The computation of the game mapping is a more complicated matter. There are only few available options, in fact, to approximate it, hence limiting the options to find a solution to (S)GNEPs. However, some algorithms have been proposed in the literature. The most common iterative scheme for these problems is stochastic gradient descent (SGD) [8] and its variants [9], [10], also referred to as stochastic forward-backward algorithms in the (S)GNEP literature [1], [3]. It is however known that SGD requires strong assumptions to guarantee convergence. In fact, having computationally light iterations comes at the price of assuming strong monotonicity or cocoercivity of the game mapping [1], [3], [9]. Other alternatives have been proposed that use mere monotonicity but are often slow and paired with relaxation steps or regularizations [2], [11]–[13]. For these reasons, we consider the extragradient (EG) scheme [14]–[16] for SGNEPs. Although this algorithm is characterized by two gradient-like steps, it weakens the monotonicity assumptions hence providing a valid alternative. Specifically, our contributions are the following:

- We adapt to SGNEPs the stochastic extragradient (SEG) scheme proposed in [16] for stochastic variational inequality (SVI). This algorithm provides the first scheme for NE seeking that converge under pseudomonotonicity, a condition that is even weaker than monotonicity, hence improving the results in [2], [3], [5], [11], [13].
- For the particular case where the cost functions can be written as a finite sum, we propose the loopless extragradient (LEG) scheme [15] to compute a NE. In this case, monotonicity is required but we gain a computationally lighter approximation schemes for NE seeking, compared to [2], [3], [12], [13], [16].

Both the variants are named *variance-reduced* EG scheme. Iteratively reducing the variance is a key component of (SG)NE learning but it can be achieved in different ways. In SEG, an increasing number of samples is taken at each iteration so that asymptotically the approximation will be close enough to the exact game mapping. LEG instead reduces the variance by computing from time to time the full gradient on a suitably selected point. The point of view of the two schemes is slightly different: while the first is mostly meant to approximate fully stochastic problems, the second was born to improve the complexity of deterministic methods [15]. In our numerical experiments, LEG shows similar performance as the single-step schemes in [1]–[3].

a) Notation and preliminaries: $\mathbb{R}$ is the set of real numbers while $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$. $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ denotes the standard inner product and $\| \cdot \|$ is the associated euclidean norm. We write $\Phi \succ 0$ to indicate a a positive definite matrix, i.e., $x^\top \Phi x > 0$. Given a symmetric $\Phi \succ 0$, the Φ-

B. Franci is with the Department of Mathematical Sciences, Politecnico di Torino, Turin, Italy (barbara.franci@polito.it).

induced inner product is $\langle x,y\rangle_\Phi = \langle \Phi x, y\rangle$ and the Φ-induced norm is $\|x\|_\Phi = \sqrt{\langle \Phi x, x\rangle}$. $A \otimes B$ denotes the Kronecker product between matrices A and B. $\mathbf{0}_m$ indicates the vector with m entries all equal to 0. Given $x_1,\dots,x_N \in \mathbb{R}^n$, $\boldsymbol{x} := \operatorname{col}(x_1,\dots,x_N) = \left[x_1^\top,\dots,x_N^\top\right]^\top$. $J_F = (\mathrm{Id}+F)^{-1}$ is the resolvent of the operator $F : \mathbb{R}^n \to \mathbb{R}^n$ and Id indicates the identity operator For a closed set $C \subseteq \mathbb{R}^n$, the mapping $\operatorname{proj}_C : \mathbb{R}^n \to C$ denotes the projection onto C, i.e., $\operatorname{proj}_C(x) = \operatorname{argmin}_{y\in C}\|y-x\|$. The residual mapping is defined as $\operatorname{res}(x^k) = \|x^k - \operatorname{proj}_C(x^k - F(x^k))\|$. We denote the subdifferential as the operator $\partial g(x) = \{u \in \Omega \mid (\forall y \in \Omega)\langle y-x,u\rangle + g(x) \le g(y)\}$. The proximal operator is defined as $\operatorname{prox}_g(v) = \operatorname{argmin}_{u\in\Omega}\{g(u) + \frac{1}{2}\|u-v\|^2\} = J_{\partial g}(v)$. ι_C is the indicator function of the set C, i.e., $\iota_C(x) = 0$ if $x \in C$ and $\iota_C(x) = \infty$ otherwise while $\mathrm{N}_C : \mathbb{R}^n \to \mathbb{R}^n$ is the normal cone operator of the set C, i.e., $\mathrm{N}_C(x) = \varnothing$ if $x \notin C$, $\left\{v \in \mathbb{R}^n \middle| \sup_{z\in C} v^\top(z-x) \le 0\right\}$ otherwise.

Given a mapping $F : \operatorname{dom}(F) \subseteq \mathbb{R}^n \to \mathbb{R}^n$, we say that: F is ℓ-Lipschitz continuous if for all $x,y \in \operatorname{dom}(F)$ $\|F(x) - F(y)\| \le \ell\|x-y\|$, for $\ell > 0$; F is nonexpansive if it is 1-Lipschitz continous; F is monotone if for all $x,y \in \operatorname{dom}(F)$ $\langle F(x)-F(y), x-y\rangle \ge 0$; F is pseudomonotone if for all $x,y \in \operatorname{dom}(F)$ $\langle F(y), x-y\rangle \ge 0 \Rightarrow \langle F(x), x-y\rangle \ge 0$;

II. Stochastic Generalized Nash equilibrium problems

Let us consider a set $\mathcal{I} = \{1,\dots,N\}$ of noncooperative agents, each choosing a strategy $x_i \in \mathbb{R}^{n_i}$ with the aim of solving a collection of optimization problems within a certain constraints set parametrized by the decision of variables of the other participants $\boldsymbol{x}_{-i} = \operatorname{col}((x_j)_{j\neq i}) \in \mathbb{R}^{n-n_i}$, $n = \sum_{i=1}^N n_i$. Namely, the agents are involved in an SGNEP:

$$\forall i \in \mathcal{I}: \quad \begin{cases} \min\limits_{x_i \in \Omega_i} & \mathbb{J}_i(x_i, \boldsymbol{x}_{-i}) \\ \text{s.t.} & g(x_i, \boldsymbol{x}_{-i}) \le \mathbf{0}_m. \end{cases} \tag{1}$$

