ADAPTIVE LEARNING FOR STOCHASTIC GENERALIZED NASH EQUILIBRIUM PROBLEMS

Chung-Kai Yu, Mihaela van der Schaar, and Ali H. Sayed

Department of Electrical Engineering, University of California, Los Angeles

ABSTRACT

This work examines a stochastic formulation of the generalized Nash equilibrium problem (GNEP) where agents are subject to randomness in the environment of unknown statistical distribution. Three stochastic gradient strategies are developed by relying on a penalty-based approach where the constrained GNEP formulation is replaced by a penalized unconstrained formulation. It is shown that this penalty solution is able to approach the Nash equilibrium in a stable manner within $O(\mu)$, for small step-size values $\mu$. The operation of the algorithms is illustrated by considering the Cournot competition problem.

Index Terms— Adaptive learning, generalized Nash equilibrium, penalized approximation, diffusion learning.

1. INTRODUCTION AND RELATED WORK

The generalized Nash equilibrium problem (GNEP) refers to a setting where each agent in a collection of agents seeks to minimize its own cost function subject to certain constraints and where both the cost function and the constraints are generally dependent on the actions selected by the other agents [1–4]. In these types of problems, the Nash equilibrium is a desired and stable solution since at the Nash equilibrium no agent can benefit by unilaterally deviating from the solution. In general, GNEP formulations do not admit closed-form solutions. It is therefore common to appeal to penalty-based formulations where the original cost function is modified by including a penalty term. The purpose of the penalty term is to assign large penalties to deviations from the constraints [5–8].

In most prior works, the individual cost functions are assumed to be deterministic and agents are able to compute exactly their gradient vectors [4, 6, 9]. However, when the agents are subject to randomness in the environment, it is customary to define the cost functions in terms of expectations of certain loss functions. The expectation operations are in relation to the distribution of the random data. Since this distribution is rarely known beforehand, it becomes impossible for the agents to compute the exact gradients of their cost functions to permit gradient-descent implementations. Instead, the agents need to resort to stochastic gradient implementations where the actual gradient vectors are replaced by approximations. One stochastic implementation along these lines is considered in [10] albeit with a vanishing step-size parameter and without constraints. The use of step-sizes that decay to zero is problematic in scenarios that require continuous adaptation and learning. For example, in nonstationary environments, the Nash equilibrium may drift with time due to changes in the statistical distribution of the data. When the step-size approaches zero, adaptation stops and the resulting stochastic gradient algorithm will lose its ability to track the drift.

For this reason, we focus in this work on the use of constant step-sizes to enable adaptation and learning. When this is done, gradient noise seeps into the operation of the algorithm. By gradient noise we mean the difference between the true gradient vector and its approximation. In decaying step-size implementations, this gradient noise component is annihilated over time by the diminishing step-size parameter at the expense of a deteriorating tracking performance. In contrast, in the constant step-size implementation, the gradient noise process is persistently present in the operation of the algorithm. One main challenge in the analysis is to establish that the stochastic-gradient implementation is able to keep the influence of gradient noise under check and to deliver an accurate estimation of the minimizer. Arriving at these conclusions for networked agents is the key contribution of this work. In the simulations section, we illustrate the theoretical results and apply the algorithms to the Cournot competition problem, which is widely used in applications such as economic trading with geographical considerations, power management over smart grids, and resource allocation [11–13].

Notation: We use lowercase letters to denote vectors and scalars, uppercase letters for matrices, plain letters for deterministic variables, and boldface letters for random variables.

2. PROBLEM FORMULATION

Consider a connected network of $N$ agents indexed by the set $\mathcal{N} = \{1, \ldots, N\}$. The neighborhood of each agent $k$, denoted by $\mathcal{N}_k$, includes agent $k$ and the neighboring agents connected to $k$. We denote the action of each agent by a vector $w_k \in \mathbb{R}^{M_k}$ and associate with $k$ an individual convex cost function denoted by $J_k(\cdot)$. The argument of $J_k(\cdot)$ does not depend solely on $w_k$ but also on the action vectors of the neighboring agents. Let us collect the actions of all agents in $\mathcal{N}_k$ into the block vector $w^k = \text{col}\{w_{\ell} : \ell \in \mathcal{N}_k\} \in \mathbb{R}^{M^k}$, and the actions of all agents in $\mathcal{N}$ into $w = \text{col}\{w_1, \ldots, w_N\} \in \mathbb{R}^M$ where

$$M^k \triangleq \sum_{\ell \in \mathcal{N}_k} M_\ell, \quad M \triangleq \sum_{k=1}^N M_k$$ (1)

