

## Article

# Distributed GNE-Seeking under Partial Information Based on Preconditioned Proximal-Point Algorithms

Zhongzheng Wang, Huaqing Li, Menggang Chen, Jialong Tang, Jingran Cheng and Yawei Shi \*

Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

\* Correspondence: tomson@swu.edu.cn

**Abstract:** This paper proposes a distributed algorithm for games with shared coupling constraints based on the variational approach and the proximal-point algorithm. The paper demonstrates the effectiveness of the proximal-point algorithm in distributed computing of generalized Nash equilibrium (GNE) problems using local data and communication with neighbors in any networked game. The algorithm achieves the goal of reflecting local decisions in the Nash–Cournot game under partial-decision information while maintaining the distributed nature and convergence of the algorithm.

**Keywords:** generalized Nash equilibrium; distributed algorithms; variational approach; proximal point algorithm; partial information; networked game

## 1. Introduction

The problem of finding a generalized Nash equilibrium (GNE) for networked systems has gained significant interest recently owing to its applicability to multi-agent decision-making scenarios, such as demand-side management in intelligent grids [1], demand response in competitive markets [2], and electric vehicle charging [3]. In such systems, the agents aim to minimize cost functions under joint feasibility constraints with non-cooperative settings. They cannot reduce their local costs by unilaterally changing their decisions but by relying on the decisions of the other agents. The GNE is a self-executing outcome, and once computed, all agents execute it to achieve their own minimum cost. In these systems, each user has a local cost function to minimize, which is a function of their own and other users' decisions. Moreover, each user's feasible decision set is coupled with other users' decisions due to the limited network resources. A natural solution concept for such systems is the GNE, which captures the non-cooperative behavior of multiple interacting agents. A GNE is a vector of decisions such that no user can reduce their local cost by unilaterally deviating from it, given the decisions of other users. Hence, a GNE is a self-enforcing outcome that all agents will implement once computed, as it achieves their individual minimum costs.

To compute GNE introspectively, most algorithms for finding Nash equilibrium require a central information center to collect and store all game information such as cost functions, coupling constraints, and feasible local sets in a traditional computing environment. However, this approach is not suitable for large-scale network games because it is difficult to observe the true decisions of all other players in this type of game. Additionally, having a central coordinator node may be impractical due to technical, geographical, or game-related reasons. In this case, each player needs to compute their local decision corresponding to GNE in a distributed way by utilizing their local objective function, local feasible set, and possibly local data related to the coupling constraints and communicating with their neighbors. Therefore, our goal is to develop a distributed algorithm that estimates the strategies of all other agents only by communicating with adjacent agents.



**Citation:** Wang, Z.; Li, H.; Chen, M.; Tang, J.; Cheng, J.; Shi, Y. Distributed GNE-Seeking under Partial Information Based on Preconditioned Proximal-Point Algorithms. *Appl. Sci.* **2023**, *13*, 6405. <https://doi.org/10.3390/app13116405>

Academic Editor: Krzysztof Koszela

Received: 16 April 2023

Revised: 15 May 2023

Accepted: 22 May 2023

Published: 24 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

This algorithm can achieve the goal of reflecting local decisions in the absence of global information and eventually reconstructing actual value.

The proximal-point algorithm has been widely used in distributed computing of GNE problems. In our study, we demonstrate that it can achieve the goal of reflecting local decisions in the absence of global information while maintaining the distributed nature and convergence of the algorithm. That is to say, in the absence of a central coordinator, the generalized Nash equilibrium point is calculated by each participant using local information and communication with neighbors. This approach enables each player to compute a local decision corresponding to GNE by utilizing its local objective function, local feasible set, and possibly local data related to the coupling constraints and communicating with its neighbors. By continuously updating the proximal point, the proximal-point algorithm can achieve the convergence of restricted monotone and Lipschitz continuous pseudo-gradient games under appropriate fixed step size conditions. Our study proposes a new distributed GNE algorithm for games with shared biomimetic coupling constraints, which is based on the variational GNE approach and the proximal-point algorithm. Our results demonstrate the effectiveness of the proposed algorithm in achieving GNE in a distributed manner in the presence of shared constraints.