The cost function of agent $i \in \mathcal{I}$ is defined as

$$\mathbb{J}_i(x_i, \boldsymbol{x}_{-i}) = \mathbb{E}_{\mathbb{P}}[J_i(x_i, \boldsymbol{x}_{-i}, \varpi(\xi))] + f_i(x_i), \tag{2}$$

for a measurable function $J_i : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$. $\mathbb{E}_{\mathbb{P}}$ indicates the mathematical expectation with respect to the distribution of the random variable $\varpi : \Xi \to \mathbb{R}^d$ in the probability space $(\Xi, \mathcal{F}, \mathbb{P})$. As common in the literature [16], [17], from now on, we write the sample ξ for the random variable $\varpi(\xi)$ and we avoid indicating the subscript $\mathbb{P}$. Hence, the cost function in (2) is composed of a smooth part $\mathbb{E}[J(\cdot)]$ and a nonsmooth part $f_i : \mathbb{R}^{n_i} \to \bar{\mathbb{R}}$ indicating a local cost. We assume that $\mathbb{E}[J_i(\boldsymbol{x},\xi)]$ is well defined for all $\boldsymbol{x} = \operatorname{col}((x_i)_{i\in\mathcal{I}})$. Moreover, the agents face private constraints $\Omega_i \subseteq \mathbb{R}^{n_i}$, $i \in \mathcal{I}$, and coupling constraints expressed via the function $g : \mathbb{R}^n \to \mathbb{R}^m$ which takes the form $g(\boldsymbol{x}) := \sum_{i=1}^N g_i(x_i)$, with $g_i : \mathbb{R}^{n_i} \to \mathbb{R}^m, i \in \mathcal{I}$. The constraints are collected in

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\Omega} \cap \{\boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}) \le \mathbf{0}_m\}, \tag{3}$$

where $\boldsymbol{\Omega} = \prod_{i=1}^N \Omega_i$. For agent i, we let $\mathcal{X}_i(\boldsymbol{x}_{-i}) = \{y_i \in \Omega_i \mid g(y_i, \boldsymbol{x}_{-i}) \le \mathbf{0}_m\}$ while $\boldsymbol{\mathcal{X}}_{-i} = \prod_{j\neq i} \mathcal{X}_j(\boldsymbol{x}_{-j})$ denotes the constraints of all agents but i.

Assumption 1: For all $i \in \mathcal{I}$, the function f_i in (2) is lower semicontinuous and convex and $\operatorname{dom}(f_i) = \Omega_i$. For all $\boldsymbol{x}_{-i} \in \boldsymbol{\mathcal{X}}_{-i}$ the function $\mathbb{E}[J_i(\cdot, \boldsymbol{x}_{-i}, \xi)]$ is convex and continuously differentiable. □

Assumption 2: For all $i \in \mathcal{I}$, the set Ω_i is nonempty, closed and convex and the set $\boldsymbol{\mathcal{X}}$ satisfies Slater's constraint qualification. The mapping g in (3) is ℓ_g-Lipschitz continuous and with bounded gradient, i.e., there exists $B_{\nabla g} > 0$ such that $\sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \|\nabla g(\boldsymbol{x})\| \le B_{\nabla g}$. □

The goal of the agents is to find a stochastic generalized Nash equilibrium (SGNE), i.e., a collective strategy $\boldsymbol{x}^* \in \boldsymbol{\mathcal{X}}$ such that for all $i \in \mathcal{I}$

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

To ensure that at least one SGNE exists, we make further assumptions on the cost functions [17, Section 3.1].

Assumption 3: For each $i \in \mathcal{I}$ and for each $\xi \in \Xi$, the function $J_i(\cdot, \boldsymbol{x}_{-i}, \xi)$ is convex, continuously differentiable, and Lipschitz continuous with Lipschitz constant $\ell_i(\boldsymbol{x}_{-i}, \xi)$, integrable in ξ for each $\boldsymbol{x}_{-i}$. The function $J_i(x_i, \boldsymbol{x}_{-i}, \cdot)$ is measurable. □

A common approach [1]–[3], [11] to find an SGNE of (1) is to rewrite the problem as an SVI. A variational problem is that of finding $\boldsymbol{x}^* \in \boldsymbol{\mathcal{X}}$ such that for all $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$

$$\langle \mathbb{F}(\boldsymbol{x}^*), \boldsymbol{x} - \boldsymbol{x}^*\rangle + \sum_{i\in\mathcal{I}}\{f_i(x_i) - f_i(x_i^*)\} \ge 0, \tag{4}$$

where we have used the pseudogradient mapping

$$\mathbb{F}(\boldsymbol{x}) = \operatorname{col}((\mathbb{E}[\nabla_{x_i} J_i(x_i, \boldsymbol{x}_{-i}, \xi)])_{i\in\mathcal{I}}). \tag{5}$$

This mapping is well defined because of Assumption 3. It has been shown in [18, Prop. 12.4], [19, Thm. 3.9], [17, Lem. 3.3] that, under Assumptions 1–3, any solution of the SVI in (4) is also an SGNE of the game in (1), called, in this case, variational stochastic generalized Nash equilibrium (v-SGNE). Note that, according to [18], [19], the two problems are not completely equivalent, i.e., there might be SGNE that are not v-SGNE. We make the blanket assumption that a v-SGNE exists but more details on this can be found in [18, Sec. 12.3], [17, Sec. 3.1].

To properly account for the coupling constraints, we recast the SGNEP as a monotone inclusion by characterizing its equilibria in terms of the KKT conditions of the optimization problems in (1). Therefore, by letting $\lambda_i \in \mathbb{R}^m_{\ge 0}$, $i \in \mathcal{I}$ be the dual variable associated with the coupling constraints, we write the KKT conditions in compact form as

$$\begin{aligned} \mathbf{0}_{n+m} &\in \mathcal{T}(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathcal{A}(\boldsymbol{x}, \boldsymbol{\lambda}) + \mathcal{B}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ &= \begin{bmatrix} \mathbb{F}(\boldsymbol{x}) + \nabla g(\boldsymbol{x})^\top \boldsymbol{\lambda} \\ -g(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} \partial f(\boldsymbol{x}) + \mathrm{N}_{\boldsymbol{\Omega}}(\boldsymbol{x}) \\ \mathrm{N}_{\mathbb{R}^m_{\ge 0}}(\boldsymbol{\lambda}) \end{bmatrix}, \end{aligned} \tag{6}$$

where $\partial f(\boldsymbol{x}) = \partial f_1(x_1) \times \cdots \times \partial f_N(x_N)$. The v-SGNE of the game in (1) correspond to the zeros of the set-valued mapping $\mathcal{T} : \mathcal{X} \times \mathbb{R}^m_{\ge 0} \rightrightarrows \mathbb{R}^n \times \mathbb{R}^m$, when the agents reach consensus on the dual variables, i.e., $\lambda_i = \lambda$ for all $i \in \mathcal{I}$ [19, Thm. 4.6].