We consider that the action of each agent $k$ should satisfy a set of local constraints $\{g_{k,q}(w^k) \leq 0, q = 1, \ldots, L_k\}$. The local constraint functions $\{g_{k,q}(w^k)\}$ are known to agent $k$ and assumed to be twice-differentiable and convex in $w^k$. We further assume that the constraints are shared by the neighbors, i.e., if the argument of $g_{k,q}(w^k)$ contains the action of some neighbor $\ell$, then agent $k$ is subject to the same constraint $g_{k,q}(w^k) \leq 0$. Figure 1 illustrates an example showing the dependence of the cost functions and the shared constraint within agents 3 and 5 over a network topology. We note that while there is no direct link between agents 2 and 4, the action vectors $w_2$ and $w_4$ are coupled through the intermediate agent 3. Therefore, in general, the actions of agents are affected explicitly...
by the neighbors and also implicitly by other agents in the network. This scenario is common in applications [5, 6, 9, 14].

Each agent $k$ then seeks an optimal action vector that solves the following constrained optimization problem:

$$
\begin{align*}
\min_{w_k} & \quad J_k(w^k) \\
\text{subject to} & \quad g_{k,q}(w^k) \leq 0, \quad q = 1, \ldots, L_k
\end{align*}
$$

(2)

which is known as the generalized Nash equilibrium problem (GNEP).

For convenience, we shall collect all distinct individual constraints into a global set denoted by \( \{g_q(w) \leq 0, q = 1, \ldots, L\} \) by removing the repeated shared constraints. We assume that the GNEP in (2) is feasible for each agent, meaning that the set \( \{w|g_q(w) \leq 0, q = 1, \ldots, L\} \) is non-empty. In this work, we focus on proposing distributed learning strategies by which agents can adaptively learn to solve (2) using local observations of the actions of neighboring agents. Although the treatment can be extended to more general cases, we illustrate the main construction and results by focusing on the next example.

Example 1. We partitioned the state space \( \mathcal{N} \) into two disjoint sets \( \mathcal{N}_1 \) and \( \mathcal{N}_2 \). The agents can only observe which set their state belongs to (i.e., what \( \mathcal{N}_i \) their state is in) but not the state itself. This is a common case in many distributed optimization problems [5].

Next note that the gradient vector of $J_k(w)$ is symmetric, we have

$$
\nabla w_k J_k(w^k) = \sum_{\ell \in \mathcal{N}_k} 2b^k_{\ell k} w_{\ell} + b_k.
$$

(4)

If we collect these individual gradient vectors into

$$
F(w) \triangleq \text{col}\{\nabla w_1 J_1(w^1), \ldots, \nabla w_L J_L(w^L)\}
$$

then this block column vector satisfies

$$
F(w) = Bw + b
$$

(6)

where $b = \text{col}\{b_1, \ldots, b_N\}$ and

$$
B \triangleq \begin{bmatrix}
2B_{11} & \cdots & 2B_{1N} \\
\vdots & \ddots & \vdots \\
2B_{N1} & \cdots & 2B_{NN}
\end{bmatrix} \in \mathbb{R}^{M \times M}
$$

(7)

To continue, we introduce the following condition.

