Fully Distributed Nash Equilibrium Seeking Algorithm with Quantization Effects in a Directed Graph

Xinpei Rao and Wenying Xu(✉)

School of Mathematics, Southeast University, Nanjing 21189, China
wyxu@seu.edu.cn

Abstract. This paper considers a distributed Nash equilibrium (NE) seeking problem with limited communication capacity. A fully distributed NE seeking algorithm is proposed with quantized information, including projected pseudo-gradient dynamics, distributed decision estimation and adaptive quantization. Based on a proposed encoder-decoder scheme, the algorithm is able to converge to the theoretical NE without any errors caused by quantization. Finally, a numerical simulation is provided to validate the effectiveness of our algorithm.

Keywords: Distributed Nash equilibrium seeking · Quantization communication · Projected pseudo-gradient dynamics · Consensus

1 Introduction

In recent years, a noncooperative game problem has been a hot issue due to its widespread applications, including but not limited to the modelling of some engineering, economic, and social problems [1,2]. In a game, each player can be considered as a selfish decision maker that only aims optimizing its individual but inter-dependent payoff function. Nash equilibrium (NE) computation is one key issue of the noncooperative game. Note that most of traditional NE seeking algorithms depend on a central node, which broadcasts information to all players involved in a game, however, such a node could not exist in practice. Thus, distributed algorithms have been proposed, which make it possible to calculate NE only through communication with neighbors. With the popularization of large-scale networks, the distributed NE seeking problem has become a new thriving research topic [3–11].

In a non-cooperative game, the objective function of each player not just depends on its own decision, but also other players' decisions, some of which may be not directly obtained. In order to make up for the necessary decisions, each player estimates the decisions of other players by exchanging information with neighbors. This implies that communication plays an increasingly vital role in distributed algorithms. The increase in number of players leads to a great deal of information generated that needs to be transmitted through the network. Note

Z. Ren et al. (Eds.): *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, LNEE 934, pp. 593–604, 2023.
https://doi.org/10.1007/978-981-19-3998-3_57

that the network bandwidth is limited in practice and may not be capable of transiting such much information [12–15]. As such, data quantization is hardly avoided, and becomes one of the mostly investigated network-induced effects.

In general, the quantization effects are likely to degrade the algorithm performance since the quantization inevitably brings some errors to the design and implementation of NE algorithms [16]. In this case, how to guarantee the convergence of NE becomes rather challenging. As we know, there have been some initial works on distributed optimization with quantized information [17–21]. Unfortunately, to the best of the authors' knowledge, distributed NE seeking problem for noncooperative games subject to quantization has not gained adequate research attention yet. The recent work [22] has discussed impacts of quantization on discrete-time gradient-based Nash equilibrium seeking algorithm, where each player is assumed to have capacity of broadcasting its quantized information to all the other players in the game. Such an algorithm could not be appropriate for large-scale games, which motivates us to further study distributed NE seeking with quantized information.

In this paper, we concentrate on a distributed NE seeking problem with limited bandwidth constraints, where each player exchanges quantized information with its neighbors. A encoder-decoder scheme is well-designed for each player, and a effective NE seeking algorithm is designed with quantized information. The proposed distributed algorithm is proved to guarantee the convergence of NE without any errors caused by quantization.

The rest of the paper is organized as follows: the formulation of the problem is presented in Sect. 2. In Sect. 3, we design an adaptive uniform quantizer, and the distributed NE seeking strategy with quantization is proposed. Simulation results are performed in Sect. 4 to demonstrate the effectiveness of the proposed algorithm and conclusions of the paper are provided in Sect. 5.

Notation. The notation used here is fairly standard except where otherwise stated. $\mathbb{N}$ denotes the set of natural numbers containing 0. $\mathbb{R}$ and $\mathbb{R}^n$ are the set of real numbers and n dimensional Euclidean space, respectively; $\mathbf{0}_n$ and $\mathbf{1}_n$ respectively represent n-dimension vectors with all elements being 0 and 1; I_n the n-dimension identity matrix, and the subscripts could be omitted if no ambiguity. Given m vectors $x_1, \cdots, x_m, \mathcal{N} = \{1, 2, \cdots, m\}$, $x := \text{col}\left((x_i)_{i\in\mathcal{N}}\right) = \left[x_1^\top \dots x_m^\top\right]^\top$ and $x_{-i} = \text{col}\left((x_j)_{j\in\mathcal{N}\setminus\{i\}}\right) = \left[x_1^\top, \cdots, x_{i-1}^\top, x_{i+1}^\top, \cdots, x_m^\top\right]^\top$. The Euclidean vector norm is represented by $\|\cdot\|$. For a given positive number $a \in \mathbb{R}$, $\lceil a \rceil$ stands for the smallest integer greater than or equal to a. Given a matrix $A \in \mathbb{R}^{m\times n}$, $[A]_{i,j}$ stands for its (i, j) and $A^\top$ represents its transpose, the second superscript will be omitted when $n = 1$. Let $\sigma(A)$ denote its singular, and $\|A\| = \sigma_{max}(A)$ stand for its 2-induced matrix norm, where $\sigma_{\max}(A)$ represent its maximum singular value. For a square matrix $A \in \mathbb{R}^{n\times n}$, let $A > 0$ denote that it is a symmetric positive definite matrix. For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, $\lambda = \{\lambda_1, \cdots, \lambda_m\}$ denotes the set of eigenvalues of matrix A, and as the subscript increases, the corresponding eigenvalues also increase, i.e., $\lambda_{\min}(A) = \lambda_1(A)$, $\lambda_{\max}(A) = \lambda_m(A)$. Let $\text{diag}\,(A_1, \dots, A_N)$ denote the block diagonal matrix with $A_1, \ldots, A_N$ on the main diagonal. Given matrices A and B, $A \otimes B$ stands for the Kronecker product.