III. Distributed Extragradient algorithms

In this section we describe the details that lead to the distributed iterations in Algorithms 1 and 2. These algorithms are two instances of the EG scheme [14]–[16] and, loosely

Algorithm 1 Stochastic Extragradient (SEG)

Initialization: $x_i^0 \in \Omega_i, \lambda_i^0 \in \mathbb{R}^m_{\geq 0}$, and $z_i^0 \in \mathbb{R}^m$.
Iteration k: Agent i
(1) Receives x_j^k for all $j \in \mathcal{N}_i^J$ and z_j^k, λ_j^k for all $j \in \mathcal{N}_i^\lambda$, then updates:

$$\begin{aligned}
x_i^{k+\frac{1}{2}} &= \mathrm{prox}_{f_i}[x_i^k - \alpha_i(F_i^{S_k}(x_i^k, \boldsymbol{x}_{-i}^k, \xi_i^k) + \nabla g_i(x_i^k)^\top \lambda_i^k)] \\
z_i^{k+\frac{1}{2}} &= z_i^k - \nu_i \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}(\lambda_i^k - \lambda_j^k) \\
\lambda_i^{k+\frac{1}{2}} &= \mathrm{proj}_{\mathbb{R}^m_{\geq 0}}\{\lambda_i^k + \tau_i g_i(x_i^k) \\
&\quad + \tau \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}[(z_i^k - z_j^k) - (\lambda_i^k - \lambda_j^k)]\}
\end{aligned}$$

(2) Receives $x_j^{k+\frac{1}{2}}$ for all $j \in \mathcal{N}_i^J$ and $z_j^{k+\frac{1}{2}}, \lambda_j^{k+\frac{1}{2}}$ for all $j \in \mathcal{N}_i^\lambda$, then updates:

$$\begin{aligned}
x_i^{k+1} &= \mathrm{prox}_{f_i}[x_i^k - \alpha_i(F_i^{S_k}(x_i^{k+\frac{1}{2}}, \boldsymbol{x}_{-i}^{k+\frac{1}{2}}, \xi_i^{k+1}) \\
&\quad + \nabla g_i(x_i^{k+\frac{1}{2}})^\top \lambda_i^{k+\frac{1}{2}})] \\
z_i^{k+1} &= z_i^k - \nu_i \textstyle\sum_{j\in\mathcal{N}_i^\lambda} w_{ij}(\lambda_i^{k+\frac{1}{2}} - \lambda_j^{k+\frac{1}{2}}) \\
\lambda_i^{k+1} &= \mathrm{proj}_{\mathbb{R}^m_{\geq 0}}\{\lambda_i^k + \tau_i g_i(x_i^{k+\frac{1}{2}}) \\
&\quad + \tau \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}[(z_i^{k+\frac{1}{2}} - z_j^{k+\frac{1}{2}}) - (\lambda_i^{k+\frac{1}{2}} - \lambda_j^{k+\frac{1}{2}})]\}
\end{aligned}$$

Algorithm 2 Loopless Extragradient (LEG)

Initialization: $x_i^0 \in \Omega_i, y_i^0 = x_i^0, \lambda_i^0 \in \mathbb{R}^m_{\geq 0}$, and $z_i^0 \in \mathbb{R}^m$.
Iteration k: Agent i
(1) Updates the primal variable

$$\bar{x}_i^k = \alpha x_i^k + (1-\alpha) y_i^k$$

(2) Receives y_j^k for all $j \in \mathcal{N}_i^J$ and z_j^k, λ_j^k for all $j \in \mathcal{N}_i^\lambda$, then updates:

$$\begin{aligned}
x_i^{k+\frac{1}{2}} &= \mathrm{prox}_{f_i}[\bar{x}_i^k - \alpha_i(\mathbb{F}_i(y_i^k, \boldsymbol{y}_{-i}^k) + \nabla g_i(y_i^k)^\top \lambda_i^k)] \\
z_i^{k+\frac{1}{2}} &= z_i^k - \nu_i \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}(\lambda_i^k - \lambda_j^k) \\
\lambda_i^{k+\frac{1}{2}} &= \mathrm{proj}_{\mathbb{R}^m_{\geq 0}}\{\lambda_i^k + \tau_i g_i(y_i^k) \\
&\quad + \tau \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}[(z_i^k - z_j^k) - (\lambda_i^k - \lambda_j^k)]\}
\end{aligned}$$

(3) Receives $x_j^{k+\frac{1}{2}}$ for all $j \in \mathcal{N}_i^J$ and $z_j^{k+\frac{1}{2}}, \lambda_j^{k+\frac{1}{2}}$ for all $j \in \mathcal{N}_i^\lambda$, then updates:

$$\begin{aligned}
x_i^{k+1} &= \mathrm{prox}_{f_i}[\bar{x}_i^k - \alpha_i(\mathbb{F}_i(y_i^k, \boldsymbol{y}_{-i}^k) + F_i^\xi(x_i^{k+\frac{1}{2}}, \boldsymbol{x}_{-i}^{k+\frac{1}{2}}, \xi_i^{k+1}) \\
&\quad - F^\xi(y_i^k, \boldsymbol{y}_{-i}^k, \xi_i^k) + \nabla g_i(x_i^{k+\frac{1}{2}})^\top \lambda_i^{k+\frac{1}{2}})] \\
z_i^{k+1} &= z_i^k - \nu_i \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}(\lambda_i^{k+\frac{1}{2}} - \lambda_j^{k+\frac{1}{2}}) \\
\lambda_i^{k+1} &= \mathrm{proj}_{\mathbb{R}^m_{\geq 0}}\{\lambda_i^k + \tau_i g_i(x_i^{k+\frac{1}{2}}) \\
&\quad + \tau \sum_{j\in\mathcal{N}_i^\lambda} w_{ij}[(z_i^{k+\frac{1}{2}} - z_j^{k+\frac{1}{2}}) - (\lambda_i^{k+\frac{1}{2}} - \lambda_j^{k+\frac{1}{2}})]\}
\end{aligned}$$

(4) Updates

$$y_i^{k+1} = \begin{cases} x_i^{k+1} & \text{with probability } p \\ y_i^k & \text{with probability } 1-p \end{cases} \tag{7}$$

speaking, differ on the approximation scheme used to compute the estimate of the pseudogradient mapping. More details are postponed to the dedicated sections, while here we collect the assumptions that are common to both instances.