Condition 1. There exists a positive constant $\nu$ such that $B \geq \nu I$, i.e., for any $M \times 1$ vector $a$ we have

$$
a^T (B - \nu I) a \geq 0 \iff a^T B a \geq \nu \|a\|^2
$$

(8)

Note that since $B$ is not necessarily symmetric, we know from [15, pp. 259] that Condition 1 holds if, and only if, the symmetric part of $B$ satisfies:

$$
\frac{1}{2} (B + B^T) \geq \nu I
$$

(9)

It follows from this condition that the largest singular value of $B$, denoted by $\sigma_{\text{max}}$, is greater than or equal to $\nu$ since

$$
\sigma_{\text{max}} = \|B\| \geq \left\|\frac{1}{2} (B + B^T)\right\| \geq \nu
$$

(10)

Under Condition 1, it is easy to verify that for any two action profiles $w = w^0$ and $w = w^*$ we have

$$
(w^0 - w^*)^T [F(w^0) - F(w^*)] \geq \nu \|w^0 - w^*\|^2
$$

(11)

which means that the mapping $F(w) : \mathbb{R}^M \rightarrow \mathbb{R}^M$ is strongly monotone [16].

3. STOCHASTIC PENALTY-BASED LEARNING

3.1. Penalty Approximation for Coupled Constraints

Solving the constrained optimization problem (2) is generally demanding and may not admit a closed-form solution. Alternatively, in this work, we resort to a penalty-based approach to replace the original problem by an unconstrained optimization problem and then show that the solution to the approximate problem tends asymptotically with the penalty parameter to the desired solution to (2). More importantly, we will show that the penalty-based approach enables the agents to employ adaptive learning strategies, which in turn endow the agents with the ability to track variations in the location of the Nash equilibrium due to changes that may occur in the constraint conditions or cost measures.

The main motivation for penalty methods is to assign a large penalty weight whenever constraints are violated and a smaller or zero weight when the constraints are satisfied [7, 17, 18]. More specifically, problem (2) is replaced by the following unconstrained formulation:

$$
\min_{w_k} \quad J_k(w^k) + \rho p_k(w^k)
$$

(12)

where $\rho \geq 0$ is a penalty parameter, $p_k(w)$ denotes the penalty function for agent $k$ and is assumed to be of the following aggregate form, with one penalty factor applied to each constraint:

$$
p_k(w^k) = \sum_{q=1}^{L_k} \theta_g (g_{k,q}(w^k))
$$

(13)
where \( \theta(x) \) is a convex function. The penalty factor \( \theta(\cdot) \) returns zero value if the constraint is satisfied, i.e., when \( g_k(x) = 0 \), and introduces a large positive penalty if the constraint is violated, i.e., when \( g_k(x) > 0 \). While there exist numerous penalty-type functions in the literature, e.g., \( \gamma \)-norm [19], exponential and shifted logarithmic functions [20], in this work we assume that \( \theta(x) \) is chosen to be continuous, convex, nondecreasing, and twice-differentiable — see [7, 21] and (43) for an example.

We denote the penalized cost by
\[
J^P_k(w^p) \triangleq J_k(w^p) + \rho p_k(w^p)
\]
(14)
An action profile \( w^p \) that minimizes simultaneously all penalized costs \( \{J^P_k(w^p)\} \) is called a Nash equilibrium for the penalized formulation (12). By the results in [7, pp. 3930], we get that the distance between the solutions to problem (2) and problem (12) can be made arbitrarily small by choosing \( \rho \) appropriately. Consequently, the resulting Nash equilibrium for the unconstrained problem (12) can be arbitrarily close to a Nash equilibrium for the GNEP in (2).

Since the \( \{J^P_k(w^p)\} \) are convex functions, it follows from results in [9, pp. 522] that a Nash equilibrium exists. The following theorem ensures its uniqueness.

**Theorem 1. (Uniqueness):** Under Condition 1 and for any convex choice of \( \theta(x) \), problem (12) has a unique Nash equilibrium \( w^* \), and it satisfies
\[
F^P(w^*) = Bw^* + b + \rho \nabla w^T \nabla p(w^*) = 0
\]
(15)
where
\[
F^P(w) \triangleq \text{col}\{\nabla w_1^T J^P_1(w^p), ..., \nabla w_N^T J^P_N(w^p)\}
\]
(16)
\[
p(w) \triangleq \sum_{q=1}^Q \theta(g_q(w))
\]
(17)
We note that it holds that
\[
\nabla w^T \nabla p(w) = \text{col}\{\nabla w_1^T p_1(w^p), ..., \nabla w_N^T p_N(w^p)\}
\]
(18)