For a differentiable function $f : \mathbb{R}^m \to \mathbb{R}$, its gradient is represented by $\nabla_x f(x)$. Given a mapping $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^m$, it is called θ-Lipschitz continuous, if and only if, for some $\theta > 0$, $\|\mathcal{F}(x) - \mathcal{F}(y)\| \leq \theta\|x - y\|$, for any $x, y \in \mathbb{R}^n$, and it is called (μ-strongly) monotone if(for some $\mu > 0$), $(\mathcal{F}(x) - \mathcal{F}(y))^\top (x - y) \geq 0 \left(\geq \mu\|x - y\|^2\right)$, for any $x, y \in \mathbb{R}^n$. Let $\mathrm{proj}_S : \mathbb{R}^n \to \Omega$ denote the Euclidean projection onto a closed convex set Ω, i.e., $\mathrm{proj}_\Omega(x) := \mathrm{argmin}_{y \in \Omega} \|y - x\|$.

2 Problem State

Consider the noncooperative game $G = \{\mathcal{V}, \mathcal{J}, x\}$, where $\mathcal{V} := \{1, \cdots, N\}$ represents players involved in the game. Let $\mathcal{J} := (J_1, J_2, \cdots, J_N)$, where J_i denotes the local payoff differentiable function of each player $i \in \mathcal{V}$. Denote $x = \mathrm{col}(x_i) \in \Omega \subseteq \mathbb{R}^n$ as the decision profile, i.e. the agents' decisions, where for $\forall i \in \mathcal{V}$, $x_i \in \Omega_i \subseteq \mathbb{R}^{n_i}$ is its decision, Ω_i represents its local feasible decision set, $n = \sum_{i=1}^N n_i$ and $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_N$ denotes the overall action space. In the setup of the noncooperative game, the local payoff function J_i can be written as $J_i(x_i, x_{-i})$, where $x_{-i} = \mathrm{col}\left((x_j)_{j \in \mathcal{V} \setminus \{i\}}\right) \in \mathbb{R}^{n-n_i}$ stands for the decision profile of all other players' decisions.

Then the game is represented by the inter-dependent optimization problems:

$$\forall i \in \mathcal{V}: \quad \underset{y_i \in \Omega_i}{\mathrm{argmin}}\ J_i(y_i, x_{-i}) \tag{1}$$

Definition 1. *A set of strategies* $x^* = \mathrm{col}\left((x_i^*)_{i \in \mathcal{N}}\right) \in \Omega$ *is a Nash equilibrium, if and only if, for all* $i \in \mathcal{V}$ *:*

$$J_i\left(x_i^*, x_{-i}^*\right) \leq \inf_{y_i \in \Omega_i} J_i\left(y_i, x_{-i}^*\right)$$

In this paper, we consider a partial-decision information scenario, where each agent i has no access to all other players information and computes only by locally exchanging data with their neighbors over a directed communication network $\mathcal{G}(\mathcal{V}, \mathcal{E})$. If agent i can send information to agent j, then $(i, j) \in \mathcal{E}$ and agent j belongs to agent i's out-neighbor set $\mathcal{N}^i = \{j \mid (i, j) \in \mathcal{E}\}$. Similarly, we can define agent i's in-neighbor set $\mathcal{N}_i = \{j \mid (j, i) \in \mathcal{E}\}$. Let $W \in \mathbb{R}^{N \times N}$ denote the weighted adjacency matrix of $\mathcal{G}$ and $w_{i,j} := [W]_{i,j}$, with $w_{i,j} > 0$ if $(i, j) \in \mathcal{E}, w_{i,j} = 0$ otherwise.

Our target is to propose a distributed algorithm with quantization scheme that allows players to find a NE limited by finite communication capacity between players.

To start with, we propose some common but significant regularity assumptions to facilitate the analysis of convergence.

Assumption 1 *(Regularity and convexity): For each player* $i \in \mathcal{V}$*, the set* Ω_i *is non-empty compact and convex; given* x_{-i}*,* $J_i(x_i, x_{-i})$ *is continuously differential and convex.*

Under Assumption 1, a NE of game (1) is defined as $x^* \in \Omega$ solution of the following variational inequality $VI(F, \Omega)$ [17, Prop. 1.4.2].

$$\langle F(x^*), x - x^* \rangle \geq 0, \forall x \in \Omega. \tag{2}$$

where F is the *pseudo-gradient* of the game defined as:

$$F(x) := \operatorname{col}\left((\nabla_{x_i} J_i (x_i, x_{-i}))_{i \in \mathcal{V}} \right) \tag{3}$$

or, equivalently, for any $\alpha > 0$ [17, Prop. 1.5.8],

$$x^* = \operatorname{proj}_{\Omega} (x^* - \alpha F(x^*)) \tag{4}$$

Assumption 2. *The pseudo-gradient mapping F is μ-strongly monotone and ℓ_0-Lipschitz continuous.*

In the projected-gradient algorithm with a fixed step-size, strong monotonicity of F is a standard assumption to guarantee the convergence of the algorithm [9,10]. Under Assumption 1 and 2, there exists a unique solution x^* of the $VI(F, \Omega)$, due to that the strong monotonicity of the pseudo-gradient F is guaranteed [23]. Thus the game (1) has an unique NE.

Assumption 3. *The graph $\mathcal{G}$ is strongly connected, and for it, the following hold:*
(i) Self-loops: $w_{i,i} > 0$ for all $i \in \mathcal{V}$;
(ii) Double stochasticity: $W\mathbf{1}_N = \mathbf{1}_N, \mathbf{1}_N^\top W = \mathbf{1}_N^\top$.