The general EG algorithm is compactly given by

$$\begin{aligned}
\boldsymbol{w}^{k+\frac{1}{2}} &= (\mathrm{Id} + \Phi^{-1}\mathcal{B})^{-1}(\boldsymbol{w}^k - \Phi^{-1}\mathcal{A}(\boldsymbol{w}^k)) \\
\boldsymbol{w}^{k+1} &= (\mathrm{Id} + \Phi^{-1}\mathcal{B})^{-1}(\boldsymbol{w}^k - \Phi^{-1}\mathcal{A}(\boldsymbol{w}^{k+\frac{1}{2}}))
\end{aligned} \tag{8}$$

where Φ is a matrix including the inverse of the step-size sequences and, for the moment, $\mathcal{A}$ and $\mathcal{B}$ are as in (6). In the reminder of this section, we show how to adjust the compact EG scheme in (8) to obtain Algorithms 1 and 2. First, let $\mathcal{N}_i^J$ be the set of agents j whose decisions affect the cost function of agent i, i.e., $j \in \mathcal{N}_i^J$ if $\mathbb{J}_i(x_i, \boldsymbol{x}_{-i})$ explicitly depends on x_j. Moreover, note that in both algorithms, an auxiliary variable z_i is used to force consensus on the dual variables. To this aim, let us also introduce the graph $\mathcal{G}^\lambda = (\mathcal{I}, \mathcal{E}^\lambda)$ used to share the local copy of the dual variable λ_i and the auxiliary variable z_i. In particular, $(i,j) \in \mathcal{E}^\lambda$ if agent j shares its $\{\lambda_j, z_j\}$ with agent i. The set $\mathcal{N}_i^\lambda = \{j \in \mathcal{I} : (i,j) \in \mathcal{E}^\lambda\}$ contains the neighboring agents of $i \in \mathcal{I}$ in $\mathcal{G}^\lambda$.

Assumption 4: The graph $\mathcal{G}^\lambda$ is undirected and connected.

Let the matrix $W = [w_{ij}]_{i,j\in\mathcal{I}} \in \mathbb{R}^{N\times N}$ be the weighted adjacency matrix of the graph $\mathcal{G}^\lambda$ and $d_i = \sum_{j=1}^N w_{ij}$ be the degree of agent i. The Laplacian of $\mathcal{G}^\lambda$ is $L = D - W \in \mathbb{R}^{N\times N}$ where $D = \mathrm{diag}\{d_1, \dots, d_N\}$. By Assumption 4 and the Baillon-Haddard Theorem, the Laplacian is cocoercive. Then, to force consensus on the dual variables, the constraint $\mathbf{L}\boldsymbol{\lambda} = 0$ is usually imposed, where $\mathbf{L} = L \otimes \mathrm{Id}_m \in \mathbb{R}^{Nm\times Nm}$ and $\boldsymbol{\lambda} = \mathrm{col}(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{Nm}$ [1]. Accordingly, we expand the collective variable to $\boldsymbol{w} = \mathrm{col}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda})$ and adjust the operators $\mathcal{A}$ and $\mathcal{B}$ in (6) to account for the auxiliary variable $\boldsymbol{z} = \mathrm{col}(z_1, \dots, z_N) \in \mathbb{R}^{Nm}$:

$$\bar{\mathcal{A}} : \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} \mapsto \begin{bmatrix} \mathbb{F}(\boldsymbol{x}) \\ 0 \\ \mathbf{L}\boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \nabla g(\boldsymbol{x})^\top \boldsymbol{\lambda} \\ \mathbf{L}\boldsymbol{\lambda} \\ -g(\boldsymbol{x}) - \mathbf{L}\boldsymbol{z} \end{bmatrix}, \tag{9}$$

$$\bar{\mathcal{B}} : \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} \mapsto \begin{bmatrix} \partial f(\boldsymbol{x}) + \mathrm{N}_{\boldsymbol{\Omega}}(\boldsymbol{x}) \\ \mathbf{0} \\ \mathrm{N}_{\mathbb{R}^m_{\geq 0}}(\lambda) \end{bmatrix}. \tag{10}$$

The zeros of $\bar{\mathcal{A}} + \bar{\mathcal{B}}$ are the zeros of $\mathcal{T}$ in (6), i.e., given $\boldsymbol{w}^* \in \mathrm{zer}(\bar{\mathcal{A}} + \bar{\mathcal{B}})$, $\boldsymbol{x}^*$ is a v-SGNE of game in (1) and $(\boldsymbol{x}^*, \lambda^*)$ satisfy the KKT conditions [2, Lem. 1], [1, Thm. 2]. The properties of the operator $\bar{\mathcal{A}}$ in (9) depend on the properties of $\mathbb{F}$ and are described in the next sections since the two algorithms have different requirements. The operator $\bar{\mathcal{B}}$ instead is the same for both schemes.

Lemma 1: Let Assumptions 1 and 2 hold. Then, the following holds for the operator $\bar{\mathcal{B}}$ in (10):

1) $\bar{\mathcal{B}}$ is maximally monotone.
2) $\Phi^{-1}\bar{\mathcal{B}}$ is maximally monotone.

Proof: Analogous to [2, Lem. 2]. ■

We close this section by introducing some notation and assumptions on the stochastic oracle, i.e., on the approximation of the pseudogradient mapping. To this aim, let $F^{\xi}(\boldsymbol{x},\boldsymbol{\xi}) = \mathrm{col}((F_i^{\xi}(\boldsymbol{x},\xi_i))_{i\in\mathcal{I}})$ where, for all $i\in\mathcal{I}$,

$$F_i^{\xi}(\boldsymbol{x},\xi_i) = \nabla_{x_i} J_i(\boldsymbol{x},\xi_i), \tag{11}$$

and $\boldsymbol{\xi} = \mathrm{col}((\xi_i)_{i\in\mathcal{I}})$. In words, $F_i^{\xi}(\boldsymbol{x},\xi_i)$ is the partial gradient of agent $i\in\mathcal{I}$ computed using ξ_i as a sample of the random variable and $F^{\xi}(\boldsymbol{x},\boldsymbol{\xi})$ is the stacked vector of all the oracles of all the agents.

