

Asynchronous Networked Aggregative Games

Rongping Zhu * Jiaqi Zhang * Keyou You * Tamer Başar †

January 25, 2021

Abstract

We propose a fully asynchronous networked aggregative game (Asy-NAG) where each player minimizes a cost function that depends on its local action and the aggregate of all players' actions. In sharp contrast to the existing NAGs, each player in our Asy-NAG can compute an estimate of the aggregate action at *any* wall-clock time by only using (possibly stale) information from nearby players of a directed network. Such an asynchronous update does not require any coordination among players. Moreover, we design a novel distributed algorithm with an aggressive mechanism for each player to adaptively adjust the optimization stepsize per update. Particularly, the slow players in terms of updating their estimates smartly increase their stepsizes to catch up with the fast ones. Then, we develop an augmented system approach to address the asynchronicity and the information delays between players, and rigorously show the convergence to a Nash equilibrium of the Asy-NAG via a perturbed coordinate algorithm which is also of independent interest. Finally, we evaluate the performance of the distributed algorithm through numerical simulations.

1 Introduction

An aggregative game is a Nash game where each player's cost is a function of its action and the aggregate of all players' actions, and has been widely used in flow control (Alpcan & Başar 2002, Barrera & Garcia 2014), resource allocation (Ma et al. 2013, Gharesifard et al. 2016), and demand response (Li et al. 2015, Ma et al. 2013). In this work, we consider the *Networked Aggregative Games* (NAGs) over a directed peer-to-peer (P2P) network, where each player individually makes decisions by only using (possibly stale) information from neighboring players of the network, see Fig. 1 for an illustrative example.

The NAGs can be categorized as synchronous or asynchronous ones depending on whether players make decisions in a synchronized manner or not. In this paper, we propose an asynchronous NAG (Asy-NAG), which in fact is *fully asynchronous*, adopting the asynchronism definition in Herz & Marcus (1993). To elaborate the differences between the synchronous NAG and the Asy-NAG, we use Fig. 1, whose running states for each player are shown in Fig. 2. A gray bar represents the computation time per update of a player and the red arrows represent information flow directions between players. In a synchronous NAG (Fig. 2a), *all* players are essentially synchronized to compute at the same wall-clock time $t(k)$. While in an Asy-NAG (Fig. 2b), each player acts *independently* without waiting for other players or delayed messages. Thus, the index k in Fig. 2(a) increases by one only when all players are synchronized to start their updates, and k in Fig. 2(b) increases whenever a player starts an update. Importantly, the Asy-NAG does not require any (usually central) coordinator to synchronize the update process and k is not known to players.

We believe such Asy-NAGs have not been studied yet, even though they are of more practical importance, in addition to being notoriously difficult to solve. First, players in a non-cooperative aggregative game can hardly be synchronized to update their actions. For example, players may pause to operate

*R. Zhu, J. Zhang and K. You are with the Department of Automation and BNRist, Tsinghua University, Beijing 100084, China. Emails: zhurp19@mails.tsinghua.edu.cn, zjq16@mails.tsinghua.edu.cn, youky@tsinghua.edu.cn.

†T. Başar is with Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. Email: basar1@illinois.edu.

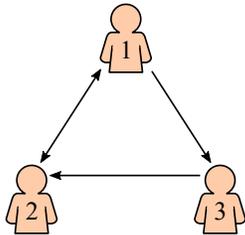


Figure 1: A directed P2P network with three players.

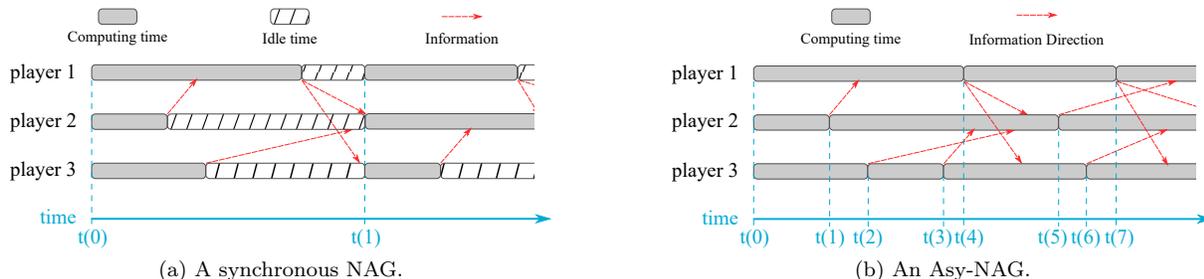


Figure 2: Comparison of synchronous NAGs and Asy-NAGs.

other tasks, or just remain idle. In fact, the synchronization of a large-scale NAG can be hard and time-consuming. Second, due to the asynchronicity and information delays, the updating frequency of each player is unknown and unpredictable. Thus, it is considerably different from the random gossip-based NAG in Koshal et al. (2016), Salehisadaghiani & Pavel (2016), Cenedese et al. (2020), requiring the development of a new approach for algorithm design.

For synchronous NAGs, there are a number of distributed algorithms that converge to a Nash equilibrium (NE) by adopting the consensus-based technique (Koshal et al. 2016, Belgioioso et al. 2020), ADMM-based methods (Salehisadaghiani et al. 2019), the best response dynamics (Li & Başar 1987b, Lei et al. 2020), model-free method (Frihauf et al. 2012), etc. It is worth stressing that the random gossip-based NAG requires a global coordinator, which is key to their algorithm design and does not satisfy the asynchronism of the Asy-NAGs. Notably, they can be simply modeled as synchronous NAGs over randomly time-varying networks, which is impossible for our Asy-NAG and an augmented network has to be constructed in this work. Thus, the existing distributed algorithms for NAGs cannot be applied to the Asy-NAG.

In fact, the fully asynchronous setting of this work is not new and has been studied for quite a long history in the parallel and distributed computing problems. Chazan & Miranker (1969) first consider it for solving linear equations problems with a parallel system. Bertsekas (1983) and Tsitsiklis et al. (1986) proposed asynchronous coordinate algorithms for fixed point problems. Note that Li & Başar (1987a) studied asynchronous non-cooperative games based on the best response dynamics and every player is able to directly access all other players' actions, which is different from the NAGs. Recently, it has been exploited for solving distributed optimization problems (Nedić & Ozdaglar 2010, Zhang & You 2019, Assran & Rabbat 2021, Sha et al. 2020), where players (or computing nodes) cooperate to optimize a common objective function. This is clearly different from NAGs as each player needs to minimize an individual cost function.

Accordingly, we propose a novel distributed algorithm with an aggressive stepsizes scheme to find a NE of the Asy-NAGs. First, we design an asynchronous perturbed push-sum algorithm to dynamically estimate the aggregate action for each player over the directed network. When being activated to update, each player uses its locally estimated aggregate action to update its own action. Second, to address the unpredictable update time instances of players, we propose a novel aggressive scheme for each player to

adaptively adjust its optimization stepsize per update. Specifically, each player maintains a local counter to roughly track the maximum number of updates of its neighboring players, based on which slow players smartly increase stepsizes to catch up with fast ones. Such an aggressive mechanism is in sharp contrast to the existing works by reducing the stepsize of fast players, and helps to accelerate the convergence to the NE. A similar idea has been adopted to address the non-exact convergence issue in distributed optimization problems in Zhang & You (2020).

The proposed algorithm is then proved to converge to an NE of the Asy-NAG asymptotically via an augmented network approach. Particularly, we use the virtual index k to model the delayed information and a sequence of virtual nodes to construct an augmented network. As a byproduct, we propose a perturbed coordinate pseudo-gradient algorithm for the augmented system, based on which the distributed algorithm for Asy-NAG is proved to converge to the NE. Finally, we validate the performance of our algorithm with numerical simulations on the renowned Nash-Cournot games over several directed networks.

The remainder of the paper is organized as follows. In Section 2, we introduce the AGs and our Asy-NAG over a P2P network in details. Section 3 proposes the distributed algorithm for the Asy-NAG. In Section 4, we show how to construct the augmented network and Section 5 provides a novel perturbed coordinate pseudo-gradient algorithm for the AG. In Section 6, we prove the convergence to an NE of the proposed algorithm. Section 7 presents the results of numerical experiments. Concluding remarks are drawn in Section 8. Some technical proofs are included in the appendices.

Notations: Throughout this paper, \mathbf{x}^\top and X^\top denote the transposes of a vector \mathbf{x} and a matrix X , respectively. Let $\mathbf{x}^\top \mathbf{y}$ denote the inner product of vectors \mathbf{x} and \mathbf{y} , and $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$ denote the Euclidean norm of a vector \mathbf{x} . We use $[\mathbf{x}_1, \dots, \mathbf{x}_n]$ to denote the horizontal stack of vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$, and $\mathbf{1}_n$ and $\mathbf{0}_n$ to denote the n -dimensional vectors with all ones and all zeros, respectively. For a matrix A , we write $[A]_{ij}$ to denote its (i, j) -th element. For a scalar a , we let $[a]$ denote the largest integer less than a . For a set \mathcal{N} , let $|\mathcal{N}|$ denote its cardinality, i.e., the number of elements in \mathcal{N} . The Minkowski sum of two sets \mathcal{X} and \mathcal{Y} is formed by $\mathcal{X} + \mathcal{Y} := \{\mathbf{x} + \mathbf{y} | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$, and let $a\mathcal{X} := \{a\mathbf{x} | \mathbf{x} \in \mathcal{X}\}$. Finally, we use $\Pi_{\mathcal{X}}$ to denote the Euclidean projection operator over a closed convex set \mathcal{X} , i.e., $\Pi_{\mathcal{X}}(\mathbf{x}) := \arg \min_{\mathbf{z} \in \mathcal{X}} \|\mathbf{x} - \mathbf{z}\|$.