In practice, for connected undirected graphs, Assumption 3 is easily to meet. In our setting that considers a directed graph, which is strongly connected, we can use an iterative distributed strategy [24] to compute a weight assignment to make the obtained adjacency matrix with self-loops doubly stochastic.

Under Assumption 3, we can figure out $\sigma_{N-1}(W) < 1$, where $\sigma_{N-1}(W)$ stands for the second largest singular value of W. What's more, for any $x \in \mathbb{R}^N$,

$$\|W(x - \mathbf{1}_N \bar{x})\| \leq \sigma_{N-1}(W) \|x - \mathbf{1}_N \bar{x}\| \tag{5}$$

where $\bar{x} = \frac{1}{N}\mathbf{1}_N^\top x$ is the average of x. In order to simplify the notation, we use $\bar{\sigma}$ to represent $\sigma_{N-1}(W)$, and obviously $\bar{\sigma} \in (0, 1)$.

3 Algorithm Design

In this section, we propose a encoder-decoder scheme and based on it, we design a quantized pseudo-gradient fully distributed algorithm to seek a NE of the game (1). To handle the problems brought by partial-decision information, an auxiliary variable $\boldsymbol{x}_i$ is endowed with each player to provide an estimate of all other agents' decisions. Let $\boldsymbol{x}_i = \operatorname{col}\left((\boldsymbol{x}_{i,j})_{j \in \mathcal{V}} \right) \in \mathbb{R}^n$, where $\boldsymbol{x}_{i,i} := x_i$ and $\boldsymbol{x}_{i,j}$ denotes agent i 's estimate of agent j 's decision, for all $j \neq i$; $\boldsymbol{x}_{i,-i} = \operatorname{col}\left((\boldsymbol{x}_{i,l})_{l \in \mathcal{V} \setminus \{i\}} \right)$ is agent i's estimates of decisions made by all agents except himself. Also, let $\boldsymbol{x}_{ij}^Q \in \mathbb{R}^n$ denotes agent j 's estimate of $\boldsymbol{x}_i$, for all $j \neq i$.

3.1 Quantization Scheme Design

To satisfy the data-rate constraint, each i, $i \in \mathcal{V}$ transmits the quantified data of $\boldsymbol{x}_i$ to and also receives the quantified data of $\boldsymbol{x}_j$ from its in-neighbor j, $j \in \mathcal{N}_i$. A standard uniform quantizer $Q[\gamma]$ is defined for a vector $\gamma = \left(\gamma^1, \cdots, \gamma^m\right)^T \in \mathbb{R}^m$ with $2K+1$ quantization levels as follows:

$$Q[\gamma] = \left(q\left[\gamma^1\right], \ldots, q\left[\gamma^m\right]\right)^\top$$

where

$$q\left[\gamma^i\right] = \begin{cases} 0 & -\frac{1}{2} \leq \gamma^i \leq \frac{1}{2} \\ j & \frac{2j-1}{2} < \gamma^i \leq \frac{2j+1}{2}, j = 1, \ldots, K-1 \\ K & \frac{2K-1}{2} < \gamma^i \\ -q\left[-\gamma^i\right] & \gamma^i < -\frac{1}{2} \end{cases} \tag{6}$$

As we can see, the quantization bin width of $Q[\gamma]$ is 1. When $\|\gamma\|_\infty \leq K + \frac{1}{2}$, it is called unsaturated and then the quantization error could be bounded, i.e.,

$$\|\gamma - Q[\gamma]\|_\infty \leq \frac{1}{2} \tag{7}$$

Remark 1. *The quantization of* (6) *could be expressed as* $\lceil \log_2(2K+1) \rceil$ *bits, so through a* $(2K+1)$*-level quantizer* $Q[\ \cdot\ ]$*, we only need to transfer* $\lceil m \log_2(2K+1) \rceil$*-bit data if we want to transmit an* m*-dimensional vector* $\gamma \in \mathbb{R}^m$*.*

To achieve the consensus with quantization error and with the enlightenment of the adaptive quantization ideas to solve the problem of quantized average consensus problem in [17,25,26], we put forward an encoder-decoder proposal. A global scaling function $s(k)$ is introduced to control the quantization error, which decreases to 0 as $k \to \infty$. For each $i \in \mathcal{V}$ and $j \in \mathcal{N}^i$, agent i generates the quantized data $\boldsymbol{z}_i$ through the encoder $F_{i \to j}$ and sends it to agent $j \in \mathcal{N}^i$. Then, agent j decodes what received from agent i through the decoder $\daleth_{j \to i}$, and then obtains the estimation of agent i 's decision $\boldsymbol{x}_{ij}^Q$.

Correspondingly, the dynamic encoder F_i for agent i is given as follows:

$$\begin{cases} \boldsymbol{z}_i(k) = Q\left[\frac{1}{s(k)}\left(\boldsymbol{x}_i(k) - \boldsymbol{\xi}_i(k-1)\right)\right] \\ \boldsymbol{\xi}_i(k) = s(k)\boldsymbol{z}_i(k) + \boldsymbol{\xi}_i(k-1),\ \boldsymbol{\xi}_i(-1) = 0 \end{cases} \tag{8}$$

And the decoder $\daleth_{j \to i}$ is designed for agent j to handle the data received from agent i and obtain an estimation of $\boldsymbol{x}_i$

$$\boldsymbol{x}_{ij}^Q(k) = s(k)\boldsymbol{z}_i(k) + \boldsymbol{x}_{ij}^Q(k-1), \quad \boldsymbol{x}_{ij}^Q(-1) = 0 \tag{9}$$

Remark 2. *Note that $\boldsymbol{x}_{ij}^{Q}(k)$ and $\boldsymbol{\xi}_i(k)$ have the same dynamics with the same initial value, which implies that*

$$\boldsymbol{\xi}_i(k) = \boldsymbol{x}_{ij}^{Q}(k), \forall j \in \mathcal{N}^i \quad k = -1, 0, 1, \cdots \tag{10}$$

In this case, agent i is able to know the value of agent j's estimation of its decision, i.e., $\boldsymbol{x}_{ij}^{Q}$. Such a point is important for eliminating quantization errors in our algorithm design.