### 3.2. Stochastic Learning Dynamics

Again, a closed form solution to problem (12) is not generally possible. If this were possible, then agents could learn \( w^p \) given knowledge of the other agents’ actions; this solution method would lead to the best response dynamics [22]. Since this approach is rarely applicable, agents can instead appeal to learning strategies where they gradually approach the desired \( w^p \) through successive estimation from streaming data. For example, one well-known solution is to use the gradient information \( \nabla w^T J^P_k(w^p) \) to update the agents’ actions at discrete-time instants \( t \) [23–25]. However, knowledge of the mean quantities \( \{B_{k,i}^b, b_{k,i}\} \) is required to compute \( \nabla w^T J_k(w^p) \).

When the statistics of \( B_{k,i}^b \) and \( b_{k,i} \) are unavailable, we need to resort to instantaneous realizations of these random variables, which we denote by \( \tilde{B}_{k,i}^b \) and \( \tilde{b}_{k,i} \).

We assume these realizations are independent over both \( k, \ell \), and \( i \). Using the realizations \( \{\tilde{B}_{k,i}^b, \tilde{b}_{k,i}\} \), agents can update their actions at each time instant \( i \) by employing the following localized stochastic-gradient rule:
\[
w_{k,i} = w_{k,i-1} - \mu \left( \sum_{\ell \in N_k} 2B_{\ell,i}^b w_{\ell,i-1} + b_{k,i} \right) - \rho \nabla w^T \nabla p_k(w_{k,i-1})
\]
(19)
where \( \mu \) is the step-size. Alternatively, motivated by the arguments from [7] and [26], one can implement (19) incrementally by using a two-step learning strategy to improve the individual and penalty costs separately. For example, agent \( k \) can use an Adapt-then-Penalize (ATP) diffusion learning strategy to update first the iterate along the negative gradient direction of \( J_k(\cdot) \) and then correct along the gradient of the penalty term:

\[
(\text{ATP}) \begin{cases}
\psi_{k,i} = w_{k,i-1} - \mu \left( \sum_{\ell \in N_k} 2B_{\ell,i}^b w_{\ell,i-1} + b_{k,i} \right) \\
w_{k,i} = \psi_{k,i} - \rho \nabla w^T \nabla p_k(\psi_{k,i})
\end{cases}
\]
(20)
Agents can also switch the order of these two steps and use a Penalize-then-Adapt (PTA) diffusion learning strategy:

\[
(\text{PTA}) \begin{cases}
\psi_{k,i} = w_{k,i-1} - \mu \rho \nabla w^T \nabla p_k(w_{k,i-1}) \\
w_{k,i} = \psi_{k,i} - \mu \left( \sum_{\ell \in N_k} 2B_{\ell,i}^b \psi_{\ell,i} + b_{k,i} \right)
\end{cases}
\]
(22)

Observe that we are denoting the weight iterates in boldface since they are random quantities due to the randomness in the realizations \( \{B_{k,i}^b, b_{k,i}\} \).

### 4. PERFORMANCE ANALYSIS

We now examine the convergence and stability properties of these stochastic algorithms. In particular, we examine how close their limiting point gets to the equilibrium point, \( w^* \). To continue, we introduce the following condition on the penalty function. This condition is not restrictive since the choice of the penalty function is under the designer’s control.

**Condition 2.** Consider two arbitrary block vectors \( w^o \) and \( w^* \) collecting all agents’ actions. For each \( k \), and the corresponding \( w_{k,i}^o \) and \( w_{k,i}^* \), the gradient vector \( \nabla w^T \nabla p_k(\cdot) \) is assumed to satisfy the Lipschitz condition:
\[
\left\| \nabla w^T \nabla p_k(w_{k,i}^o) - \nabla w^T \nabla p_k(w_{k,i}^*) \right\| \leq \gamma_k \| w_{k,i}^o - w_{k,i}^* \|
\]
(24)
where \( \gamma_k \) is a positive constant.