Assumption 5: For all $\boldsymbol{x}\in\boldsymbol{\mathcal{X}}$, $\mathbb{E}[F^{\xi}(\boldsymbol{x},\boldsymbol{\xi})] = \mathbb{F}(\boldsymbol{x})$ a.s., i.e., the stochastic oracle is unbiased. □

It follows that the stochastic error $\epsilon(\boldsymbol{x},\boldsymbol{\xi}) = F^{\xi}(\boldsymbol{x},\boldsymbol{\xi}) - \mathbb{F}(\boldsymbol{x})$ has zero mean. In particular, let $\mathcal{F} = \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ be the filtration defined as the family of σ-algebras where $\mathcal{F}_0 = \sigma(\boldsymbol{x}_0)$ and $\mathcal{F}_k = \sigma(\boldsymbol{x}_0,\xi_1,\xi_2,\dots,\xi_k)$ is such that $\mathcal{F}_k\subseteq\mathcal{F}_{k+1}$ for all $k\in\mathbb{N}$. Then, for all $k\in\mathbb{N}$, a.s., $\mathbb{E}\left[\epsilon^k|\mathcal{F}_k\right] = 0$.

A. Stochastic Extragradient Scheme

The approximation scheme used in Algorithm 1 relies on mini-batches [2], [3], [16]. In particular, at each iteration k, the agents have access to a pool of samples of the random variable and are able to compute an approximation of $\mathbb{F}(\cdot)$ of the form $F^{S_k}(\boldsymbol{x}^k,\boldsymbol{\xi}^k) = \mathrm{col}((F_i^{S_k}(\boldsymbol{x}^k,\bar{\xi}_i^k))_{i\in\mathcal{I}})$ with

$$F_i^{S_k}(\boldsymbol{x}^k,\bar{\xi}_i^k) = \frac{1}{S_k}\sum_{t=1}^{S_k} F_i^{\xi}(x_i^k,\boldsymbol{x}_{-i}^k,\xi_i^{(t)}), \tag{12}$$

where $\boldsymbol{\xi}^k = \mathrm{col}(\bar{\xi}_1^k,\dots,\bar{\xi}_N^k)$, $k\in\mathbb{N}$, and $\bar{\xi}_i^k = \mathrm{col}(\xi_i^{(1)},\dots,\xi_i^{(S_k)})$, $i\in\mathcal{I}$, collect the sequences of i.i.d. samples of the random variable.

Assumption 6: The batch size sequence $(S_k)_{k\geq 1}$ is such that $\sum_{k\in\mathbb{N}}\frac{1}{S_k}<\infty$. □

Assumption 6 is satisfied choosing for all $k>0$, $S_k\geq a(k+b)^{c+1}$ for some $a,b,c>0$, and it is a common assumption together with the bounded variance of the stochastic error. For simplicity, we assume uniform bounded variance but more general bounds can be considered [2], [12], [16].

Assumption 7: There exists $\sigma>0$ such that $\sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}}\mathbb{E}[\|\epsilon(\boldsymbol{x},\boldsymbol{\xi})\|^2]\leq\sigma^2$. □

Remark 1: Variance reduction for the approximation scheme in (12) follows from Assumptions 6 and 7. In particular, for all $k>0$, $\mathbb{E}[\|F^{S_k}(\boldsymbol{x},\xi)-\mathbb{F}(\boldsymbol{x})\|^2 \mid \mathcal{F}_k]\leq\frac{C\sigma^2}{S_k}$ a.s., where $C>0$ is a constant [3, Lem. 6]. □

In light of the approximation scheme in (12), we replace $\bar{\mathcal{A}}$ with $\mathcal{A}^{S_k}$, i.e., the operator in (9) where instead of the exact pseudogradient mapping $\mathbb{F}(\boldsymbol{x})$ we use the approximation $F^{S_k}(\boldsymbol{x},\boldsymbol{\xi})$; the remaining quantities involving the constraints and dual variables remain the same, being deterministic.

Then, Algorithm 1 can be written in compact form as

$$\begin{aligned}\boldsymbol{w}^{k+\frac{1}{2}} &= (\mathrm{Id}+\Phi^{-1}\bar{\mathcal{B}})^{-1}(\boldsymbol{w}^k-\Phi^{-1}\mathcal{A}^{S_k}(\boldsymbol{w}^k,\boldsymbol{\xi}^k)).\\ \boldsymbol{w}^{k+1} &= (\mathrm{Id}+\Phi^{-1}\bar{\mathcal{B}})^{-1}(\boldsymbol{w}^k-\Phi^{-1}\mathcal{A}^{S_k}(\boldsymbol{w}^{k+\frac{1}{2}},\boldsymbol{\xi}^{k+\frac{1}{2}})),\end{aligned} \tag{13}$$

where $\Phi = \mathrm{diag}(\alpha^{-1},\nu^{-1},\sigma^{-1})\succ 0$ contains the inverse of step size sequences and α^{-1}, ν^{-1}, σ^{-1} are diagonal matrices of appropriate dimensions. Now, we study the convergence of Algorithm 1. First, to ensure that $\bar{\mathcal{A}}$ has the properties that we use for the analysis, we make the following assumption.

Assumption 8: $\mathbb{F}$ in (5) is pseudomonotone and $\ell_{\mathbb{F}}$-Lipschitz continuous for some $\ell_{\mathbb{F}}>0$. □

Lemma 2: Let Assumptions 1–4 and 8 hold and let $\Phi\succ 0$. Then, the following holds for the operator $\bar{\mathcal{A}}$ in (9):

1) $\bar{\mathcal{A}}$ is pseudomonotone and $\ell_{\bar{\mathcal{A}}}$-Lipschitz continuous.
2) $\Phi^{-1}\bar{\mathcal{A}}$ is pseudomonotone and ℓ_{Φ}-Lipschitz continuous.

Proof: Item 1) follows from the fact $\bar{\mathcal{A}}$ in (9) is given by the sum of two terms. Let $\mathcal{A}_1$ be the first, containing the pseudogradient $\mathbb{F}$ while $\mathcal{A}_2$ contains the coupling constraints. Then pseudomonotonicy of $\mathcal{A}_1$ follows from Assumption 8 and from the cocoercivity of the Laplacian (Assumption 4). In fact, given $\boldsymbol{w}=\mathrm{col}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\lambda})$ and $\boldsymbol{v}=\mathrm{col}(\boldsymbol{y},\boldsymbol{u},\boldsymbol{\mu})$, $\langle\mathcal{A}_1(\boldsymbol{w}),\boldsymbol{v}-\boldsymbol{w}\rangle = \langle F(\boldsymbol{x}),\boldsymbol{y}-\boldsymbol{x}\rangle + \langle\boldsymbol{L}\boldsymbol{\lambda},\boldsymbol{\lambda}-\boldsymbol{\mu}\rangle \geq 0$; analogously, it follows that $\langle\mathcal{A}_1(\boldsymbol{v}),\boldsymbol{v}-\boldsymbol{w}\rangle = \langle F(\boldsymbol{y}),\boldsymbol{y}-\boldsymbol{x}\rangle + \langle\boldsymbol{L}\boldsymbol{\mu},\boldsymbol{\lambda}-\boldsymbol{\mu}\rangle \geq 0$. $\mathcal{A}_2$ instead is monotone (hence pseudomonotone) [20, Thm. 1], [2, Lem. 2]. The sum is then pseudomonotone. Lipschitz continuity follows along the line of [2, Lem. 2] and the Lipschitz constant depends on $\ell_{\mathbb{F}}$, ℓ_g and $B_{\nabla g}$. $\Phi^{-1}\bar{\mathcal{A}}$ is then pseudomonotone and Lipschitz continuous in the Φ-induced norm because $\bar{\mathcal{A}}$ is. ■

Lastly, we indicate how to choose the step-size sequence.