Assumption 4. *There exists positive constant M_0 and M^* such that for $i \in \mathcal{V}$*

$$\|x_i(0)\|_\infty \leq M_0, \ \|x^*\|_\infty \leq M^*. \tag{11}$$

Assumption 4 ensures the quantizer is not saturated at the initial moment $t = 0$. It has been widely used in the research of quantitative cooperative control of multi-agent system [17,18].

3.2 Distributed and Quantized Algorithm Design

Algorithm 1. Distributed quantized algorithm for agent i

Initialize: for all $i \in \mathcal{V}$, set $\boldsymbol{x}_{i,i}(0) \in \Omega_i$, $\boldsymbol{x}_{i,-i}(0) \in \mathbb{R}^{n-n_i}$, $\boldsymbol{\xi}_i(-1) = 0$
  for $k \in \mathbb{N}$ **do**
    for all $i = 1, 2, \cdots, N$ **do**
      agent i sends $\boldsymbol{z}_i(k)$ to $j \in \mathcal{N}_i$
      for all $j \in \mathcal{N}_i$ **do**
        $\boldsymbol{x}_{ij}^{Q}(k) = s(k)\boldsymbol{z}_i(k) + \boldsymbol{x}_{ij}^{Q}(k-1)$,
        $\boldsymbol{x}_{ij}^{Q}(-1) = 0$
      end for
      $\widehat{\boldsymbol{x}}_i(k) = \sum_{j=1 j\neq i}^{N} w_{j,i}\boldsymbol{x}_{ji}^{Q}(k) + w_{i,i}\boldsymbol{\xi}_i(k)$
      $x_{i,i}(k+1) = \mathrm{proj}_{\Omega_i}\left(\widehat{\boldsymbol{x}}_{i,i}(k) - \alpha\nabla_{x_i}J_i(\widehat{\boldsymbol{x}}_i(k))\right)$
      $x_{i,-i}(k+1) = \widehat{\boldsymbol{x}}_{i,-i}(k)$
      $\boldsymbol{z}_i(k+1) = Q\left[\frac{1}{s(k+1)}\left(\boldsymbol{x}_i(k+1) - \boldsymbol{\xi}_i(k)\right)\right]$
      $\boldsymbol{\xi}_i(k+1) = s(k+1)\boldsymbol{z}_i(k+1) + \boldsymbol{\xi}_i(k)$
    end for
  end for

Next, Algorithm 1 is written in compact form. Let, $\boldsymbol{x} = \mathrm{col}(\boldsymbol{x}_i)_{i\in\mathcal{V}} \in \mathbb{R}^{Nn}$, and from (10), for all $i \in \mathcal{V}$, we can use $\boldsymbol{\xi}_i$ to represent its out-neighbors' estimates of its decision, i.e., $\boldsymbol{x}_{ij}^{Q}$, $j \in \mathcal{N}^i$. Then we can define $\boldsymbol{x}^Q = \mathrm{col}(\boldsymbol{\xi}_i)_{i\in\mathcal{V}} \in \mathbb{R}^{Nn}$. And, for all $i \in \mathcal{V}$

$$\mathcal{R}_i := \left[\mathbf{0}_{n_i\times n_{<i}} \ I_{n_i} \ \mathbf{0}_{n_i\times n_{>i}}\right] \in \mathbb{R}^{n_i\times n} \tag{12}$$

where $n_{<i} := \sum_{j<i, j\in\mathcal{V}} n_j$, $n_{>i} := \sum_{j>i, j\in\mathcal{V}} n_j$. Then we can use $\mathcal{R}_i$ to select the i th component from an n-dimensional vector, i.e., $\mathcal{R}_i\boldsymbol{x}_i = \boldsymbol{x}_{i,i} = x_i$. With

$\mathcal{R}_i \boldsymbol{x}_i = \boldsymbol{x}_{i,i} = x_i$, $\boldsymbol{x}$ can be written as $x = \mathcal{R}\boldsymbol{x}$, where $\mathcal{R} := \operatorname{diag}\left((\mathcal{R}_i)_{i\in\mathcal{V}}\right) \in \mathbb{R}^{n\times Nn}$. The *extended pseudo-gradient mapping* $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F}(\boldsymbol{x}) := \operatorname{col}\left(\left(\nabla_{x_i} J_i\left(x_i, \boldsymbol{x}_{i,-i}\right)\right)_{i\in\mathcal{V}}\right) \tag{13}$$

With above notations, Algorithm 1 reads in compact form as:

$$\boldsymbol{x}\left(k+1\right) = \operatorname{proj}_{\boldsymbol{\Omega}}\left(\boldsymbol{W}\boldsymbol{x}^{Q}\left(k\right) - \alpha\mathcal{R}^{\top}\boldsymbol{F}\left(\boldsymbol{W}\boldsymbol{x}^{Q}\left(k\right)\right)\right) \tag{14}$$

where $\boldsymbol{\Omega} := \left\{\boldsymbol{x}\in\mathbb{R}^{Nn} \mid \mathcal{R}\boldsymbol{x}\in\Omega\right\}$ and $\boldsymbol{W} := W\otimes I_n$.