### 1.1. Literature Review

The research of GNE was initiated by the seminal works of [4,5]. In recent years, the computation of distributed GNE in monotone games has received considerable interest. However, most existing studies rely on the assumption that every agent in the system has access to the decision information of other agents, i.e., by employing a central node coordinator to orchestrate the information processing. The initial approach was based on the variational inequality method [6]. The variational inequality framework for generalized Nash equilibrium problems established in [6] has been extended to more general settings, such as time-dependent or stochastic problems in [7]. Ref. [8] developed a Tikhonov-regularized primal-dual algorithm, and [9] devised a primal-dual gradient method, which can reduce the compensation and assumed that it can acquire the decisions of all adversaries that influence each cost. In ref. [10], a payoff-based GNE algorithm was introduced, which has a diminishing step size and converges to a class of convex games. Recently, an operator-splitting method has emerged as a very powerful design technique, which ensures global convergence and features a fixed step size and succinct convergence analysis. However, most GNE application scenarios are aggregative games, such as [3,11–14]. In refs. [12,13], the algorithms are semi-decentralized, requiring a central node to disseminate the common multipliers and aggregate variables, thus resulting in a star topology. In ref. [14], this is relaxed by a projection gradient algorithm based on two-layer consensus, both of which are only suitable for aggregative games.

In networked non-cooperative games, a distributed forward-backward operator-splitting algorithm for seeking the GNE has been proposed by Yi and Pavel [15], based on the convex analysis framework of Bauschke and Combettes [16]. Under partial information, the algorithm requires only local objective function, feasible set, and block of affine constraints for each agent. It also introduces a local estimate of all agents' decisions. They further extended their algorithm to an asynchronous setting by using auxiliary variables associated with Graphology [17]. In this setting, each agent uses private data and delay information from their neighbors' latency information to iterate asynchronously. They proved the convergence and effectiveness of their algorithm under mild assumptions. Ref. [18] investigated a generalized Nash equilibrium problem in which players are modeled as nodes of a network and each player's utility function depends on its own and its neighbors' actions, and derived a variational decomposition of the game under a quadratic reference model with shared constraints, which is illustrated with numerical examples. Bianchi et al. [19] has developed a novel algorithm that differs from the existing projective pseudo gradient dynamics in that it is fully distributed, one-layer, and uses a proximal best response with consensus terms. It can overcome the limitations of double-layer iterations or

conservative step sizes of the gradient-based methods. It can also extend the applicability of the proximal point method by analyzing the restricted monotonicity property. Pavel [20] applied a preconditioned proximal point algorithm (PPPA) that decomposes the merit-seeking task into Nash equilibrium (NE) computation of regularised subgame sequences based on local information. The PPPA also performs distributed updates of multipliers and auxiliary variables, which require multiple communications among agents to solve the subgame. For games with general coupling costs and affine coupling constraints, [15,21] adopt operator methods to perform distributed decentralized GNE optimization: forward-backward optimization algorithm for strongly monotone games [15] and preconditioned proximal optimization algorithm for monotone games [21]. The players exchange local multipliers over the network with arbitrary topology, but each user can access all the agent decisions that affect its cost, thus obtaining complete decision information.

### 1.2. Contributions

Compared with most existing distributed optimization algorithms, the main contributions of this paper are summarized as follows:

- The proposal of a GNE algorithm for games with shared biomimetic coupling constraints. The algorithm is based on the variational GNE approach and the proximal-point algorithm, and is improved by introducing two choice matrices to enhance its accuracy, as in [22,23], where we design a novel preconditioning matrix to distribute the computation and obtain a single-layer iteration. Each player has an auxiliary variable to estimate the decisions of other agents. The algorithm is distributed, where each player only utilizes its local objective function, local feasible set, and local data related to the coupling constraints, and there is no centralized coordinator to update and propagate dual variables.
- An original dual analysis of the Karush–Kuhn–Tucker (KKT) conditions of the variational inequality (VI) is conducted, which introduces a local copy of the multiplier and an auxiliary variable for each player. It is observed that the KKT conditions mandate consensus among all agents on the multiplier for shared constraints. By reformulating the original problem as finding the zero point of a monotone operator that includes the Laplacian matrix of the connected graph, the consistency of local multipliers is enhanced.