Assumption 9: The steps size sequence is such that $0<\|\Phi^{-1}\|\leq\frac{1}{\ell_{\bar{\mathcal{A}}}\sqrt{6}}$ where $\ell_{\bar{\mathcal{A}}}$ is the Lipschitz constant of $\bar{\mathcal{A}}$. □

We are now ready to state our convergence result.

Theorem 1: Let Assumptions 1–9 hold. Then, the sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ generated by Algorithm 1 with the approximation (12) converges a.s. to a v-SGNE of the game in (1).

Proof: Let us define $V(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_{i\in\mathcal{I}} f_i(x_i)+\iota_{\Omega_i}(x_i)+\iota_{\mathbb{R}^m_{\geq}}(\lambda_i)$. Then, $\Phi^{-1}\bar{\mathcal{B}} = \partial V(\boldsymbol{x},\boldsymbol{\lambda})$. Moreover, by [21, Prop. 16.44], $(\mathrm{Id}+\Phi^{-1}\bar{\mathcal{B}})^{-1} = (\mathrm{Id}+\partial V(\boldsymbol{x},\boldsymbol{\lambda}))^{-1} = \mathrm{prox}_V$. Since this operator is nonexpansive by Lemma 1 and $\Phi^{-1}\bar{\mathcal{A}}$ is pseudomonotone by Lemma 2, we can use the same step as [16, Thm. 3.18] to show that the residual $\mathrm{res}(\boldsymbol{w}^k) = \|\boldsymbol{w}^k-(\mathrm{Id}+\Phi^{-1}\bar{\mathcal{B}})^{-1}(\boldsymbol{w}^k-\Phi^{-1}\bar{\mathcal{A}}(\boldsymbol{w}^k))\|^2$ goes to zero, hence showing that $\boldsymbol{w}^k$ converges asymptotically to a fixed point of (8) and therefore to a v-SGNE. ■

B. Loopless Extragradient Scheme

Let us consider the particular case where the cost function, hence the partial gradients, can be written as a finite sum. Specifically, let the i-th component of $\mathbb{F}$ in (5) be such that

$$\mathbb{F}_i(\boldsymbol{x}) = \sum_{t=1}^{T} h_{it}(\boldsymbol{x}). \tag{14}$$

Note that this case is deterministic. We assume however that T is sufficiently big that $\mathbb{F}_i(\boldsymbol{x})$ is hard to compute for agent $i\in\mathcal{I}$ or simply too expensive. As a particular case, one can consider the scenario where for all $i\in\mathcal{I}$, $\mathbb{F}_i(\boldsymbol{x})$ is given exactly as the sum of T empirical gradients F_i^{ξ} as in (11) so that $h_{it}(\boldsymbol{x}) = F_i^{\xi}(\boldsymbol{x},\xi_i^{(t)})$, $t=1,\dots,T$. Note that $\xi_i^{(t)}\in\Xi$ is a given element of the sample space, which here samples the indexes of the components of the sum.

Example 1: Consider a quadratic game defined as

$$\min_{x_i\in\Omega_i}\frac{1}{2}d_i x_i^2 + \pi_i^{\top}A(\boldsymbol{\xi})\boldsymbol{x}$$

where $\pi_i \in \mathbb{R}^T$, $A \in \mathbb{R}^{T\times N}$ and $d_i > 0$ for all $i \in \mathcal{I}$. Let π_i be such that $\pi_i = \mathrm{col}((\pi_{it})_{t=1}^T)$, $\pi_{it} \geq 0$ and $\sum_{t=1}^T \pi_{it} = 1$, i.e., π_i is a probability distribution over a finite sample set of size T. Therefore, $\mathbb{F}_i(\boldsymbol{x}) = d_i x_i + \sum_{t=1}^T \pi_{it} a_{ti}(\xi_i^{(t)}) = \sum_{t=1}^T h_{it}(\boldsymbol{x})$ where $h_{it}(\boldsymbol{x}) = \frac{1}{T} d_i x_i + \pi_{it} a_{ti}(\xi_i^{(t)}) = F_i^\xi(\boldsymbol{x}, \xi_i^{(t)})$ is a single-sample partial gradient as in (11). □

The approximation in Algorithm 2 still relies on the stochastic oracle in (11) that now, in sampling an element $\xi_i^{(t)}$, samples an entry $h_{it}(\cdot)$ of the sum in (14). The scheme is then based on selecting a suitable point to compute the full gradient only with a certain probability $p \in (0, 1]$ while at the general iteration k only the single component $h_{it_k}(\cdot) = F_i^\xi(\cdot, \xi_i^{(t_k)})$ is computed. The reason for this choice is that the single iteration k can be computationally light, i.e., computing only $h_{it_k}(\boldsymbol{x}^k)$ and $h_{it_k}(\boldsymbol{y}^k)$, while guaranteeing asymptotically a good approximation. Assumption 6 is therefore not required anymore. Each agent can use its own p_i but we use p for simplicity. Formally, we approximate (14) with

$$H_i(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{F}_i(\boldsymbol{y}) + h_{it}(\boldsymbol{x}) - h_{it}(\boldsymbol{y}), \tag{15}$$

where $\boldsymbol{y}$ is updated according to (7), hence only once in a while. Note that $\mathbb{E}[H_i(\boldsymbol{x}, \boldsymbol{y})] = \mathbb{F}_i(\boldsymbol{x})$ since Assumption 5 holds for $h_{it}(\cdot) = F_i^\xi(\cdot, \xi_i^{(t)})$, $t = 1, \dots, T$. The advantage of this approach is that the variance is reduced without extra assumptions on the stochastic error and without requiring an increasing batch. This implies that the computational burden is lower compared to Algorithm 1.