### 4.1. Stochastic Gradient Dynamics

We can describe the evolution of the dynamics of the first algorithm (19) in the following manner:
\[
w_i = w_{i-1} - \mu (B_i w_{i-1} + b_i) - \mu \rho \nabla w^T \nabla p(w_{i-1})
\]
(25)

In terms of the aggregate quantities \( w_i \triangleq \text{col}\{w_{1,i}, ..., w_{N,i}\}, b_i \triangleq \text{col}\{b_{11,i}, ..., b_{NN,i}\} \), and
\[
B_i \triangleq \begin{bmatrix}
2B_{11,i}^1 & \cdots & 2B_{1N,i}^1 \\
\vdots & \ddots & \vdots \\
2B_{N1,i}^N & \cdots & 2B_{NN,i}^N
\end{bmatrix}
\]
(26)

Subtracting \( w^* \) from both sides of (25), introducing the error vector \( \bar{w}_i = w^* - w_i \), and using (15) we find that
\[
\bar{w}_i = \bar{w}_{i-1} - \mu [F^P(w^*) - F^P(w_{i-1})] + \mu s_i(w_{i-1})
\]
(27)
where \( s_i(w) = -\partial_i \bar{w} - \tilde{b}_i \) is the gradient noise, \( \tilde{B}_i = B - B_i \), and \( \tilde{b}_i = b - b_i \). From the assumed independence of \( B_i \) and \( b_i \), and \( w_{i-1} \), it can be verified that
\[
E[s_i(w_{i-1})] = 0
\]
(28)
\[
E[||s_i(w_{i-1})||^2] \leq \alpha E[||\bar{w}_{i-1}||^2 + \beta
\]
(29)

for some constants \( \alpha \) and \( \beta \). The proof of the following result is omitted for brevity.
Theorem 2. (Mean-square-error stability) For the stochastic gradient implementation (19), if the step-size $\mu$ satisfies
\[
0 < \mu < \frac{2\nu}{\sigma_{\text{max}} + \rho^2 \delta_p^2 + \alpha}
\]  
(30)
where $\alpha > 0$ is a constant parameter, $\delta_p \triangleq \sqrt{\sum_{k=1}^{N} \gamma_k^2}$, and $\sigma_{\text{max}}$ is the largest singular value of $B$, then it holds that
\[
\lim_{i \to \infty} \sup_{i} \mathbb{E}[\|\tilde{w}_i\|^2] = O(\mu)
\]  
(31)

4.2. Stochastic ATP and PTA Strategies

We can aggregate the ATP and PTA strategies in (20)–(21) and (22)–(23), respectively, across all agents into the following unified description:
\[
w_i = \psi_{i-1} - c_1 \mu \nabla \psi_{i-1} (\psi_{i-1})
\]
(32)
\[
\phi_i = w_i - \mu (B, w_i + b_i)
\]
(33)
\[
\tilde{\psi}_i = \phi_i - c_2 \mu \psi_{i-1} (\phi_i)
\]
(34)
for some constants $(c_1, c_2)$. By setting $(c_1, c_2) = (0, 1)$ we get the ATP recursions (20)–(21) while for $(c_1, c_2) = (1, 0)$ we obtain the PTA recursions (22)–(23). The following result establishes that without gradient noise, the deterministic version of recursions (32)–(34) converges to a unique fixed point.

Theorem 3. (Unique fixed point) Without gradient noise, the mapping from $\psi_{i-1}$ to $\psi_i$ in recursions (32)–(34) converges to a unique fixed point, denoted by $\psi^\infty$, for small step-sizes that satisfy:
\[
0 < \mu < \frac{2\nu}{\sigma_{\text{max}} + \rho^2 \delta_p^2}
\]  
(35)

Let us denote by $w^\infty$ the fixed point corresponding to the deterministic variable $w_i$, and introduce the fixed-point error $\tilde{w}_i^\infty = w^\infty - w_i$ for the stochastic variable $w_i$ with gradient noise. The following theorem shows that the variance of this error is bounded.