Lemma 1. *([21, Lemma 3]): Let Assumption 2 hold. Then there exists some* $\mu \le \ell \le \ell_0$*, the extended pseudo-gradient mapping* $\boldsymbol{F}$ *is* ℓ*-Lipschitz continuous.*

Lemma 2. *Suppose Assumption 1–4 hold and let*

$$M_\alpha = \begin{bmatrix} 1 - \frac{2\alpha\mu}{N} + \frac{\alpha^2\ell_0^2}{N} & \left(\frac{\alpha(\ell+\ell_0)+\alpha^2\ell_0\ell}{\sqrt{N}}\right)\bar{\sigma} \\ \left(\frac{\alpha(\ell+\ell_0)+\alpha^2\ell_0\ell}{\sqrt{N}}\right)\bar{\sigma} & \left(1+2\alpha\ell+\alpha^2\ell^2\right)\bar{\sigma}^2 \end{bmatrix} \tag{15}$$

If the step size $\alpha > 0$ *and*

$$\begin{aligned} &\alpha < min\left\{\frac{\bar{\sigma}}{3\ell_0}, \frac{2\mu}{\ell_0^2}\right\} \\ &0 < 2\mu\left(1-\bar{\sigma}^2\right) - \alpha\left(\bar{\sigma}^2\left(2\ell_0\ell + \ell^2 + 4\mu\ell + 2\ell_0^2\right) - \ell_0^2\right) \\ &\quad - \alpha^2\left(\ell_0\ell^2 + \mu\ell^2 + 2\ell_0^2\ell\right)2\bar{\sigma}^2 - \alpha^3 2\ell_0^2\ell^2\bar{\sigma}^2 \end{aligned} \tag{16}$$

then

$$\rho_\alpha := \lambda_{\max}\left(M_\alpha\right) = \|M_\alpha\| < 1. \tag{17}$$

In order to guarantee that the quantizer will never be saturated, i.e.

$$\forall k,\ \|\frac{1}{s(k)}\left(\boldsymbol{x}_i(k) - \boldsymbol{\xi}_i(k-1)\right)\|_\infty \le K + \frac{1}{2} \tag{18}$$

we select the quantizer's parameters as follows,

(a) Design the scaling function $s(k)$ as:

$$s\left(k\right) = M\left(1-\rho_\alpha^{1/4}\right)\rho_\alpha^{(k-1)/4} \tag{19}$$

where $M := M_0 + M^*$.

(b) Choose K:

$$K = \lceil \frac{\rho_\alpha^{1/2} + \rho_\alpha^{1/4} + 2\rho_\alpha^{-1/4}}{2\left(1-\rho_\alpha^{1/4}\right)} - \frac{1}{2} \rceil \tag{20}$$

Theorem 1. *Suppose Assumption 1–4 hold, make sure that positive number* α *meet* (16)*, and the quantizer's parameters are chosen according to* (19) *and* (20)*. Then the sequence* $\left(\boldsymbol{x}\left(k\right)\right)_{k\in\mathbb{N}}$ *generated by Algorithm 1 will converge to* $x^* = 1_N \otimes x^*$*, where* x^* *is the NE of the game in (1), with*

$$\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\| \le M(\sqrt{\rho_\alpha})^{k+1} \tag{21}$$

4 Simulation

For the simulation purpose, we consider a quadratic model from classical economic in [3]. There are a total of N manufacturers producing homogeneous products commodity. Its decision x_i is the vector of firm i's production quantity, for $i = 1, \cdots, N$. u_i stands for firm i's cost function of producing the commodity and f denotes the demand price, and they're functions of x_i and $\sum_{i=1}^{N} x_i^2$, respectively. Each firm i intends to maximize its profits, i.e., minimize its total cost function $J_i(x_i, x_{-i}) = u_i(x_i) - x_i f\left(\sum_{i=1}^{N} x_i^2\right)$.

In the following we verify our fully distributed quantized NE seeking algorithm via a numerical simulation. The setup we considered consists of five companies ($N = 5$), each has a production cost function with the form $u_i(x_i) = c_i x_i$ where $c_i = 100 + 50(i-1)$, for $i = 1, \cdots, 5$. The form of the demand price function is $f\left(\sum_{i=1}^{5} x_i^2\right) = 600 - \sum_{i=1}^{5} x_i^2$. The communication graph G is as in [1, Fig. 1]. We compare our quantized algorithm with algorithm 1 in [3] and verify the correctness of our quantized algorithm through this example over $X_i = [0, 100]$ for all $i \in \{1, \cdots, 5\}$. Figure 1 demonstrates the convergence of the quantities produced by companies to the theoretical Nash equilibrium $\left(x^* = [10.35, 9.06, 7.56, 5.67, 2.67]^T\right)$ with no quantization errors. The relative error evolution of all agents' decisions denoted as $\frac{\|x - x^*\|}{x^*}$ is shown in Fig. 2.

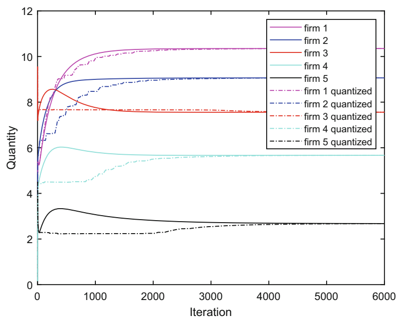

Fig. 1. The tracking trajectory

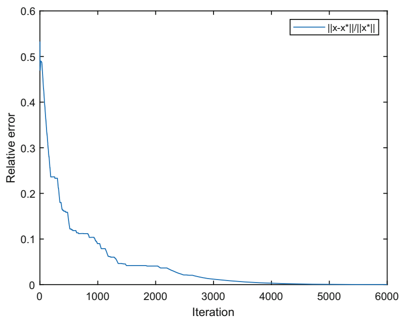

Fig. 2. The tracking error

5 Conclusions

This paper has considered distributed NE seeking with finite communication bandwidth constraints over a directed graph. Based on a proposed encoder-decoder scheme, we proposed a quantized fully distributed NE seeking algorithm. Both theoretical proof and numerical simulation verified that our algorithm would exponentially converge to the real NE.