Remark 2: Variance reduction here is achieved by computing the full pseudogradient mapping. An explicit bound can be obtained using Lipschitz continuity [9] and it is based on the idea that a "good" point $\boldsymbol{y}$ has a lower variance $\mathbb{E}[\|H(\boldsymbol{x}, \boldsymbol{y}) - \mathbb{F}(\boldsymbol{x})\|^2]$ than $\mathbb{E}[\|F^\xi(\boldsymbol{x}, \xi) - \mathbb{F}(\boldsymbol{x})\|^2]$ [15]. □

Remark 3: The case of infinite sums is usually not considered for these problems, because the stochastic oracle would be biased, hence violating Assumption 5. A possible solution, proposed by [22] for Generative Adversarial Networks (a particular instance of SGNEPs [7], [23]), is to use a different finite sum at each epoch, i.e., after the fixed batch T has been entirely used. Otherwise, one could use a finite sum from the beginning, approximating already the continuous expected value with its discrete average. □

Let us indicate with $\mathcal{A}^\xi$ the operator $\bar{\mathcal{A}}$ in (9) where we replace $\mathbb{F}(\boldsymbol{x})$ with $F^\xi(\boldsymbol{x}, \boldsymbol{\xi})$. Then, the LEG algorithm in compact form reads as

$$\begin{aligned}
\bar{\boldsymbol{w}}^k &= \alpha \boldsymbol{w}^k + (1-\alpha)\boldsymbol{v}^k \\
\boldsymbol{w}^{k+\frac{1}{2}} &= (\mathrm{Id} + \Phi^{-1}\bar{\mathcal{B}})^{-1}(\bar{\boldsymbol{w}}^k - \Phi^{-1}\bar{\mathcal{A}}(\boldsymbol{v}^k)) \\
\boldsymbol{w}^{k+1} &= (\mathrm{Id} + \Phi^{-1}\bar{\mathcal{B}})^{-1}(\bar{\boldsymbol{w}}^k - \Phi^{-1}\bar{\mathcal{A}}(\boldsymbol{v}^k) \\
&\qquad - \Phi^{-1}\mathcal{A}^\xi(\boldsymbol{w}^{k+\frac{1}{2}}) + \Phi^{-1}\mathcal{A}^\xi(\boldsymbol{v}^k)) \\
\boldsymbol{v}^{k+1} &= \begin{cases} \boldsymbol{w}^{k+1} & \text{with probability } p \\ \boldsymbol{v}^k & \text{with probability } 1-p \end{cases}
\end{aligned} \tag{16}$$

where $\bar{\boldsymbol{w}}^k = \mathrm{col}(\bar{\boldsymbol{x}}^k, \boldsymbol{z}^k, \boldsymbol{\lambda}^k)$ and $\bar{\boldsymbol{v}}^k = \mathrm{col}(\bar{\boldsymbol{y}}^k, \boldsymbol{z}^k, \boldsymbol{\lambda}^k)$. The dual variables and auxiliary variables are not affected by the computational issues involving the pseudogradients and therefore are not updated with probability p but at every iteration (i.e., with $p = 1$). This makes sense also from the optimization point of view: the agents can play with how deep the gradient-descent step is but the constraints must be satisfied at every iteration. Moreover, when $p = 1$, (16) reduces to the deterministic EG scheme in (8). This scheme is called *loopless* because the full batch is not computed after a fixed number of iterations, hence there is no loop. The LEG scheme requires slightly stronger assumptions then the SEG algorithm.

Assumption 10: $\mathbb{F}$ in (5) is monotone and F^ξ in (11) is ℓ_ξ-Lipschitz continuous in mean for some $\ell_\xi > 0$. □

Assumption 10 implies $\ell_\mathbb{F}$-Lipschitz continuity of $\mathbb{F}$, with $\ell_\mathbb{F} \leq N \sum_{t=1}^T \ell_\xi$. Then, analogous results to Lemma 2 hold.

Lemma 3: Let Assumption 10 hold and let $\Phi \succ 0$. Then, the following holds for the operator $\bar{\mathcal{A}}$ in (9):

1) $\bar{\mathcal{A}}$ is monotone and $\mathcal{A}^\xi$ is $\ell_{\mathcal{A}^\xi}$-Lipschitz continuous.
2) $\Phi^{-1}\bar{\mathcal{A}}$ is monotone and $\Phi^{-1}\mathcal{A}^\xi$ is ℓ_Φ-Lipschitz continuous.

Proof: Analogous to [2, Lem. 2]. ■

The last assumption is on the parameters of Algorithm 2.

Assumption 11: The parameters in Algorithm 2 are such that $\alpha \in [0, 1), p \in (0, 1]$, $\|\Phi^{-1}\| = \frac{\sqrt{1-\alpha}}{\ell_{\bar{\mathcal{A}}}}\gamma$, and $\gamma \in (0, 1)$.

Theorem 2: Let Assumptions 1–5, 10, 11 hold. Then, the sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ generated by Algorithm 2 with the approximation (15) converges a.s. to a v-SGNE of the game in (1).

Proof: Analogously to Theorem 1, $(\mathrm{Id} + \Phi^{-1}\bar{\mathcal{B}})^{-1} = (\mathrm{Id} + \partial V(\boldsymbol{x}, \boldsymbol{\lambda}))^{-1} = \mathrm{prox}_V$. By [21, Prop. 12.26] and Assumption 10 we can hence use, mutatis mutandis, the same steps as [15, Lem. 1] to obtain a quasi-Fejer property for the sequence $\{\mathrm{col}(\boldsymbol{w}^k, \boldsymbol{v}^k)\}_{k\in\mathbb{N}}$ [15, Thm. 2], showing that $\{\boldsymbol{w}^k\}$ converges to a zero of $\bar{\mathcal{A}} + \bar{\mathcal{B}}$, i.e., to a v-SGNE. ■

IV. Numerical Simulations

To validate the analysis, we propose a numerical experiment on a network Cournot game that model the electricity market with capacity constraints [1], [24]. The simulations are performed with Matlab R2023b on a laptop with an Apple M1 chip featuring an 8-core CPU and 16 GB RAM.

To validate the analysis, we propose a numerical experiment on a network Cournot game that model the electricity market with capacity constraints [1], [24]. The simulations are performed with Matlab R2023b on a laptop with an Apple M1 chip featuring an 8-core CPU and 16 GB RAM.