2 Problem Formulation

Consider a set of n players indexed by $\mathcal{N} = \{1, \dots, n\}$. For $i \in \mathcal{N}$, let $f_i(\mathbf{x}_i, \bar{\mathbf{x}})$ denote the cost function of player i , where $\mathbf{x}_i \in \mathcal{X}_i \subseteq \mathbb{R}^p$ is the action of player i and $\bar{\mathbf{x}} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ is the aggregate of all players' actions. The objective of player i in an aggregative game (AG) is to solve a local optimization problem

$$\begin{aligned} & \text{minimize} && f_i \left(\mathbf{x}_i, \frac{1}{n} \mathbf{x}_i + \frac{1}{n} \sum_{j \neq i}^n \mathbf{x}_j \right) \\ & \text{subject to} && \mathbf{x}_i \in \mathcal{X}_i, \end{aligned} \tag{1}$$

where \mathbf{x}_j , $j \neq i$, $j \in \mathcal{N}$ are fixed, and to reach an NE among all players, which is defined in precise terms below (Başar & Olsder 1998).

Definition 1. *An n -tuple of actions*

$$\mathbf{x}^* := [(\mathbf{x}_1^*)^\top, (\mathbf{x}_2^*)^\top, \dots, (\mathbf{x}_n^*)^\top]^\top$$

is a Nash equilibrium (NE) if for all $i \in \mathcal{N}$ and $\mathbf{x}_i \in \mathcal{X}_i$,

$$f_i(\mathbf{x}_i^*, \bar{\mathbf{x}}^*) \leq f_i \left(\mathbf{x}_i, \frac{1}{n} \mathbf{x}_i + \frac{1}{n} \sum_{j \neq i}^n \mathbf{x}_j^* \right).$$

In a centralized AG, a coordinator is needed to gather the actions from all players to compute and broadcast the aggregate action $\bar{\mathbf{x}}$. On the other hand, in a networked AG (NAG), the players are spatially distributed on a P2P network and each player can only collect information from neighboring players under the topology of the network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where \mathcal{N} is the set of players and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the edge set

for information directions, i.e., $(i, j) \in \mathcal{E}$ if and only if i can directly send information to j . We let $\mathcal{N}_{\text{in}}^i = \{j | (j, i) \in \mathcal{E}\} \cup \{i\}$ denote the set of in-neighbors of player i , and $\mathcal{N}_{\text{out}}^i = \{j | (i, j) \in \mathcal{E}\} \cup \{i\}$ the set of out-neighbors of i . In this network, player i is able to collect information from players in $\mathcal{N}_{\text{in}}^i$ and broadcast information to players in $\mathcal{N}_{\text{out}}^i$. If $|\mathcal{N}_{\text{in}}^i| \neq |\mathcal{N}_{\text{out}}^i|$ for some $i \in \mathcal{N}$, then \mathcal{G} is unbalanced.

In a synchronous NAG, all players are synchronized to update their actions per iteration. In our Asy-NAG, however, players are free to update without waiting for others by only using available information from in-neighbors, which brings significant challenges to the algorithm design and analysis.

3 The Distributed Algorithm for the Asy-NAG

In this section, we first introduce some basic conditions for the existence of an NE of the AG and formulate it as a variational inequality problem. Then, we propose a distributed algorithm for the Asy-NAG where each player asynchronously maintains an estimate of the aggregate action by using information from its in-neighbors, and updates its local action by performing a projected pseudo-gradient step.

3.1 Variation Inequality Formulation of the AG

The following condition is widely adopted in the context of AGs (Belgioioso et al. 2020, Liu et al. 2020).

Assumption 1. *For each $i \in \mathcal{N}$, the action set \mathcal{X}_i is convex and compact. Each local objective function $f_i(\mathbf{x}_i, \mathbf{z})$ is continuously differentiable in $(\mathbf{x}_i, \mathbf{z})$ over some open set containing $\mathcal{X}_i \times \bar{\mathcal{X}}$ where $\bar{\mathcal{X}} = \frac{1}{n} \sum_{i=1}^n \mathcal{X}_i$. Moreover, $f_i(\mathbf{x}_i, \frac{1}{n} \mathbf{x}_i + \bar{\mathbf{x}}_{-i})$ is convex in \mathbf{x}_i over \mathcal{X}_i for any fixed $\bar{\mathbf{x}}_{-i} = \frac{1}{n} \sum_{j \neq i} \mathbf{x}_j$.*

Under Assumption 1, the AG in (1) is equivalent to solving a variational inequality problem (Facchinei & Pang 2007, Proposition 1.4.2) which is to determine an $\mathbf{x}^* \in \mathcal{X} := \prod_{i=1}^n \mathcal{X}_i$ such that

$$(\mathbf{x} - \mathbf{x}^*)^\top \phi(\mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \quad (2)$$

where \mathcal{X} and ϕ are defined as

$$\mathcal{X} := \prod_{i=1}^n \mathcal{X}_i, \quad \phi(\mathbf{x}) := \begin{bmatrix} \nabla_{\mathbf{x}_1} f_1(\mathbf{x}_1, \bar{\mathbf{x}}) \\ \vdots \\ \nabla_{\mathbf{x}_n} f_n(\mathbf{x}_n, \bar{\mathbf{x}}) \end{bmatrix}$$

and $\mathbf{x} := [\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top]^\top \in \mathcal{X}$. The mapping ϕ is also called a pseudo-gradient mapping. Let

$$F_i(\mathbf{x}_i, \mathbf{z}) = \nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i, \mathbf{z}) + \frac{1}{n} \nabla_{\mathbf{z}} f_i(\mathbf{x}_i, \mathbf{z}), \quad \forall i \in \mathcal{N}$$

and

$$F(\mathbf{x}, \mathbf{z}) := \begin{bmatrix} F_1(\mathbf{x}_1, \mathbf{z}) \\ \vdots \\ F_n(\mathbf{x}_n, \mathbf{z}) \end{bmatrix}.$$

Clearly, $\phi(\mathbf{x}) = F(\mathbf{x}, \bar{\mathbf{x}})$.

Assumption 2 (Strict monotonicity). *The mapping $\phi(\mathbf{x})$ is strictly monotone over \mathcal{X} in the sense that*

$$(\phi(\mathbf{x}) - \phi(\mathbf{x}'))^\top (\mathbf{x} - \mathbf{x}') > 0, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad \mathbf{x} \neq \mathbf{x}'.$$

Assumption 3 (Lipschitz continuity). *Each mapping $F_i(\mathbf{x}_i, \mathbf{z})$ is uniformly Lipschitz continuous over $\mathcal{X}_i \times \bar{\mathcal{X}}$, i.e., there exists some $L > 0$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_i$ and $\mathbf{z}, \mathbf{z}' \in \bar{\mathcal{X}}$,*

$$\|F_i(\mathbf{x}_i, \mathbf{z}) - F_i(\mathbf{x}'_i, \mathbf{z}')\| \leq L \left\| \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \begin{bmatrix} \mathbf{x}' \\ \mathbf{z}' \end{bmatrix} \right\|.$$

Proposition 1 (Koshal et al. 2016). *Under Assumptions 1 and 2, the AG in (1) has a unique Nash equilibrium.*

Algorithm 1 The distributed algorithm for the Asy-NAG — from the view point of player i

Initialization

- (i) Randomly select an action $\mathbf{x}_i \in \mathcal{X}_i$.
- (ii) Let $\mathbf{v}_i = \mathbf{x}_i$, $\mathbf{v}'_i = \mathbf{v}_i/d_i$ and $y_i = 1$, $y'_i = y_i/d_i$.
- (iii) Create local buffers \mathcal{V}_i , \mathcal{Y}_i and \mathcal{L}_i .
- (iv) Let $l_i = 0$ and broadcast \mathbf{v}'_i , y'_i and l_i to its out-neighbors $\mathcal{N}_{\text{out}}^i$.

Repeat

- Keep receiving \mathbf{v}'_j , y'_j and l_j from each in-neighbor player $j \in \mathcal{N}_{\text{in}}^i$, and store them into \mathcal{V}_i , \mathcal{Y}_i and \mathcal{L}_i , respectively.
- If player i is activated for an update, it computes
 - (a) $\mathbf{w}_i \leftarrow \text{sum}(\mathcal{V}_i)$, $y_i \leftarrow \text{sum}(\mathcal{Y}_i)$, $\mathbf{z}_i \leftarrow \mathbf{w}_i/y_i$,
 - (b) $l'_i \leftarrow \max(\mathcal{L}_i)$, $\alpha_i \leftarrow \sum_{t=l_i}^{l'_i} \rho(t)$,
 - (c) $\mathbf{x}_i^- \leftarrow \mathbf{x}_i$, $\mathbf{x}_i \leftarrow \Pi_{\mathcal{X}_i}[\mathbf{x}_i - \alpha_i F_i(\mathbf{x}_i, \mathbf{z}_i)]$,
 - (d) $\mathbf{v}_i \leftarrow \mathbf{w}_i + \mathbf{x}_i - \mathbf{x}_i^-$,
 - (e) $\mathbf{v}'_i \leftarrow \mathbf{v}_i/d_i$, $y'_i \leftarrow y_i/d_i$,
 - (f) $l_i \leftarrow l'_i + 1$.
- Broadcast \mathbf{v}'_i , y'_i and l_i to every out-neighbors $\mathcal{N}_{\text{out}}^i$, and empty \mathcal{V}_i , \mathcal{Y}_i , \mathcal{L}_i .

Until the stopping criterion is satisfied.

3.2 Distributed Algorithm for the Asy-NAG

In this subsection, we propose a push-sum-based distributed algorithm for the Asy-NAG in Algorithm 1 where $d_i := |\mathcal{N}_{\text{out}}^i| \leq n$ is the out-degree of player i and α_i is the effective stepsize for the local optimization of player i . The positive sequence $\{\rho(t)\}$ is used to dynamically adjust the aggressive stepsize α_i .

The idea of Algorithm 1 is very natural. As the synchronous push-sum algorithm (Nedić & Olshevsky 2014) for distributed optimization problems, Line (ii) in the initialization step is used to solve the unbalancedness issue of the directed network \mathcal{G} . However, three local buffers $\mathcal{V}_i, \mathcal{Y}_i, \mathcal{L}_i$ in Line (iii) are novelly designed to handle the asynchronicity of the Asy-NAG, which is the striking difference from the synchronous NAG.