This paper presents global and distributed methods for finding GNE in games with shared affine coupling constraints under partial decision information. Section 2 introduces the game model and formulates the GNE problem. Section 3 proposes a global GNE-seeking method based on the proximal-point algorithm with global information. Section 4 develops a distributed GNE-finding method with partial information and proves its convergence and implementation feasibility. Section 5 illustrates the performance of our methods through numerical simulations. Finally, Section 6 concludes the paper and discusses some future work directions.

## 2. Game Formulation

We study a non-cooperative generalized game with a crowd of agents. Each agent  $i \in \mathcal{I}$  ( $\mathcal{I} = 1, \dots, N$ ) has a local decision set  $\Omega_i \subseteq \mathbb{R}^{n_i}$  and chooses its own decision  $x_i \in \Omega_i$ , where  $\mathbb{R}$  is the set of real numbers and  $n_i$  is the dimension of agent  $i$ 's decision-making. The global decision space is  $\Omega = \prod_{i=1}^N \Omega_i \subset \mathbb{R}^n$  where  $n = \sum_{i=1}^N n_i$  and the stacked vector of all agents' decisions is  $\mathbf{x} = \text{col}((x_i)_{i \in \mathcal{I}})$ . Let  $\mathbf{x}_{-i} = \text{col}((x_j)_{j \in \mathcal{I}, j \neq i})$  denote the decision profile of all other agents except for agent  $i$ , then  $\mathbf{x} = (x_i, \mathbf{x}_{-i})$ . The feasible set for any agent depends on the coupling constraints shared with other agents and their own decisions. Each agent aims to minimize its objective function over this feasible set as follows:

$$\min_{x_i} J_i(x_i, \mathbf{x}_{-i}) := f_i(x_i, \mathbf{x}_{-i}) + g_i(x_i). \quad (1)$$

The objective function (1) can describe how multiple participants can optimize their interests in a non-cooperative situation while being affected by sharing constraints. This model can be applied to a variety of practical scenarios, such as smart grids, competitive markets, and electric vehicle charging.

**Assumption 1.** The cost functions  $f_i$  and  $g_i$  of each agent  $i \in \mathcal{I}$  are convex. The common cost function  $f_i$  is continuously differentiable and the local idiosyncratic cost function  $g_i$  is lower semicontinuous.  $\Omega_i$  is bounded and closed for any  $i \in \mathcal{I}$ .

We denote  $A := [A_1, \dots, A_N]$  and  $b := \sum_{i=1}^N b_i$ , where  $A_i \in \mathbb{R}^{m \times n_i}$  and  $b_i \in \mathbb{R}^m$  are local parameters. By the affine function above, the set of overall feasible decisions via affine coupling constraints and cost functions is denoted as follows:

$$\mathcal{X} := \Omega \cap \left\{ x \in \mathbb{R}^{nN} \mid Ax - b \leq \mathbf{0}_m \right\} \subseteq \mathbb{R}^{nN}, \quad (2)$$

where  $\mathcal{X}$  is a closed and convex set of decisions that is nonempty. It is used to simulate some phenomena in nature, such as group behavior, competition, and cooperation. Such constraints can reflect the interactions and dependencies among participants, making the game more complex and interesting.

**Assumption 2.** Slater's constraint qualification holds for the collective set  $\mathcal{X}$ . Therefore, each agent  $i \in \mathcal{I}$  in the generalized game tries to solve this interdependent optimization problem as follows:

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

To obtain the primal-dual characterization of each agent  $i \in \mathcal{I}$ , we define a Lagrangian function with dual variable multiplier  $\lambda_i \in \mathbb{R}_+^m$  as follows:

$$\mathcal{L}_i(x_i, x_{-i}; \lambda_i) := f_i(x_i, x_{-i}) + g_i(x_i) + \lambda_i^T \left( \sum_{i=1}^N (A_i x_i - b_i) \right). \quad (4)$$

We call a decision  $x^* \in \mathcal{X}$  that satisfies (3) a GNE of the game. This implies that for any agent  $i \in \mathcal{I}$ ,

$$J_i(x_i^*, x_{-i}^*) \leq \inf \{ J_i(y, x_{-i}^*) \mid y \in \mathcal{X}_i(x_{-i}^*) \}, \quad (5)$$

where  $\mathcal{X}_i(x_{-i}) := \left\{ y_i \in \Omega_i \mid A_i y_i \leq b - \sum_{j \neq i}^N A_j x_j \right\}$ .