6 Appendix

Proof of Theorem 1

Let $\boldsymbol{E} := \{\boldsymbol{y} \in \mathbb{R}^{Nn} \mid \boldsymbol{y} = \mathbf{1}_N \otimes y, y \in \mathbb{R}^n\}$ denote the estimate consensus subspace, $\boldsymbol{E}_\perp := \left\{\boldsymbol{y} \in \mathbb{R}^{Nn} \mid (\mathbf{1}_N \otimes I_n)^\top \boldsymbol{y} = \mathbf{0}_n\right\}$ stand for its orthogonal complement with $\mathbb{R}^{Nn} = \boldsymbol{E} \oplus \boldsymbol{E}_\perp$. Any vector $\boldsymbol{x} \in \mathbb{R}^{Nn}$ has a decomposition as $\boldsymbol{x} = \boldsymbol{x}_\parallel + \boldsymbol{x}_\perp$, with $\boldsymbol{x}_\parallel = \mathrm{proj}_{\boldsymbol{E}}(\boldsymbol{x}), \boldsymbol{x}_\perp = \mathrm{proj}_{\boldsymbol{E}_\perp}(\boldsymbol{x}) = \frac{1}{N}\left(\mathbf{1}_N \mathbf{1}_N^\top \otimes I_n\right)\boldsymbol{x}$, and $\boldsymbol{x}_\parallel^\top \boldsymbol{x}_\perp = 0$. Meanwhile, for the sake of simplicity, we use $\boldsymbol{F}\boldsymbol{x}$ and Fx in place of $\boldsymbol{F}(\boldsymbol{x})$ and $F(x)$. Then the iteration in (14) can be written as

$$\boldsymbol{x}(k+1) = \mathrm{proj}_{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\xi}}(k) - \alpha \mathcal{R}^\top \boldsymbol{F}\hat{\boldsymbol{\xi}}(k)\right), \hat{\boldsymbol{\xi}}(k) = \boldsymbol{W}\boldsymbol{\xi}(k) \tag{22}$$

Let x^* denote the unique NE of the game in (1), and $\boldsymbol{x}^* := \mathbf{1}_N \otimes x^*$. Recalling that $x^* = \mathrm{proj}_\Omega(x^* - \alpha F x^*)$ by (4), then $\boldsymbol{x}^* = \mathrm{proj}_{\boldsymbol{\Omega}}\left(\boldsymbol{x}^* - \alpha\mathcal{R}^\top \boldsymbol{F}\boldsymbol{W}\boldsymbol{x}^*\right)$. Owing to the fact that, $\boldsymbol{W}\boldsymbol{x}^* = (W \otimes I_n)(\mathbf{1}_N \otimes x^*) = \mathbf{1}_N \otimes x^* = \boldsymbol{x}^*$; hence $\boldsymbol{x}^*$ is a fixed point for (22). Let $\boldsymbol{\xi}(k) = \boldsymbol{\xi} \in \mathbb{R}^{Nn}$ and $\hat{\boldsymbol{\xi}} = \boldsymbol{W}\boldsymbol{\xi} = \hat{\boldsymbol{\xi}}_\parallel + \hat{\boldsymbol{\xi}}_\perp = \mathbf{1}_N \otimes \hat{\xi}_\parallel + \hat{\boldsymbol{\xi}}_\perp \in \mathbb{R}^{Nn}$. Thereby, it holds that

$$\begin{aligned}
&\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\|^2 \\
=&\left\|\mathrm{proj}_{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\xi}} - \alpha\mathcal{R}^\top \boldsymbol{F}\hat{\boldsymbol{\xi}}\right) - \mathrm{proj}_{\boldsymbol{\Omega}}\left(\boldsymbol{x}^* - \alpha\mathcal{R}^\top \boldsymbol{F}\boldsymbol{x}^*\right)\right\|^2 \\
\leq&\left\|\left(\hat{\boldsymbol{\xi}} - \alpha\mathcal{R}^\top \boldsymbol{F}\hat{\boldsymbol{\xi}}\right) - \left(\boldsymbol{x}^* - \alpha\mathcal{R}^\top \boldsymbol{F}\boldsymbol{x}^*\right)\right\|^2 \\
=&\left\|\hat{\boldsymbol{\xi}}_\parallel + \hat{\boldsymbol{\xi}}_\perp - \boldsymbol{x}^* + \alpha\mathcal{R}^\top\left(-\boldsymbol{F}\hat{\boldsymbol{\xi}} + \boldsymbol{F}\boldsymbol{x}^* + \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel\right)\right\|^2 \\
=&\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|^2 + \left\|\hat{\boldsymbol{\xi}}_\perp\right\|^2 + \alpha^2\left\|\mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}} - \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel + \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{F}\boldsymbol{x}^*\right)\right\|^2 \\
&- 2\alpha\left(\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right)^\top \mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}} - \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel\right) - 2\alpha\left(\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right)^\top \mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{F}\boldsymbol{x}^*\right) \\
&- 2\alpha\hat{\boldsymbol{\xi}}_\perp\mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}} - \boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel\right) - 2\alpha\hat{\boldsymbol{\xi}}_\perp^\top\mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{F}\boldsymbol{x}^*\right) \\
\leq&\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|^2 + \left\|\hat{\boldsymbol{\xi}}_\perp\right\|^2 + \alpha^2\left(\ell^2\left\|\hat{\boldsymbol{\xi}}_\perp\right\|^2 + \frac{\ell_0^2}{N}\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|^2\right. \\
&\left.+\frac{2\ell_0\ell}{\sqrt{N}}\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|\left\|\hat{\boldsymbol{\xi}}_\perp\right\|\right) + \frac{2\alpha\ell}{\sqrt{N}}\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|\left\|\hat{\boldsymbol{\xi}}_\perp\right\| \\
&- \frac{2\alpha\mu}{N}\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|^2 + 2\alpha\ell\left\|\hat{\boldsymbol{\xi}}_\perp\right\|^2 + \frac{2\alpha\ell_0\left\|\hat{\boldsymbol{\xi}}_\perp\right\|\left\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\right\|}{\sqrt{N}}
\end{aligned}$$

where the first inequality follows by nonexpansiveness of the projection [22, Prop. 4.16], and to bound the addends in penultimate equation we used, in the order:

- $3^{\rm rd}$ term:$\|\mathcal{R}\| = 1$, Lipschitz continuity of $\boldsymbol{F}$, and $\|\boldsymbol{F}\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{F}\boldsymbol{x}^*\| = \|F\hat{\xi}_\parallel - Fx^*\| \leq \ell_0\|\hat{\xi}_\parallel - x^*\| = \frac{\ell_0}{\sqrt{N}}\|\hat{\boldsymbol{\xi}}_\parallel - \boldsymbol{x}^*\|$;

- 4^{th} term: $\left\|\mathcal{R}\left(\mathbf{1}\otimes\left(\hat{\xi}_{\|}-x^*\right)\right)\right\| = \left\|\hat{\xi}_{\|}-x^*\right\| = \frac{1}{\sqrt{N}}\left\|\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{x}^*\right\|$;
- 5^{th} term: $\left(\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{x}^*\right)^\top\mathcal{R}^\top\left(\boldsymbol{F}\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{F}\boldsymbol{x}^*\right) = \left(\hat{\xi}_{\|}-x^*\right)^\top\left(F\hat{\xi}_{\|}- Fx^*\right) \geq \mu\left\|\hat{\xi}_{\|}-x^*\right\|^2 = \frac{1}{N}\left\|\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{x}^*\right\|^2$;
- 6^{th} term:Lipschitz continuity of $\boldsymbol{F}$;
- 7^{th} term:$\left\|\boldsymbol{F}\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{F}\boldsymbol{x}^*\right\| \leq \frac{\ell_0}{\sqrt{N}}\left\|\hat{\boldsymbol{\xi}}_{\|}-\boldsymbol{x}^*\right\|$ as above.

Besides, for every $\boldsymbol{\xi}=\boldsymbol{\xi}_{\|}+\boldsymbol{\xi}_{\perp}\in\mathbb{R}^{Nn}$ and for all $k\in\mathbb{N}$, it holds that $\hat{\boldsymbol{\xi}} = \boldsymbol{W}\boldsymbol{\xi} = \boldsymbol{\xi}_{\|}+\boldsymbol{W}\boldsymbol{\xi}_{\perp}$, where $\boldsymbol{W}\boldsymbol{\xi}_{\perp}\in\boldsymbol{E}_{\perp}$, by doubly stochasticity of W, and $\|\hat{\boldsymbol{x}}_{\perp}\| = \|\boldsymbol{W}_k\boldsymbol{\xi}_{\perp}\| \leq \bar{\sigma}\|\boldsymbol{\xi}_{\perp}\|$ by (5), properties of the Kronecker product and the unsaturation by (18). Thus, we can finally write, for all $k\in\mathbb{N}$, for all $\boldsymbol{x}(k+1)\in\mathbb{R}^{Nn}$,

$$
\begin{aligned}
\|\boldsymbol{x}(k+1)-\boldsymbol{x}^*\|^2 &\leq \begin{bmatrix}\left\|\boldsymbol{\xi}_{\|}(k)-\boldsymbol{x}^*\right\| \\ \|\boldsymbol{\xi}_{\perp}(k)\|\end{bmatrix}^\top M_\alpha \begin{bmatrix}\left\|\boldsymbol{\xi}_{\|}(k)-x^*\right\| \\ \|\boldsymbol{\xi}_{\perp}(k)\|\end{bmatrix} \\
&\leq \rho_\alpha\left(\left\|\boldsymbol{\xi}_{\|}(k)-\boldsymbol{x}^*\right\|^2+\|\boldsymbol{\xi}_{\perp}(k)\|^2\right) \\
&= (\sqrt{\rho_\alpha}(\|\boldsymbol{\xi}(k)-\boldsymbol{x}^*\|)^2 \\
&\leq (\sqrt{\rho_\alpha}(\|\boldsymbol{x}(k)-\boldsymbol{x}^*\|+\frac{s(k)}{2}))^2 \\
&\cdots \\
&\leq \frac{1}{2}[(\sqrt{\rho_\alpha})^{k+1}s(0)+(\sqrt{\rho_\alpha})^k s(1)+\cdots+(\sqrt{\rho_\alpha})s(k)])^2 \\
&+((\sqrt{\rho_\alpha})^{k+1}\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|_\infty \\
&= ((\sqrt{\rho_\alpha})^{k+1}\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|_\infty+\frac{M\rho_\alpha^{\frac{k+1}{2}}(1-\rho_\alpha^{-\frac{k+1}{4}})}{2\rho_\alpha^{\frac{1}{4}}})^2
\end{aligned}
$$

From (11), and (17), we obtain

$$\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|_\infty \leq M$$

and

$$\frac{M\rho_\alpha^{\frac{k+1}{2}}(1-\rho_\alpha^{-\frac{k+1}{4}})}{2\rho_\alpha^{\frac{1}{4}}} < 0$$

Then,

$$\|\boldsymbol{x}(k+1)-\boldsymbol{x}^*\|_\infty \leq M(\sqrt{\rho_\alpha})^{k+1} \tag{23}$$

Finally, when $k\to\infty$, $M(\sqrt{\rho_\alpha})^{k+1}$ decrease to 0. Hence, Algorithm 1 converges.