In the repeat step, each player keeps receiving (possibly delayed) information from its in-neighbors, and store them into the corresponding buffers. Due to asynchronous updates of the Asy-NAG, player i may have stored zero, one or multiple receptions from a single neighboring player in each buffer when it is *locally* activated to update, which is not the case for the gossip-based NAG (Koshal et al. 2016, Salehisadaghiani & Pavel 2016, Cenedese et al. 2020). Instead of only using the latest reception from buffers, all data in buffers will be used to compute a new update, see Lines (a)-(f), where $\text{sum}(\cdot)$ and $\max(\cdot)$ take, respectively, summation and maximization over all elements in the input set. Then, it broadcasts the updated vectors to its out-neighbors and empties the buffers. This implies that the number of receptions in each buffer is usually limited. Interestingly, buffers are essentially not needed in practice since both summation and maximization in Lines (a)-(b) can be done recursively. For example, we can simply keep l'_i and only update its value to l_j if a new $l_j > l'_i$ has been received from an in-neighbor.

Our key idea for the Asy-NAG is essentially captured in Lines (a)-(b). Informally speaking, l_i is designed to roughly track the number of updates in each player and l'_i returns the maximum number of updates among its in-neighbors. If player i finds itself update too slow, e.g., the gap between l_i and l'_i is large, it increases its stepsize α_i to compensate for slow updates. Such an aggressive scheme is in sharp

contrast with the existing technique to reduce the stepsize of the fast player, hoping to achieve a faster convergence rate, and was initially proposed in our previous work (Zhang & You 2020) for solving the distributed optimization problem. The stopping criterion can be chosen such that the update of the local action and aggregate estimate remain small for a number of consecutive iterations.

The rest of this paper is devoted to proving the convergence of Algorithm 1 to an NE of the Asy-NAG. We first develop an augmented network approach to address the asynchronicity and information delays in Algorithm 1, under which we obtain a synchronous coordinate-wise update over the virtual network. To prove its convergence to an NE, we then propose a coordinate pseudo-gradient algorithm, which generalizes Algorithm 1 and is of independent interest. Indeed, the fully asynchronous setting has been adopted for distributed optimization (Zhang & You 2020) and the block coordinate optimization (Hannah et al. 2018), which however cannot be used for the AG as each player needs to minimize an individual objective function.

4 An Augmented Network Approach for the Asy-NAG

In this section, we construct an augmented network where a sequence of virtual nodes is introduced for each player to model the delayed information, based on which Algorithm 1 appears to be synchronous over the virtual network. Note that the proofs in Koshal et al. (2016), Salehisadaghiani & Pavel (2016), Cenedese et al. (2020) cannot be directly applied here.

Assumption 4.

- (a) (**Strong connectivity**) \mathcal{G} is strongly connected, i.e., there is a directed path between any two players.
- (b) (**Bounded information delays**) For any edge $(i, j) \in \mathcal{E}$, the time-varying information delay from player i to player j is uniformly bounded by a positive constant $\tau > 0$.
- (c) (**Bounded activation intervals**) Let t_i and t_i^+ be two consecutive activation times of player i ; then, there exist $\underline{\tau} > 0$ and $\bar{\tau} < \infty$ such that $\underline{\tau} \leq |t_i - t_i^+| \leq \bar{\tau}$.

The strong connectivity ensures each player to participate in the decision making process. The bounded communication delays assumption is common in the literature, see e.g., Yi & Pavel (2020), Lei et al. (2020). The bounded activation interval is practically satisfied and is almost necessary, e.g., if $\bar{\tau} = \infty$, it suggests that some player(s) does(do) not participate in the decision-making process. The above constants are introduced for theoretical analysis and are not known by any player.

Under Assumption 4(c), the activation time instances for updating must be discrete. Thus, we let $\mathcal{T} = \{t(k)\}_{k \geq 0}$ be an increasing sequence of the activation time instances of all players, i.e., $t \in \mathcal{T}$ if and only if there is at least one player being activated at time t . Let $\mathcal{T}_i \subseteq \mathcal{T}$ be the set of activation times of player i . Lemma 1 is key to the construction of the augmented network.

Lemma 1 (Zhang & You 2020). *The following statements hold.*

- (a) Under Assumption 4(b), letting $b_1 = (n - 1)\lfloor \bar{\tau}/\underline{\tau} \rfloor + 1$, each player is activated at least once within the time interval $(t(k), t(k + b_1)]$. Moreover, let $\mathcal{G}(k) = (\mathcal{N}, \mathcal{E}(k))$ be the activated network at $t(k)$, where $(i, j) \in \mathcal{E}(k)$ if $t(k) \in \mathcal{T}_i$ and $(i, j) \in \mathcal{E}$; then the union of networks $\bigcup_{t=k}^{k+b_1} \mathcal{G}(t)$ is strongly connected for any k .
- (b) Under Assumptions 4(b) and (c), letting $b_2 = n\lfloor \tau/\underline{\tau} \rfloor + 1$ and $b = b_1 + b_2$, the message sent from player i at $t(k)$ can be received by player j before $t(k + b_2)$ and used for computing an update before $t(k + b)$ for any k and $(i, j) \in \mathcal{E}$.
- (c) Under Assumption 4, and with b defined as in (b) above, $|l_i(k) - l_j(k)| \leq nb$ and $0 \leq l_i(k+1) - l_i(k) \leq nb + 1$ for any $i, j \in \mathcal{N}$ and k , where $l_i(k)$ is the value of l_i at $t(k)$.

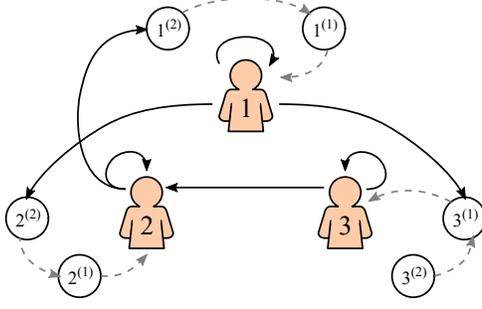


Figure 3: An augmented network of Fig. 1 with $b = 2$ at $t(k)$. Player 3's message will be received by player 2 during the time interval $(t(k), t(k+1)]$, and $(3, 2) \in \mathcal{E}(k)$. Player 3 will use information from $3^{(1)}$ and itself to compute update at $t(k+1)$. Suppose that player 2 sends message to player 1 without delay but player 1 will not be activated until $t(k+3)$; then $(2, 1^{(2)}) \in \mathcal{E}(k)$. The same case applies to player 1.

To construct an augmented network, we introduce b virtual nodes for each player. For player i , let $\{i^{(1)}, \dots, i^{(b)}\}$ be the corresponding sequence of virtual nodes. Denote by $\tilde{\mathcal{N}}$ the set containing all players and virtual nodes. As for the edge set $\tilde{\mathcal{E}}(k)$, edges $(i^{(1)}, i)$, $(i^{(2)}, i^{(1)})$, \dots , $(i^{(b)}, i^{(b-1)})$ are always included in $\tilde{\mathcal{E}}(k)$ for all i . If player i is activated to update at $t(k)$, then one of the edges (i, j) , $(i, j^{(1)})$, $(i, j^{(2)})$, \dots , $(i, j^{(b)})$ is included in $\tilde{\mathcal{E}}$, depending on the information delay between players i and j at $t(k)$. Specifically, if the information from player i is received by player j during $(t(k+s), t(k+s+1)]$ for some $s \in \{0, \dots, b\}$, then $(i, j^{(s)}) \in \tilde{\mathcal{E}}(k)$. Note that we have adopted the convention $j^{(0)} = j$. Then, $\tilde{\mathcal{G}}(k) = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}}(k))$ is a time-varying augmented network of \mathcal{G} at $t(k)$. A scenario, related to the network of Fig. 1, is depicted in Fig. 3.

Now, we reformulate Algorithm 1 as a synchronous algorithm over the time-varying virtual networks $\{\tilde{\mathcal{G}}(k)\}$. We denote the latest value of \mathbf{x}_i before time $t(k)$ by $\mathbf{x}_i(k)$. We adopt similar notations for \mathbf{v}_i , \mathbf{w}_i , \mathbf{z}_i , y_i and l_i . Letting $\mathcal{N}^{(u)} = \{i^{(u)} | i \in \mathcal{N}\}$ for $u = 1, 2, \dots, b$, we enumerate the players before virtual nodes, e.g., $\tilde{\mathcal{N}} = \{\mathcal{N}, \mathcal{N}^{(1)}, \dots, \mathcal{N}^{(b)}\}$. Let $\tilde{n} = (b+1)n$. Every virtual node i maintains $\mathbf{v}_i(k)$, $\mathbf{w}_i(k)$ and $y_i(k)$. Let the augmented state vector be

$$\begin{aligned}\tilde{\mathbf{V}}(k) &= [\mathbf{v}_1(k), \dots, \mathbf{v}_n(k), \mathbf{v}_{n+1}(k), \dots, \mathbf{v}_{\tilde{n}}(k)]^\top, \\ \tilde{\mathbf{W}}(k) &= [\mathbf{w}_1(k), \dots, \mathbf{w}_n(k), \mathbf{w}_{n+1}(k), \dots, \mathbf{w}_{\tilde{n}}(k)]^\top, \\ \tilde{\mathbf{y}}(k) &= [y_1(k), \dots, y_n(k), y_{n+1}(k), \dots, y_{\tilde{n}}(k)]^\top,\end{aligned}$$

with $\mathbf{v}_i(0) = \mathbf{0}_p$ and $y_i(0) = 0$ for $i = n+1, \dots, \tilde{n}$.