We consider $m = 7$ markets where $N = 20$ companies sell energy. The decision variable x_i of each agent $i \in \mathcal{I}$ is the quantity of energy to be delivered to the n_i markets that the company is connected with, according to [1, Fig. 1]. The local constraints are given by the production limit, i.e., $\Omega_i = \{x_i \in \mathbb{R}^{n_i} : 0 < x_i < \beta_i\}$, where each component of β_i is randomly drawn from $[1, 1.5]$. The cost function of each agent is given by $\mathbb{J}_i(x_i, \boldsymbol{x}_{-i}) = c_i(x_i) - \mathbb{E}[P(\xi)^\top (A\boldsymbol{x})A_i x_i]$, where $c_i(x_i) = 1.5x_i + q_i$ is the production cost and q_i is a given constant. Each market j has a bounded capacity b_j, randomly drawn from $[0.5, 1]$. We consider then linear coupling constraints given by $A\boldsymbol{x} \leq b$ where $A = [A_1, \dots, A_N]$. The second part of the cost, which involves the market prices $P : \mathbb{R}^m \times \Xi \to \mathbb{R}^m$, is affected by the demand uncertainty and it is a linear function defined as $P(\xi) = \bar{P}(\xi) - DA\boldsymbol{x}$. Each component of $\bar{P}(\xi) = \mathrm{col}(\bar{P}_1(\xi), \dots, \bar{P}_7(\xi))$ is taken with a normal distribution with mean 3 and finite variance while the entries of D are randomly taken in $[0.5, 1]$. The graph $\mathcal{G}^\lambda$

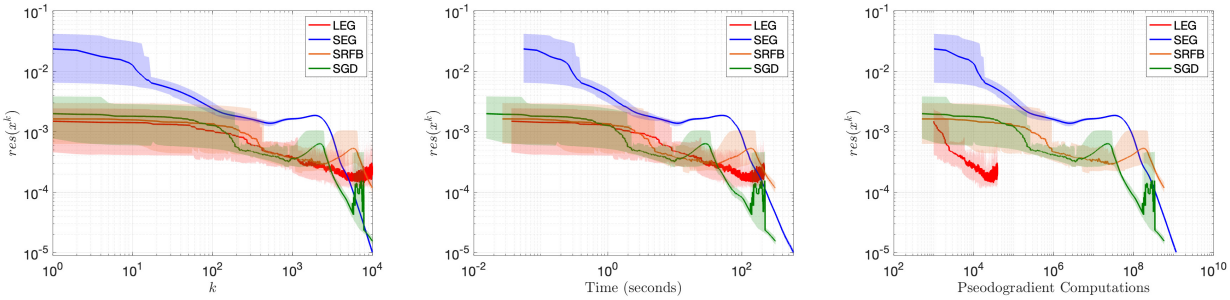

Fig. 1. Distance from an SGNE with respect to the number of iteration (left), computational time (center) and number of pseudogradient computations (right). The solid line shows the average performance while the shaded area represent the variance over 20 runs.

is a cycle graph with the addiction of the edges $(2, 15)$ and $(6, 13)$ [1]. In all the runs, the step-sizes are taken according to Assumptions 9 and 11 and [15].

We compare 20 instances of the two algorithms with the stochastic relaxed forward-backward (SRFB) algorithm [2, Alg. 1] and SGD [3, Alg. 1] which are single-step algorithms. To measure the distance from being an equilibrium, we use the residual $\mathrm{res}(\boldsymbol{x}^k)$. Fig. 1 shows on the left that the algorithms converge. The plot at the center shows the computational time, revealing, as expected, that LEG is less demanding than SEG while performing similarly to the single-step algorithms. This is confirmed by the amount of single-sample pseudogradient computation (Fig. 1, right) where $S_k = (k + 100)^{1.2}$, $k > 0$ for SEG, SRFB and SGD while $T = 1000$ for LEG. Different batch-sizes have been tested with similar results. The oscillating behavior in LEG is to be expected and it is common in single-sample approximation schemes [5], [15], [24]. Note moreover that this scheme is guaranteed to converge (and reduce the variance) only for finite sums while our stochastic example does not satisfy this requirement.

V. Conclusion

Variance-reduced extragradient methods can be applied to stochastic generalized Nash equilibrium problems and converge under weak monotonicity assumptions. Different approximation schemes can be used as long as they guarantee to asymptotically reduce the variance of the approximation error. In particular, a new approximation scheme has been introduced for the particular case of finite sums. Numerical simulations suggest that this scheme can work also for infinite sum, however a deeper analysis of this case is needed and left for future work.

References

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

[2] B. Franci and S. Grammatico, "Stochastic generalized Nash equilibrium-seeking in merely monotone games," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 3905–3919, 2021.

[3] ——, "A distributed forward–backward algorithm for stochastic generalized Nash equilibrium seeking," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5467–5473, 2020.

[4] G. Belgioioso, P. Yi, S. Grammatico, and L. Pavel, "Distributed generalized Nash equilibrium seeking: An operator-theoretic perspective," *IEEE Control Systems Magazine*, vol. 42, no. 4, pp. 87–102, 2022.

[5] F. Fabiani and B. Franci, "A stochastic generalized Nash equilibrium model for platforms competition in the ride-hail market," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4455–4460.

[6] G. Belgioioso, W. Ananduta, S. Grammatico, and C. Ocampo-Martinez, "Operationally-safe peer-to-peer energy trading in distribution grids: A game-theoretic market-clearing mechanism," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2897–2907, 2022.

[7] B. Franci and S. Grammatico, "Training generative adversarial networks via stochastic Nash games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1319–1328, 2021.

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

[9] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[10] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in neural information processing systems*, vol. 26, 2013.

[11] 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.

[12] R. I. Boţ, P. Mertikopoulos, M. Staudigl, and P. T. Vuong, "Minibatch forward-backward-forward methods for solving stochastic variational inequalities," *Stochastic Systems*, vol. 11, no. 2, pp. 112–139, 2021.

[13] B. Franci and S. Grammatico, "Distributed projected–reflected–gradient algorithms for stochastic generalized Nash equilibrium problems," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 369–374.

[14] G. M. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[15] A. Alacaoglu and Y. Malitsky, "Stochastic variance reduction for variational inequality methods," in *Conference on Learning Theory*. PMLR, 2022, pp. 778–816.

[16] 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.

[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] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

[19] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[20] R. T. Rockafellar, "Monotone operators associated with saddle-functions and minimax problems," *Nonlinear functional analysis*, vol. 18, no. part 1, pp. 397–407, 1970.

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

[22] T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien, "Reducing noise in gan training with variance reduced extragradient," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] B. Franci and S. Grammatico, "A game–theoretic approach for generative adversarial networks," in *2020 59th IEEE conference on decision and control (CDC)*. IEEE, 2020, pp. 1646–1651.

[24] 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.