Theorem 4. (Bounded MSE) For the stochastic recursion (32)–(34), if the step-size $\mu$ satisfies
\[
0 < \mu < \frac{2\nu}{\sigma_{\text{max}} + \rho^2 \delta_p^2 + \alpha}
\]  
(36)
then it holds that
\[
\lim_{i \to \infty} \sup_{i} \mathbb{E}[\|\tilde{w}_i^\infty\|^2] = O(\mu)
\]  
(37)

We note that condition (36) for the stochastic recursion implies condition (35) for the deterministic case. Therefore, any step-size $\mu$ satisfying (36) ensures the existence of the fixed point $w^\infty$. Furthermore, comparing (36) with (30) we observe that the stochastic ATP and PTA learning strategies allow to use a larger step-size, which can assist with faster convergence performance. However, the fixed point $w^\infty$ would be different from the Nash equilibrium, $w^*$. In the following theorem, we examine the bias $\tilde{w} = w^* - w^\infty$. We show that for small $\mu$, the norm of the bias is asymptotically upper bounded by $O(\mu)$.

Theorem 5. (Small bias) For sufficiently small step-size $\mu$ satisfying the following condition:
\[
0 < \mu < \frac{2\nu}{\sigma_{\text{max}}^2 + 2\rho^2 \delta_p^2}
\]  
(38)
it holds that
\[
\lim_{i \to \infty} \sup_{i} \mathbb{E}[\|w^* - w_i\|^2] \leq O(\mu) + O(\mu^2 \rho^2)
\]  
(39)

5. SIMULATION RESULTS

In this section, we consider the Cournot competition problem [11, 12] with stochastic parameters. Thus, consider a network with $N = 10$ factories and $L = 3$ markets connected to the factories as shown in Fig. 2. Each factory $k$ needs to determine a continuous quantity of products to be produced and delivered to each connected market, which is defined as the action of factory $k$ denoted by $w_k = [w_k(1), ..., w_k(M_k)]^T$ where we assumed $M_k$ markets are connected to factory $k$. The individual cost function of each factory $k$ is defined as
\[
J_k(w^k) = \mathbb{E} \left( C_{k,i}(w_k) - \sum_{u \subseteq \ell} w_k(u) \cdot P_{\ell,i}(r_\ell) \right)
\]  
(40)
where
\[
C_{k,i}(w_k) = (\tilde{\alpha}_k + \psi_k^\delta) \left( \sum_{n=1}^{M_k} w_k(n)^2 \right)
\]  
(41)
is a random quadratic production cost function to generate $\sum_{n=1}^{M_k} w_k(n)$ amount of products for $\tilde{\alpha}_k > 0$ and with some random disturbance $\psi_{k,\ell}^\delta$ and
\[
P_{\ell,i}(r_\ell) = q_\ell - (s_\ell + \psi_{\ell,i}^\delta) r_\ell
\]  
(42)
is the price of products in market $\ell$ and $r_\ell$ is the total amount of products sold in market $\ell$, e.g., $r_2 = w_3(2) + w_4(1) + w_5(1) + w_6(2)$ and so on. In (40), we use the notation $u \subseteq \ell$ to represent that $w_k(u)$ is the quantity of factory $k$ sold in market $\ell$. We set $\tilde{\alpha}_k = 1$, $q_\ell = 3$, and $s_\ell = 1$ for all $k$ and $\ell$. The random variables $\psi_{k,\ell}^\delta$ and $\psi_{\ell,i}^\delta$ for all $k$, $\ell$ and $i$ are assumed to be independent and follow normal distributions with zero mean and standard deviation 10. Each market $\ell$ has an upper bound of 0.5 for the capacity of products, i.e., $r_\ell \leq 0.5$. The penalty function $\theta(x)$ is chosen as [7]:
\[
\theta(x) \triangleq \max \left\{ 0, \frac{x^3}{\sqrt{x^2 + \eta^2}} \right\}
\]  
(43)
with $\eta = 0.015$. We set the penalty parameter $\rho = 1000$ and the step-size $\mu = 0.001$. In Fig. 3, we simulate the learning curves for the stochastic gradient, ATP, and PTA strategies. To be specific, we compare the mean-squared error $\mathbb{E}[\|w^* - w_i\|^2]$ in these three cases. We observe from the figure that while the three strategies exhibit approximately the same steady-state performance, the stochastic ATP and PTA strategies have better convergence performance in the transient phase.
6. REFERENCES