Then, Algorithm 1 is compactly written as

$$\begin{aligned}\tilde{\mathbf{W}}(k+1) &= \tilde{\mathbf{A}}(k)\tilde{\mathbf{V}}(k), \\ \tilde{\mathbf{y}}(k+1) &= \tilde{\mathbf{A}}(k)\tilde{\mathbf{y}}(k), \\ \mathbf{z}_i(k+1) &= \frac{\mathbf{w}_i(k+1)}{y_i(k+1)}, \quad \forall i \in \mathcal{N} \\ \mathbf{x}_i(k+1) &= \begin{cases} \Pi_{\mathcal{X}_i} [\mathbf{x}_i(k) - \alpha_i(k)F_i(\mathbf{x}_i(k), \mathbf{z}_i(k+1))], & \text{if } i \in \mathcal{N} \text{ and } t(k) \in \mathcal{T}_i, \\ \mathbf{x}_i(k), & \text{if } i \in \mathcal{N} \text{ and } t(k) \notin \mathcal{T}_i, \end{cases} \\ \tilde{\mathbf{V}}(k+1) &= \tilde{\mathbf{W}}(k+1) + \Delta X(k),\end{aligned}\tag{3}$$

where

$$\Delta X(k) = [\mathbf{x}_1(k+1) - \mathbf{x}_1(k), \dots, \mathbf{x}_n(k+1) - \mathbf{x}_n(k), \mathbf{0}_p, \dots, \mathbf{0}_p]^\top.$$

$\tilde{A}(k)$ is given as

$$[\tilde{A}(k)]_{ij} = \begin{cases} 1/d_j, & \text{if } j \in \mathcal{N}, t(k) \in \mathcal{T}_j, i = un + i', \tau_{ji'}(k) = u \\ 1, & \text{if } j \in \mathcal{N}, t(k) \notin \mathcal{T}_j, \text{ and } i = j \\ 1, & \text{if } j \notin \mathcal{N}, \text{ and } i = j - n \\ 0, & \text{otherwise} \end{cases}$$

where $d_j = |\mathcal{N}_{\text{out}}^j|$ is the out-degree of player j and $\tau_{ji'}(k) \geq 0$ is the information delay from player j to node i' at $t(k)$. If each player can have immediate access to its own action, then $\tau_{jj}(k) = 0$ for all $j \in \mathcal{N}$ and $k \geq 0$. If $(j, i) \notin \mathcal{E}$, we have $\tau_{ji}(k) = \infty$ for all $i \in \mathcal{N}$. Note that $\tilde{A}(k)$ is column-stochastic and $[\tilde{A}(k)]_{ii} > 0$ for all $i \in \mathcal{N}$ and $k \geq 0$. Thus, $y_i(k) > 0$ always holds and the division in (3) is well-defined.

5 The Perturbed Coordinate Pseudo-gradient Algorithm

Inspired by the block-coordinate method for a single objective function (Nesterov 2012), we first propose a novel perturbed coordinate pseudo-gradient algorithm (PCPA) which allows partial updates of players via perturbed pseudo-gradients. Subsequently, we prove its convergence, the results being not only of independent interest, but also critical to the proof of the convergence of Algorithm 1 in the next section.

Consider the AG in (1) with a single player to update with a perturbed pseudo-gradient per iteration. We denote the sequence of updating players by $\{s(0), s(1), \dots\}$, where $s(k) \in \mathcal{N}$, i.e., player $s(k)$ is selected to update at the k th iteration. An integer-valued sequence $\{r_i(k)\}$ is defined recursively, i.e.,

$$r_i(k+1) = \begin{cases} r_i(k) + \Delta r_i(k), & \text{if } s(k) = i, \\ r_i(k), & \text{otherwise,} \end{cases}$$

and $r_i(0) = 0$ for all $i \in \mathcal{N}$. In light of the above sequence, we propose the following PCPA

$$\mathbf{x}_i(k+1) = \begin{cases} \Pi_{\mathcal{X}_i} \left[\mathbf{x}_i(k) - \alpha_i(k) \hat{F}_i(k) \right], & \text{if } s(k) = i, \\ \mathbf{x}_i(k), & \text{otherwise,} \end{cases} \quad (4)$$

where

$$\begin{aligned} \alpha_i(k) &= \sum_{t=r_i(k)}^{r_i(k+1)-1} \rho(t), \\ \hat{F}_i(k) &= F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)) + \boldsymbol{\epsilon}_i(k), \end{aligned}$$

and $\boldsymbol{\epsilon}_i(k)$ is the perturbation of the gradient in (3.1). We make the following assumption.

Assumption 5.

- (a) There exists an integer $\sigma_1 \geq 0$ such that $\mathcal{N} \subseteq \{s(k+1), \dots, s(k+\sigma_1)\}$.
- (b) There exists an integer $\sigma_2 \geq 0$ such that $|r_i(k) - r_j(k)| \leq \sigma_2$ and $\Delta r_i(k) \leq \sigma_2$ for all $i, j \in \mathcal{N}$ and $k \geq 0$.
- (c) The sequence $\{\rho(t)\}$ is chosen to satisfy that $0 < \rho(t+1) \leq \rho(t)$, $\sum_{t=0}^{\infty} \rho(t) = \infty$, $\sum_{t=0}^{\infty} \rho^2(t) < \infty$.
- (d) $\sum_{k=0}^{\infty} \rho(k) \|\boldsymbol{\epsilon}_i(k)\| < \infty$ for all $i \in \mathcal{N}$.

Assumption 5(a) guarantees that each player updates at least once in a bounded time interval, which is different from the randomized block-coordinated method (Richtárik & Takáč 2014). Assumption 5(b) ensures the boundedness of the updating time difference between two players. It is obvious that Assumptions 5(a)-(b) hold for the Asy-NAG with $\sigma_1 = b_1$ and $\sigma_2 = nb + 1$, where b_1 and b are defined in Lemma 1. Unlike Yousefian et al. (2013) and Bottou (2010), the perturbation $\boldsymbol{\epsilon}_i(k)$ is not stochastic. Instead, it should be controlled by non-increasing sequences in Assumptions 5(c)-(d), which is essential to the convergence of the PCPA.

Proposition 2. *Suppose that Assumptions 1-3 and 5 hold. Then, the sequence $\{\mathbf{x}_i(k)\}$, $i = 1, \dots, n$, generated by the PCPA in (4) converges to the unique NE of the AG in (1).*

Proof. See Appendix A. □

6 Convergence of Algorithm 1

Under a non-increasing sequence $\{\rho(t)\}$ in Assumption 5(c), we show that each player is able to asymptotically track the aggregate action $\bar{\mathbf{x}}(k)$.

Lemma 2. *Under Assumptions 1, 3, 4 and 5(c), let $\{\mathbf{z}_i(k)\}$, $i = 1, \dots, n$, be the sequence generated by the augmented network formulation (3) of Algorithm 1. Then, the following statements hold.*

(a) For all $k \geq 0$,

$$\|\mathbf{z}_i(k+1) - \bar{\mathbf{x}}(k)\| \leq B \left(\lambda^k n M + \sum_{t=0}^k \lambda^{k-t} \Delta(t) \right),$$

where $\Delta(t) = \sum_{i=1}^n \|\mathbf{x}_i(t+1) - \mathbf{x}_i(t)\|$ and the constants are defined as $B = 8n^{nb}(1+n^{nb})$, $\lambda = (1 - \frac{1}{n^{nb}})^{\frac{1}{nb}}$, and $M = \max_{i \in \mathcal{N}, \mathbf{x}_i \in \mathcal{X}_i} \|\mathbf{x}_i\|$.

(b) $\lim_{k \rightarrow \infty} \|\mathbf{z}_i(k+1) - \bar{\mathbf{x}}(k)\| = 0$, $\forall i \in \mathcal{N}$.

(c) $\sum_{k=0}^{\infty} \rho(k) \|\mathbf{z}_i(k+1) - \bar{\mathbf{x}}(k)\| < \infty$, $\forall i \in \mathcal{N}$.

Proof. See Appendix B. □

The convergence proof of Algorithm 1 then directly follows from Proposition 2.

Proposition 3. *Suppose that Assumptions 1-4 hold and the sequence $\{\rho(t)\}$ is chosen to satisfy Assumption 5(c). Then, the sequence $\{\mathbf{x}_i(k)\}$, $i = 1, \dots, n$, generated by the augmented network formulation (3) of Algorithm 1 converges to the unique NE of the AG in (1).*

Proof. Let

$$\boldsymbol{\epsilon}_i(k) = F_i(\mathbf{x}_i(k), \mathbf{z}_i(k+1)) - F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)).$$

Under the Lipschitz continuity of F_i , we have

$$\|\boldsymbol{\epsilon}_i(k)\| \leq L \|\mathbf{z}_i(k+1) - \bar{\mathbf{x}}(k)\|.$$

Thus, it follows from Lemma 2 that $\lim_{k \rightarrow \infty} \|\boldsymbol{\epsilon}_i(k)\| = 0$ and $\sum_{k=0}^{\infty} \rho(k) \|\boldsymbol{\epsilon}_i(k)\| < \infty$. Let $\mathcal{A}(k)$ denote the set of updating players at time $t(k)$, i.e., $\mathcal{A}(k) = \{i | t(k) \in \mathcal{T}_i, i \in \mathcal{N}\}$. It follows from Algorithm 1 that

$$\mathbf{x}_i(k+1) = \begin{cases} \Pi_{\mathcal{X}_i} \left[\mathbf{x}_i(k) - \alpha_i(k) \widehat{F}_i(k) \right], & \text{if } i \in \mathcal{A}(k), \\ \mathbf{x}_i(k), & \text{otherwise,} \end{cases} \quad (5)$$

where $\alpha_i(k) = \sum_{t=l_i(k)}^{l_i(k+1)-1} \rho(t)$, and the perturbed gradient $\widehat{F}_i(k) = F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)) + \boldsymbol{\epsilon}_i(k)$. Then, two cases are considered.

Case 1: If $\mathcal{A}(k)$ is a singleton for all $k \geq 0$, it follows from Proposition 2 that $\mathbf{x}_i(k)$ converges to the NE.

Case 2: If for some k , $\mathcal{A}(k)$ includes multiple elements, say i_1, \dots, i_m , where $m = |\mathcal{A}(k)| \leq n$, we can incrementally select the players via m iterations.