References

1. Yang, S., Wang, J., Liu, Q.: Cooperative-competitive multiagent systems for distributed minimax optimization subject to bounded constraints. IEEE Trans. Autom. Control **64**(4), 1358–1372 (2019)
2. Yang, S., Wang, J., Liu, Q.: A multi-agent system with a proportional-integral protocol for distributed constrained optimization. IEEE Trans. Autom. Control **62**(7), 3461–3467 (2017)
3. Salehisadaghiani, F., Pavel, L.: Nash equilibrium seeking by a gossip-based algorithm. In: 53rd IEEE Conference on Decision and Control, Los Angeles, pp. 1155–1160. IEEE Press (2014)
4. Ye, M., Hu, G.: Distributed Nash equilibrium seeking by a consensus based approach. IEEE Trans. Autom. Control **62**(9), 4811–4818 (2017)
5. Gadjov, D., Pavel, L.: A passivity-based approach to Nash equilibrium seeking over networks. IEEE Trans. Autom. Control **64**(3), 1077–1092 (2019)
6. De Persis, C., Grammatico, S.: Distributed averaging integral Nash equilibrium seeking on networks. Automatica **110**, 108548 (2019)
7. Lu, K., Jing, G., Wang, L.: Distributed algorithms for searching generalized Nash equilibrium of noncooperative games. IEEE Trans. Cybern. **49**(6), 2362–2371 (2019)
8. Zeng, X., Chen, J., Liang, S., Hong, Y.: Generalized Nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game. Automatica **103**, 20–26 (2019)
9. Pavel, L.: Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach. IEEE Trans. Autom. Control **65**(4), 1584–1597 (2020)
10. Bianchi, M., Grammatico, S.: Fully distributed Nash equilibrium seeking over time-varying communication networks with linear convergence rate. IEEE Control Syst. Lett. **5**(2), 499–504 (2021)
11. Koshal, J., Nedi, A., Shanbhag, U.V.: Distributed algorithms for aggregative games on graphs. Oper. Res. **64**, 680–704 (2016)
12. Xu, W., Wang, Z., Hu, L., Kurths, J.: State estimation under joint false data injection attacks: dealing with constraints and insecurity. IEEE Trans. Autom. Control (2021) . https://doi.org/10.1109/TAC.2021.3131145
13. Xu, W., Kurths, J., Wen, G., Yu, X.: Resilient event-triggered control strategies for second-order consensus. IEEE Trans. Autom. Control (2021) . https://doi.org/10.1109/TAC.2021.3122382
14. Xu, W., He, W., Ho, D.W.C., Kurths, J.: Fully distributed observer-based consensus protocol: adaptive dynamic event-triggered schemes. Automatica **139**, 110188 (2021)
15. Li, Q., Liang, J.: Dissipativity of the stochastic Markovian switching CVNNs with randomly occurring uncertainties and general uncertain transition rates. Int. J. Syst. Sci. **51**(6), 1102–1118 (2020)
16. Rabbat, M.G., Nowak, R.D.: Quantized incremental algorithms for distributed optimization. IEEE J. Sel. Areas Commun. **23**(4), 798–808 (2005)
17. Yi, P., Hong, Y.: Quantized subgradient algorithm and data-rate analysis for distributed optimization. IEEE Trans. Control of Netw. Syst. **1**(4), 380–392 (2014)
18. Li, H., Liu, S., Soh, Y.C., Xie, L.: Event-triggered communication and data rate constraint for distributed optimization of multiagent systems. IEEE Trans. Syst. Man Cybern. Syst. **48**(11), 1908–1919 (2018)

19. Ji, M., Ji, H., Sun, D., Feng, G.: An approach to quantized consensus of continuous-time linear multi-agent systems. Automatica **91**, 98–104 (2018)
20. Lei, J., Yi, P., Shi, G., Anderson, B.D.: Distributed algorithms with finite data rates that solve linear equations. SIAM J. Optim. **30**, 1191–1222 (2020)
21. Chen, Z., Ji, H.: Distributed quantized optimization design of continuous-time multiagent systems over switching graphs. IEEE Trans. Syst. Man Cybern. Syst. **51**, 7152–7163 (2020). https://doi.org/10.1109/TSMC.2020.2966636
22. Nekouei, E., Nair, G.N., Alpcan, T.: Performance analysis of gradient-based Nash Seeking algorithms under quantization. IEEE Trans. Autom. Control **61**(12), 3771–3783 (2016)
23. Facchinei, F., Pang, J.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, Cham (2007). https://doi.org/10.1007/b97543
24. Gharesifard, B., Corts, J.: Distributed strategies for generating weight-balanced and doubly stochastic digraphs. Eur. J. Control. **18**(6), 539–557 (2012)
25. Li, T., Xie, L.: Distributed consensus over digital networks with limited bandwidth and time-varying topologies. Automatica **47**(9), 2006–2015 (2011)
26. Zhang, Q., Zhang, J.: Quantized data based distributed consensus under directed time-varying communication topology. SIAM J. Control Optim. **51**(1), 332–352 (2013)
27. Bianchi, M., Grammatico, S.: A continuous-time distributed generalized Nash equilibrium seeking algorithm over networks for double-integrator agents, St. Petersburg. In: 2020 European Control Conference, pp. 1474–1479 (2020)
28. Bauschke, H.H., Combettes, P.L.: Convex analysis and monotone operator theory in Hilbert spaces. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-48311-5