Specifically, letting $\mathbf{x}^{(0)} = \mathbf{x}(k)$ and $\mathbf{x}^{(m)} = \mathbf{x}(k+1)$, we can rewrite (5) as

$$\mathbf{x}_i^{(u)} = \begin{cases} \Pi_{\mathcal{X}_i} \left[\mathbf{x}_i^{(u-1)} - \alpha_i(k) \widehat{F}_i^{(u-1)} \right], & \text{if } i = i_u \\ \mathbf{x}_i^{(u-1)}, & \text{otherwise} \end{cases}$$

for $u = 1, \dots, m$, where

$$\begin{aligned}\widehat{F}_i^{(u-1)} &= F_i(\mathbf{x}_i^{(u-1)}, \bar{\mathbf{x}}^{(u-1)}) + \boldsymbol{\epsilon}_i^{(u-1)}, \\ \boldsymbol{\epsilon}_i^{(u-1)} &= \boldsymbol{\epsilon}_i(k) + F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)) - F_i(\mathbf{x}_i^{(u-1)}, \bar{\mathbf{x}}^{(u-1)}).\end{aligned}$$

This implies that

$$\begin{aligned}& \|F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)) - F_i(\mathbf{x}_i^{(u-1)}, \bar{\mathbf{x}}^{(u-1)})\| \\ & \leq \|F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k)) - F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}^{(u-1)})\| + \|F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}^{(u-1)}) - F_i(\mathbf{x}_i^{(u-1)}, \bar{\mathbf{x}}^{(u-1)})\| \\ & \leq \frac{L}{n} \sum_{s=1}^{u-1} \|\mathbf{x}_{i_s}(k+1) - x_{i_s}(k)\| + L \|\mathbf{x}_{i_{u-1}}(k+1) - x_{i_{u-1}}(k)\|\end{aligned}$$

for $u = 2, \dots, m$. Moreover, it holds that

$$\|\mathbf{x}_{i_s}(k+1) - \mathbf{x}_{i_s}(k)\| \leq \alpha_{i_s}(k) \|\widehat{F}_i^{(s-1)}\| \leq \alpha_{i_s}(k)(C + \boldsymbol{\epsilon}_i(k)),$$

where $C = \max_{\mathbf{x} \in \mathcal{X}} \max_{i \in \mathcal{N}} F_i(\mathbf{x}_i, \bar{\mathbf{x}}) < \infty$ due to the compactness of \mathcal{X} . Thus, it is easy to verify that $\sum_{k=0}^{\infty} \rho(k) \|\sum_u \boldsymbol{\epsilon}_i^{(u)}(k)\| < \infty$. Finally, the convergence of Algorithm 1 follows from Proposition 2 again. \square

For the Asy-NAGs with strictly monotone pseudo-gradients, it does not seem to be possible to explicitly evaluate the convergence rate of Algorithm 1, and thus we resort to numerical experiments. In the next section, numerical results illustrate that such a rate is related to network connectivity and the level of asynchronicity.

Remark 1. *It is interesting to note that the convergence in Proposition 3 is in the deterministic sense, rather than in the stochastic sense as in Koshal et al. (2016), Salehisadaghiani & Pavel (2016), Cenedese et al. (2020).*

7 Numerical Experiments

In this section, we validate the performance of Algorithm 1 on networked Nash-Cournot games. The convergence to the NE is confirmed under different network topologies with different numbers of players. We also compare Algorithm 1 with its synchronous counterpart. Moreover, we validate the advantages of using the aggressive stepsize in Line (b) of Algorithm 1.

The Nash-Cournot game model is adopted from Koshal et al. (2016). Specifically, consider n firms competing over L markets. Let firm i 's production and sales at market l be denoted by g_{il} and s_{il} , respectively, while its cost of production at market l is $c_{il}(g_{il})$ and defined as

$$c_{il}(g_{il}) = a_{il}g_{il} + b_{il}g_{il}^2,$$

where a_{il} and b_{il} are positive parameters for firm i . The revenue of firm i at market l is $p_l(s_l)s_{il}$, where $p_l(s_l)$ denotes the sale price at market l and $s_l = \sum_{i=1}^n s_{il}$ is the total sales at market l . The sale price function captures the inverse demand function and is defined as

$$p_l(s_l) = d_l - s_l,$$

where d_l is the overall demand at market l . Firm i 's production at market l is capacitated by cap_{il} . The transportation costs between any two markets are set to zero. Let $\mathbf{x}_{il} = (g_{il}, s_{il})$ for all $l = 1, \dots, L$,

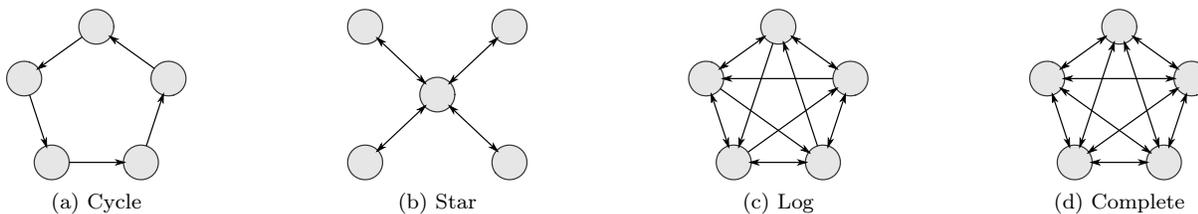


Figure 4: An illustration of network structures used in experiments.

$\mathbf{x}_i = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{iL})^\top$, and $\mathbf{x} = (\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top)^\top$, firm i 's optimization problem is then given by the following:

$$\begin{aligned} & \text{minimize} && f_i(\mathbf{x}_i, \bar{\mathbf{x}}) = \sum_{l=1}^L (c_{il}(g_{il}) - s_{il} \cdot p_l(s_l)) \\ & \text{subject to} && g_{il}, s_{il} \geq 0, g_{il} \leq \text{cap}_{il}, \\ & && \sum_{l=1}^L g_{il} = \sum_{l=1}^L s_{il}. \end{aligned}$$

Let \mathcal{X}_i denote the constraint set for firm i . Obviously, for each i , both \mathcal{X}_i and f_i satisfy Assumption 1, and the pseudo-gradient mapping ϕ satisfies Assumptions 2 and 3. Thus, it follows from Proposition 1 that the Nash-Cournot game admits a unique NE. Besides, ϕ is also *strongly monotone*¹, i.e., there exists a constant $\mu > 0$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$,

$$(\mathbf{x} - \mathbf{x}')^\top (\phi(\mathbf{x}) - \phi(\mathbf{x}')) \geq \mu \|\mathbf{x} - \mathbf{x}'\|^2.$$

The Nash-Cournot game consists of $n \in \{5, 10, 20, 30\}$ firms over $L = 10$ markets. The parameters a_{il} , b_{il} , and d_l are drawn from a uniform distribution. Specifically, for all $i \in \mathcal{N}$ and $l \in \{1, \dots, L\}$, we set $a_{il} \sim U(2, 12)$, $b_{il} \sim U(2, 3)$, where $U(u_1, u_2)$ denotes the uniform distribution over an interval $[u_1, u_2]$. The demand d_l is drawn from $U(90, 100)$. Moreover, we set the production capacities as $\text{cap}_{il} = 500$.

We adopt the Message Passing Interface (MPI) on a multi-core server to simulate the P2P network. Particularly, the MPI uses n cores to denote the players, and the communication is performed between neighboring cores in the predefined network. To simulate a heterogeneous environment, the computation time of player i is sampled from an exponential distribution $\exp(1/\mu_i)$ ms. For player i , μ_i is set as $1 + |\bar{\mu}_i|$ where $\bar{\mu}_i$ follows the standard normal distribution $N(0, 5^2)$. The information delays are sampled from $\exp(1/5)$ ms.

In the figures to follow, we plot the trajectories of the normalized suboptimality gap

$$\frac{\|\mathbf{x}(t) - \mathbf{x}^*\|_\infty}{\|\mathbf{x}^*\|_\infty}.$$

The curves are averaged over 50 Monte-Carlo simulations with randomly initiated points and sampled delays.

Firstly, we consider an Asy-NAG with $n = 20$ players over four network structures which are described below and illustrated in Figure 4.

- *Cycle*: every player has only one in-neighbor and one out-neighbor.
- *Log*: player i sends information to players $\text{mod}(2^j + i, n) + 1$, $0 \leq j < \log_2 n$.
- *Star*: a player is able to send and receive information from any other player.

¹In the experiments, let $\rho(t) = \rho$ to achieve a linear convergence rate, which is to be studied in the future work.

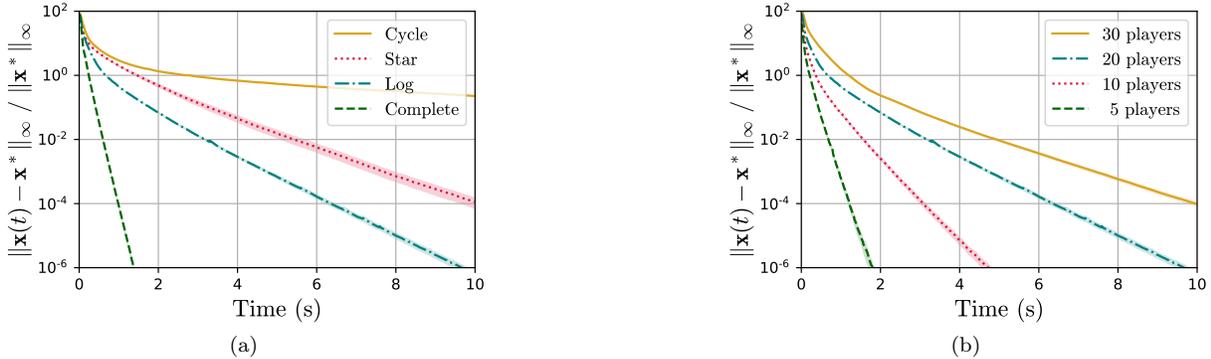


Figure 5: The suboptimality gap versus the running time. (a) The trajectories of the suboptimality gap for Algorithm 1 with 20 players over different network structures. (b) The trajectories of the suboptimality gap for Algorithm 1 with different numbers of players over the *log* network.

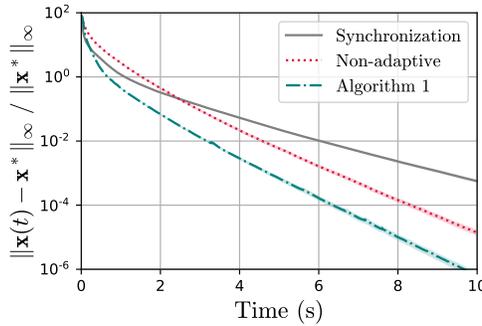


Figure 6: The suboptimality gap versus the running time. Comparison of Algorithm 1, the asynchronous algorithm without an aggressive stepsize scheme (Non-adaptive), and the synchronous algorithm with 20 players over the *log* network.

- *Complete*: every player sends information to every other player.

Figure 5(a) confirms the convergence of Algorithm 1 to the NE over different network structures. Specifically, the convergence rate over the *complete* network is the fastest while that over the *cycle* is the slowest, which coincides with the fact that the *complete* network has the best connectivity. Note that although the *star* network seems denser than the *log* one when $n = 20$, the algorithm is bottlenecked by the central player, thus the convergence rate is slower.

Secondly, we consider Asy-NAGs with $n = 5, 10, 20, 30$ players over the *log* network. Figure 5 demonstrates the convergence of Algorithm 1. The asynchronicity of the Asy-NAG increases with the number of players, which decelerates the convergence rate of the estimated aggregate action. Thus, Algorithm 1 requires more time to converge when the number of players increases.

Finally, we show the effectiveness of the aggressive stepsize scheme over the *log* network with $n = 20$ players in Figure 6. While Algorithm 1 converges faster than the one without an aggressive stepsize scheme, both algorithms outperform their synchronous version.

8 Conclusion

In this paper, we have introduced Asy-NAG, which is challenging due to the full asynchrony of computation and communication. For the AG, we have proposed a distributed algorithm and proved its

convergence to a Nash equilibrium. It leverages two key ideas: an asynchronous push-sum algorithm to dynamically track the aggregate action, and an aggressive stepsize scheme to facilitate the convergence. Moreover, we have developed an augmented system approach to address the asynchronicity and designed a novel perturbed coordinated algorithm to help prove the convergence. Theoretical results have been validated by numerical experiments.

Future work will consider the algorithm design for the Asy-NAGs with linear coupling constraints and the fully asynchronous non-aggregative games. The linear convergence rate for the Asy-NAGs with strongly monotone pseudo-gradients is also an open problem.

A Proof of Proposition 2

Before the proof of Proposition 2, we state and prove a lemma on the convergence of a deterministic sequence.

Lemma 3. *Let $\{x_k\}$, $\{u_k\}$, $\{w_k\}$ and $\{v_k\}$ be four scalar sequences such that*

$$x_{k+1} \leq x_k - u_k + w_k + v_k, \quad k = 0, 1, \dots \quad (6)$$

where $\{x_k\}$, $\{u_k\}$, $\{w_k\}$ are nonnegative, $\sum_{k=0}^{\infty} w_k < \infty$ and $\sum_{k=0}^{\infty} v_k$ exists. Then, $\{x_k\}$ converges and $\sum_{k=0}^{\infty} u_k \leq \infty$.

Proof. The proof is motivated by that of Bertsekas (2015, Proposition A.4.4). Using the nonnegativity of $\{u_k\}$, we have

$$x_{k+1} \leq x_k + w_k + v_k \leq x_{\bar{k}} + \sum_{l=\bar{k}}^k (w_l + v_l),$$

where $\bar{k} \leq k$. By taking upper limit of both sides as $k \rightarrow \infty$, we have that

$$\limsup_{k \rightarrow \infty} x_k \leq x_{\bar{k}} + \sum_{l=\bar{k}}^{\infty} w_l + \sum_{l=\bar{k}}^{\infty} v_l. \quad (7)$$

Since $\sum_{k=0}^{\infty} w_k < \infty$ and $\sum_{k=0}^{\infty} v_k$ exists, it follows that the left-hand side of (7) is bounded above and that $\lim_{\bar{k} \rightarrow \infty} \sum_{l=\bar{k}}^{\infty} v_l = 0$. By taking the lower limit of the right-hand side (RHS) as $\bar{k} \rightarrow \infty$, we have

$$0 \leq \limsup_{k \rightarrow \infty} x_k \leq \liminf_{\bar{k} \rightarrow \infty} x_{\bar{k}} < \infty.$$

This implies that $\{x_k\}$ converges to a finite value. By writing (6) for the index k set to $0, \dots, k$, and adding, we have

$$\sum_{l=0}^k u_k \leq x_0 + \sum_{l=0}^k w_l + \sum_{l=0}^k v_l - x_{k+1}.$$

Thus, by taking the limit as $k \rightarrow \infty$, we have $\sum_{l=0}^{\infty} u_l < \infty$. \square

Proof of Proposition 2: To simplify notations, let $\sigma = \sigma_1 + \sigma_2$ and $\tilde{r}(k) = \max_{j \in \mathcal{N}} r_j(k)$. It follows from Assumption 5 that every player updates at least once in σ iterations and $\tilde{r}(k+t) - \tilde{r}(k) \leq t\sigma_2 \leq t\sigma$ for any integer $t \geq 0$. Let I_i^k be an indicator function which takes value 1 when $s(k) = i$ and 0 otherwise. We abuse the notation of stepsize by $\alpha_i(k) = I_i^k \sum_{t=r_i(k)}^{r_i(k+1)-1} \rho(t)$. Then, for all $i \in \mathcal{N}$,

$$\begin{aligned} \|\mathbf{x}_i(k+\sigma) - \mathbf{x}_i^*\|^2 &= \left\| \mathbf{x}_i(k) - \mathbf{x}_i^* + \sum_{u=k}^{k+\sigma-1} [\mathbf{x}_i(u+1) - \mathbf{x}_i(u)] \right\|^2 \\ &= \|\mathbf{x}_i(k) - \mathbf{x}_i^*\|^2 + \|\mathbf{x}_i(k+\sigma) - \mathbf{x}_i(k)\|^2 \\ &\quad + 2 \sum_{u=k}^{k+\sigma-1} [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [\mathbf{x}_i(u+1) - \mathbf{x}_i(u)]. \end{aligned} \quad (8)$$

Since the action set \mathcal{X}_i is compact and the mapping F_i is continuous, $\|F_i\|$ is bounded. Besides, the perturbation $\|\epsilon_i\|$ is bounded under Assumption 5. Thus, the perturbed mapping $\|\widehat{F}_i\|$ is bounded. Let C denote its upper bound; then we have that for $u = k + 1, \dots, k + \sigma$,

$$\|\mathbf{x}_i(u) - \mathbf{x}_i(k)\| = \left\| \sum_{v=k}^{u-1} [\mathbf{x}_i(v+1) - \mathbf{x}_i(v)] \right\| \leq C \sum_{t=r_i(k)}^{r_i(u)-1} \rho(t) \leq C \sum_{t=\tilde{r}(k)-\sigma}^{\tilde{r}(k+\sigma)-1} \rho(t).$$

Thus, it holds that

$$\tilde{\rho}_{1i}(k) := \|\mathbf{x}_i(k + \sigma) - \mathbf{x}_i(k)\|^2 \leq C^2 \left[\sum_{t=\tilde{r}(k)-\sigma}^{\tilde{r}(k+\sigma)-1} \rho(t) \right]^2 \leq C^2 \sigma(\sigma + 1) \sum_{t=\tilde{r}(k)-\sigma}^{\tilde{r}(k+\sigma)-1} \rho^2(t).$$

It follows from Assumption 5(a) that $\tilde{r}(k + \sigma^2 + \sigma) - \sigma \geq \tilde{r}(k + \sigma)$.

Taking summation over k , it yields that

$$\sum_{k=0}^{\infty} \tilde{\rho}_{1i}(k) \leq C^2 \sigma(\sigma + 1) \sum_{k=0}^{\infty} \sum_{t=\tilde{r}(k)-\sigma}^{\tilde{r}(k+\sigma)-1} \rho(t)^2 \leq C^2 \sigma^2 (\sigma + 1)^2 \sum_{t=0}^{\infty} \rho(t)^2 < \infty.$$

Next, consider the last term of the RHS of (8), we obtain that

$$\begin{aligned} & \sum_{u=k}^{k+\sigma-1} [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [\mathbf{x}_i(u+1) - \mathbf{x}_i(u)] \\ & \leq \sum_{u=k}^{k+\sigma-1} [\mathbf{x}_i(u) - \mathbf{x}_i^*]^\top [\mathbf{x}_i(u+1) - \mathbf{x}_i(u)] + \sum_{u=k}^{k+\sigma-1} \|\mathbf{x}_i(k) - \mathbf{x}_i(u)\| \|\mathbf{x}_i(u+1) - \mathbf{x}_i(u)\|. \end{aligned} \quad (9)$$

Denote the last term of the RHS of (9) by $\tilde{\rho}_{2i}(k)$. Similarly, we have $\sum_{k=0}^{\infty} \tilde{\rho}_{2i}(k) < \infty$. Consider the first term, when $I_i^u = 1$; we then have that

$$\begin{aligned} & [\mathbf{x}_i(u) - \mathbf{x}_i^*]^\top [\mathbf{x}_i(u+1) - \mathbf{x}_i(u)] \\ & = -\|\mathbf{x}_i(u+1) - \mathbf{x}_i(u)\|^2 + [\mathbf{x}_i(u+1) - \mathbf{x}_i^*]^\top [\mathbf{x}_i(u+1) - \mathbf{x}_i(u) + \alpha_i(u) \widehat{F}_i(u)] \\ & \quad + \alpha_i(u) [\mathbf{x}_i(u) - \mathbf{x}_i(u+1)]^\top \widehat{F}_i(u) - \alpha_i(u) [\mathbf{x}_i(u) - \mathbf{x}_i^*]^\top \widehat{F}_i(u) \\ & \leq -\alpha_i(u) [\mathbf{x}_i(u) - \mathbf{x}_i^*]^\top \widehat{F}_i(u) + \alpha_i(u) \|\mathbf{x}_i(u) - \mathbf{x}_i(u+1)\| \|\widehat{F}_i(u)\|, \end{aligned} \quad (10)$$

where the inequality follows from the projection theorem (Bertsekas 2016, Proposition 1.1.4). Similarly, let $\tilde{\rho}_{3i}(k)$ denote the last term of (10), and we have $\sum_{k=0}^{\infty} \tilde{\rho}_{3i}(k) < \infty$.

Considering the first term of the RHS of (10), we obtain

$$\begin{aligned} & -\alpha_i(u) [\mathbf{x}_i(u) - \mathbf{x}_i^*]^\top \widehat{F}_i(u) \\ & = -\alpha_i(u) [\mathbf{x}_i(u) - \mathbf{x}_i(k)]^\top \widehat{F}_i(u) - \alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [\widehat{F}_i(u) - F_i(u)] \\ & \quad - \alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(u) - F_i(k)] - \alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*] \\ & \quad - \alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top F_i^* \\ & \leq -\alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*] \\ & \quad + C \alpha_i(u) \|\mathbf{x}_i(u) - \mathbf{x}_i(k)\| + 2M \alpha_i(u) \|F_i(u) - F_i(k)\| + 2M \alpha_i(u) \|\epsilon_i(u)\|, \end{aligned} \quad (11)$$

where we have used $F_i(u)$ and F_i^* for $F_i(\mathbf{x}_i(u), \bar{\mathbf{x}}(u))$ and $F_i(\mathbf{x}_i^*, \bar{\mathbf{x}}^*)$ for simplicity, and $M = \max_{i \in \mathcal{N}, \mathbf{x} \in \mathcal{X}_i} \|\mathbf{x}\|$. Note that with the Lipschitz continuity of the mapping F_i , we have that

$$\begin{aligned} & \|F_i(u) - F_i(k)\| \\ & \leq \|F_i(\mathbf{x}_i(u), \bar{\mathbf{x}}(u)) - F_i(\mathbf{x}_i(u), \bar{\mathbf{x}}(k))\| + \|F_i(\mathbf{x}_i(u), \bar{\mathbf{x}}(k)) - F_i(\mathbf{x}_i(k), \bar{\mathbf{x}}(k))\| \\ & \leq \frac{L}{n} \sum_{j=1}^n \|\mathbf{x}_j(u) - \mathbf{x}_j(k)\| + L \|\mathbf{x}_i(u) - \mathbf{x}_i(k)\|. \end{aligned}$$

Thus, denoting by $\tilde{\rho}_{4i}(k)$ the summation over u of the last three terms of (11), we have $\sum_{k=0}^{\infty} \tilde{\rho}_{4i}(k) < \infty$.

Finally, we evaluate the summation over i and u of the first term of the RHS of (11).

By the definition of I_i^u and $\alpha_i(u)$, we know that

$$\sum_{i=1}^n \sum_{u=k}^{k+\sigma-1} \alpha_i(u) = \sum_{i=1}^n \sum_{t=r_i(k)}^{r_i(k+\sigma)-1} \rho(t).$$

Then, it implies that

$$\begin{aligned} & \sum_{i=1}^n \sum_{u=k}^{k+\sigma-1} -\alpha_i(u) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*] \\ &= \sum_{i=1}^n \left[-\sum_{t=r_i(k)}^{\tilde{r}(k)-1} - \sum_{t=\tilde{r}(k)}^{\tilde{r}(k+\sigma)-1} + \sum_{t=r_i(k+\sigma)}^{\tilde{r}(k+\sigma)-1} \right] \times \rho(t) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*] \\ &\leq -\sum_{t=\tilde{r}(k)}^{\tilde{r}(k+\sigma)-1} \rho(t) [\mathbf{x}(k) - \mathbf{x}^*]^\top [\phi(\mathbf{x}(k)) - \phi(\mathbf{x}^*)] - \sum_{i=1}^n \sum_{t=r_i(k)}^{\tilde{r}(k)-1} \rho(t) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*] \\ &\quad + \sum_{i=1}^n \sum_{t=r_i(k+\sigma)}^{\tilde{r}(k+\sigma)-1} \rho(t) [\mathbf{x}_i(k+\sigma) - \mathbf{x}_i^*]^\top [F_i(k+\sigma) - F_i^*] \\ &\quad + \sum_{i=1}^n \sum_{t=r_i(k+\sigma)}^{\tilde{r}(k+\sigma)-1} \rho(t) \left\{ \|\mathbf{x}_i(k) - \mathbf{x}_i(k+\sigma)\| \|F_i(k) - F_i^*\| + \|\mathbf{x}_i(k+\sigma) - \mathbf{x}_i^*\| \|F_i(k) - F_i(k+\sigma)\| \right\} \\ &= -\sum_{t=\tilde{r}(k)}^{\tilde{r}(k+\sigma)-1} \rho(t) [\mathbf{x}(k) - \mathbf{x}^*]^\top [\phi(\mathbf{x}(k)) - \phi(\mathbf{x}^*)] - \delta(k) + \delta(k+\sigma) + \tilde{\rho}_5(k). \end{aligned} \tag{12}$$

Similarly, we may obtain that $\sum_{k=0}^{\infty} \tilde{\rho}_5(k) < \infty$. Besides,

$$\begin{aligned} \delta(k) &= \sum_{i=1}^n \sum_{t=r_i(k)}^{\tilde{r}(k)-1} \rho(t) [\mathbf{x}_i(k) - \mathbf{x}_i^*]^\top [F_i(k) - F_i^*], \\ |\delta(k)| &\leq \sum_{i=1}^n \sum_{t=\tilde{r}(k)-\sigma}^{\tilde{r}(k)-1} 2C\rho(t) \|\mathbf{x}_i(k) - \mathbf{x}_i^*\| \leq 4nMC\sigma\rho(\tilde{r}(k) - \sigma), \end{aligned}$$

which implies that $\delta(k)$ is bounded and $\lim_{k \rightarrow \infty} \delta(k) = 0$. Thus, it immediately holds that

$$\sum_{k=0}^{\infty} (\delta(k+\sigma) - \delta(k)) = -\sum_{k=0}^{\sigma-1} \delta(k).$$

Finally, combining (8), (9), (10), (11) and (12) yields the following key inequality

$$\|\mathbf{x}(k+\sigma) - \mathbf{x}^*\|^2 \leq \|\mathbf{x}(k) - \mathbf{x}^*\|^2 - 2 \sum_{t=\tilde{r}(k)}^{\tilde{r}(k+\sigma)-1} \rho(t) [\mathbf{x}(k) - \mathbf{x}^*]^\top [\phi(\mathbf{x}(k)) - \phi(\mathbf{x}^*)] + \tilde{\rho}(k) + \tilde{\delta}(k), \tag{13}$$

where $\tilde{\rho}(k) = \sum_{i=1}^n [\tilde{\rho}_{1i}(k) + 2 \sum_{j=2}^4 \tilde{\rho}_{ji}(k)] + 2\tilde{\rho}_5(k)$ and $\tilde{\delta}(k) = 2[\delta(k+\sigma) - \delta(k)]$ satisfying that

$$\tilde{\rho}(k) > 0, \quad \sum_{k=0}^{\infty} \tilde{\rho}(k) < \infty, \quad \sum_{k=0}^{\infty} \tilde{\delta}(k) \text{ exists.}$$

Letting $\mathbf{y}(k) = \mathbf{x}(k\sigma)$, it follows from (13) that

$$\|\mathbf{y}(k+1) - \mathbf{x}^*\|^2 \leq \|\mathbf{y}(k) - \mathbf{x}^*\|^2 - 2 \sum_{t=\tilde{r}(k\sigma)}^{\tilde{r}((k+1)\sigma)-1} \rho(t) [\mathbf{y}(k) - \mathbf{x}^*]^\top [\phi(\mathbf{y}(k)) - \phi(\mathbf{x}^*)] + \tilde{\rho}'(k) + \tilde{\delta}'(k), \quad (14)$$

where $\tilde{\rho}'(k) = \tilde{\rho}(k\sigma)$, and $\tilde{\delta}'(k) = \tilde{\delta}(k\sigma)$. We still have

$$\tilde{\rho}'(k) > 0, \quad \sum_{k=0}^{\infty} \tilde{\rho}'(k) < \infty, \quad \sum_{k=0}^{\infty} \tilde{\delta}'(k) \text{ exists.}$$

As (14) satisfies the conditions of Lemma 3, it follows that $\{\|\mathbf{y}(k) - \mathbf{x}^*\|\}$ is a convergent sequence and

$$\sum_{k=0}^{\infty} \sum_{t=\tilde{r}(k\sigma)}^{\tilde{r}((k+1)\sigma)-1} \rho(t) [\mathbf{y}(k) - \mathbf{x}^*]^\top [\phi(\mathbf{y}(k)) - \phi(\mathbf{x}^*)] < \infty.$$

With similar statements as those in Koshal et al. (2016, Proposition 2), the sequence $\{\mathbf{y}(k)\}$ converges to \mathbf{x}^* . Together with the fact that $\lim_{t \rightarrow \infty} \rho(t) = 0$, it implies that $\{\mathbf{x}(k)\}$ converges to \mathbf{x}^* . \square

B Proof of Lemma 2

Note that $\tilde{A}(k)$ is a column-stochastic matrix, i.e., $\mathbf{1}^\top \tilde{A}(k) = \mathbf{1}^\top$ for all $k \geq 0$. Lemma 4 below shows that the summation of $\mathbf{v}_i(k)$ is exactly equal to that of $\mathbf{x}_i(k)$.

Lemma 4. *Let $\tilde{A}(k)$ be such that $\sum_{i=1}^n [\tilde{A}(k)]_{ij} = 1$ for every j and $k \geq 0$. Then,*

$$\sum_{i=1}^{\tilde{n}} \mathbf{v}_i(k) = \sum_{i=1}^n \mathbf{x}_i(k), \quad (15)$$

where $\{\mathbf{v}_i(k)\}$ and $\{\mathbf{x}_i(k)\}$ are two sequences generated by (3).

Proof. From the initialization step of Algorithm 1, (15) holds for $k = 0$. Taking the summation over i on the both sides of the last step in (3), we have that

$$\begin{aligned} \sum_{i=1}^{\tilde{n}} \mathbf{v}_i(k+1) &= \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{\tilde{n}} [\tilde{A}(k)]_{ij} \mathbf{v}_j(k) + \sum_{i=1}^n [\mathbf{x}_i(k+1) - \mathbf{x}_i(k)] \\ &= \sum_{j=1}^{\tilde{n}} \mathbf{v}_j(k) + \sum_{i=1}^n \mathbf{x}_i(k+1) - \sum_{i=1}^n \mathbf{x}_i(k). \end{aligned}$$

The result (15) is then arrived at by induction. \square

Defining $\Phi(k, t) = \tilde{A}(k) \tilde{A}(k-1) \cdots \tilde{A}(k-t)$, Lemma 5 shows that it converges to a rank one matrix with identical columns, and that $\tilde{y}_i(k)$ is positive for all $i \in \mathcal{N}$ and $k \geq 0$.

Lemma 5 (Zhang & You 2020). *Under Assumption 4, the following statements are in force.*

(a) *There exists a nonnegative vector $\phi(k)$ satisfying $\mathbf{1}^\top \phi(k) = 1$ and*

$$|[\Phi(k, t)]_{ij} - \phi_i(k)| \leq B' \lambda^t \quad (16)$$

for all $k \geq t \geq 0$, where

$$B' = 4(1 + n^{nb}), \quad \lambda = \left(1 - \frac{1}{n^{nb}}\right)^{\frac{1}{nb}} \quad (17)$$

and b is as defined in Lemma 1(c).

(b) $\tilde{y}_i(k) = \sum_{j=1}^n [\Phi(k, k)]_{ij} \geq n^{-nb}$, $\forall i \in \mathcal{N}, k \geq 0$.

Proof of Lemma 2.

(a) By Lemma 4, we have that $\bar{\mathbf{x}}(k) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \tilde{\mathbf{v}}_i(k)$. Then, the proof is similar to that of Nedić & Olshevsky (2014, Lemma 1).

(b) It follows from (3) and the projection theorem that

$$\|\mathbf{x}_i(t+1) - \mathbf{x}_i(t)\| \leq \alpha_i(t) \|F_i(\mathbf{x}_i(t), \mathbf{z}_i(t+1))\|.$$

It follows from Assumption 3 that

$$\|F_i(\mathbf{x}_i(t), \mathbf{z}_i(t+1))\| \leq \|F_i(\mathbf{x}_i(t), \bar{\mathbf{x}}(t))\| + L\|\mathbf{z}_i(t+1) - \bar{\mathbf{x}}(t)\|.$$

Thus, $\|F_i(\mathbf{x}_i(t), \mathbf{z}_i(t+1))\|$ is bounded under Assumption 1 and (a). Besides, it follows from Lemma 1(c) and Assumption 5(c) that $\lim_{t \rightarrow \infty} \alpha_i(t) = 0$. Thus, we have that $\|\mathbf{x}_i(t+1) - \mathbf{x}_i(t)\| \rightarrow 0$ for all $i \in \mathcal{N}$. Then, the rest of the proof and the proof of (c) are similar to that of Nedić & Olshevsky (2014, Lemma 1). \square

References

- Alpcan, T. & Başar, T. (2002), A game-theoretic framework for congestion control in general topology networks, in ‘Proceedings of the 41st IEEE Conference on Decision and Control, 2002.’, Vol. 2, IEEE, pp. 1218–1224.
- Assran, M. S. & Rabbat, M. G. (2021), ‘Asynchronous gradient push’, *IEEE Transactions on Automatic Control* **66**(1), 168–183.
- Barrera, J. & Garcia, A. (2014), ‘Dynamic incentives for congestion control’, *IEEE Transactions on Automatic Control* **60**(2), 299–310.
- Başar, T. & Olsder, G. J. (1998), *Dynamic Noncooperative Game Theory*, SIAM.
- Belgioioso, G., Nedich, A. & Grammatico, S. (2020), ‘Distributed generalized Nash equilibrium seeking in aggregative games on time-varying networks’, *IEEE Transactions on Automatic Control* pp. 1–1.
- Bertsekas, D. (2016), *Nonlinear Programming*, Athena Scientific.
- Bertsekas, D. P. (1983), ‘Distributed asynchronous computation of fixed points’, *Mathematical Programming* **27**(1), 107–120.
- Bertsekas, D. P. (2015), *Convex Optimization Algorithms*, Athena Scientific Belmont.
- Bottou, L. (2010), Large-scale machine learning with stochastic gradient descent, in ‘Proceedings of COMPSTAT’2010’, Springer, pp. 177–186.
- Cenedese, C., Belgioioso, G., Grammatico, S. & Cao, M. (2020), ‘An asynchronous distributed and scalable generalized Nash equilibrium seeking algorithm for strongly monotone games’, *arXiv preprint arXiv:2005.03507*.
- Chazan, D. & Miranker, W. (1969), ‘Chaotic relaxation’, *Linear Algebra and Its Applications* **2**(2), 199–222.
- Facchinei, F. & Pang, J.-S. (2007), *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer Science & Business Media.

- Frihauf, P., Krstic, M. & Başar, T. (2012), ‘Nash equilibrium seeking in noncooperative games’, *IEEE Transactions on Automatic Control* **57**(5), 1192–1207.
- Gharesifard, B., Başar, T. & Domínguez-García, A. D. (2016), ‘Price-based coordinated aggregation of networked distributed energy resources’, *IEEE Transactions on Automatic Control* **61**(10), 2936–2946.
- Hannah, R., Feng, F. & Yin, W. (2018), A2BCD: Asynchronous acceleration with optimal complexity, *in* ‘International Conference on Learning Representations’.
- Herz, A. V. M. & Marcus, C. M. (1993), ‘Distributed dynamics in neural networks’, *Phys. Rev. E* **47**, 2155–2161.
- Koshal, J., Nedić, A. & Shanbhag, U. V. (2016), ‘Distributed algorithms for aggregative games on graphs’, *Operations Research* **64**(3), 680–704.
- Lei, J., Shanbhag, U. V., Pang, J.-S. & Sen, S. (2020), ‘On synchronous, asynchronous, and randomized best-response schemes for stochastic Nash games’, *Mathematics of Operations Research* **45**(1), 157–190.
- Li, N., Chen, L. & Dahleh, M. A. (2015), ‘Demand response using linear supply function bidding’, *IEEE Transactions on Smart Grid* **6**(4), 1827–1838.
- Li, S. & Başar, T. (1987a), ‘Asymptotic agreement and convergence of asynchronous stochastic algorithms’, *IEEE Transactions on Automatic Control* **32**(7), 612–618.
- Li, S. & Başar, T. (1987b), ‘Distributed algorithms for the computation of noncooperative equilibria’, *Automatica* **23**(4), 523 – 533.
- Liu, P., Fu, Z., Cao, J., Wei, Y., Guo, J. & Huang, W. (2020), ‘A decentralized strategy for generalized Nash equilibrium with linear coupling constraints’, *Mathematics and Computers in Simulation* **171**, 221–232.
- Ma, Z., Callaway, D. S. & Hiskens, I. A. (2013), ‘Decentralized charging control of large populations of plug-in electric vehicles’, *IEEE Transactions on Control Systems Technology* **21**(1), 67–78.
- Nedić, A. & Olshevsky, A. (2014), ‘Distributed optimization over time-varying directed graphs’, *IEEE Transactions on Automatic Control* **60**(3), 601–615.
- Nedić, A. & Ozdaglar, A. (2010), ‘Convergence rate for consensus with delays’, *Journal of Global Optimization* **47**(3), 437–456.
- Nesterov, Y. (2012), ‘Efficiency of coordinate descent methods on huge-scale optimization problems’, *SIAM Journal on Optimization* **22**(2), 341–362.
- Richtárik, P. & Takáč, M. (2014), ‘Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function’, *Mathematical Programming* **144**(1-2), 1–38.
- Salehisadaghiani, F. & Pavel, L. (2016), ‘Distributed Nash equilibrium seeking: A gossip-based algorithm’, *Automatica* **72**, 209–216.
- Salehisadaghiani, F., Shi, W. & Pavel, L. (2019), ‘Distributed Nash equilibrium seeking under partial-decision information via the alternating direction method of multipliers’, *Automatica* **103**, 27–35.
- Sha, X., Zhang, J., Zhang, K., You, K. & Başar, T. (2020), ‘Asynchronous policy evaluation in distributed reinforcement learning over networks’, *arXiv preprint arXiv:2003.00433*.
- Tsitsiklis, J., Bertsekas, D. & Athans, M. (1986), ‘Distributed asynchronous deterministic and stochastic gradient optimization algorithms’, *IEEE Transactions on Automatic Control* **31**(9), 803–812.

- Yi, P. & Pavel, L. (2020), ‘Asynchronous distributed algorithms for seeking generalized Nash equilibria under full and partial-decision information’, *IEEE Transactions on Cybernetics* **50**(6), 2514–2526.
- Yousefian, F., Nedić, A. & Shanbhag, U. V. (2013), A distributed adaptive steplength stochastic approximation method for monotone stochastic Nash games, *in* ‘2013 American Control Conference’, IEEE, pp. 4765–4770.
- Zhang, J. & You, K. (2019), ‘Asynchronous decentralized optimization in directed networks’, *arXiv preprint arXiv:1901.08215*.
- Zhang, J. & You, K. (2020), ‘AsySPA: An exact asynchronous algorithm for convex optimization over digraphs’, *IEEE Transactions on Automatic Control* **65**(6), 2494–2509.