If  $x_i^* \in \Omega_i$  satisfies the KKT conditions below with  $\lambda_i^* \in \mathbb{R}_+^m$ , then it is an optimal solution to (3).

$$\forall i \in \mathcal{I} : \begin{cases} \nabla_{x_i} \mathcal{L}_i(x_i^*, x_{-i}^*) = \mathbf{0} \\ \langle \lambda_i^*, Ax^* - b \rangle = 0 \\ -(Ax^* - b) \geq \mathbf{0} \\ \lambda_i^* \geq 0. \end{cases} \quad (6)$$

We reformulate the KKT condition using the normal cone operator as follows:

$$\begin{cases} 0 \in \nabla_{x_i} f_i(x_i^*, x_{-i}^*) + \partial g_i(x_i^*) + A_i^T \lambda_i^* \\ 0 \in \sum_{i=1}^N A_i x_i^* - \sum_{i=1}^N b_i - \mathbf{N}_{\mathbb{R}_{\geq 0}^m}(\lambda_i^*). \end{cases} \quad (7)$$

We denote the pseudogradient of  $f_i(x_i, \mathbf{x}_{-i})$  as  $\mathbb{F}(\mathbf{x}) = \text{col}(\nabla_{x_i} f_i(x_i, \mathbf{x}_{-i}))_{i \in \mathcal{I}} \in \mathbb{R}^n$ . Denote the pseudogradient of  $g_i(x_i)$  as  $g(\mathbf{x}) = \sum_{i=1}^N g_i(x_i)$ .

Let  $\lambda = \text{col}(\lambda_i)_{i \in N}$  denote the vector of Lagrangian multipliers associated with each agent  $i \in N$ . A variational GNE is a decision  $\mathbf{x}^* \in \mathcal{X}$  that satisfies the variational inequality problem  $\text{VI}(\mathbb{F}, \mathcal{X})$  and has equal Lagrangian multipliers ( $\lambda_1 = \dots = \lambda_N$ ).

$$\begin{aligned} \text{find } \mathbf{x}^* \in \mathcal{X} &= \left\{ \mathbf{x} \in \mathbb{R}^n \mid \sum_{i=1}^N A_i x_i \leq \sum_{i=1}^N b_i \right\}, \\ \text{s.t. } &\langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle + g(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathcal{X}. \end{aligned} \quad (8)$$

Under Assumption 1,  $J_i$  is a continuous and convex function, and  $\Omega_i$  is a bounded local decision set. These ensure that  $\text{VI}(\mathbb{F}, \mathcal{X})$  has a solution. Let  $\mathbf{x}^*$  be such a solution. Then it satisfies (7) with equal Lagrange multipliers  $\lambda^*$ . Therefore,  $\mathbf{x}^*$  is a variational GNE for (3).

### 3. Iterative Algorithms with Global Information

We propose a distributed algorithm based on preconditioned proximal-point algorithms that use full-decision information. This means that every agent can access the decisions of all other agents that affect its local objective function directly.

#### 3.1. Communication Graph

Let  $|\mathcal{N}| = N$  and  $|\mathcal{E}| = M$ . We define  $\mathcal{E}_i$  as the set of edges adjacent to  $i$ , which consists of  $\mathcal{E}_{i,\text{in}}$  (the incoming edges) and  $\mathcal{E}_{i,\text{out}}$  (the outgoing edges). Let  $W = [w_{i,j}]_{i,j \in \mathcal{I}} \in \mathbb{R}^{N \times N}$  be the symmetric weight matrix of  $\mathcal{G}$ , where  $w_{i,j} > 0$  if  $(i, j) \in \mathcal{E}$  and  $w_{i,j} = 0$  otherwise. We also set  $w_{i,i} = 0$  for all agents. The Laplacian matrix of  $\mathcal{G}$  is denoted by  $L := D - W$ , where  $D := \text{diag}((d_i)_{i \in \mathcal{I}})$  is the degree matrix and  $d_i := \sum_{j=1}^N w_{i,j}$  for all  $i \in \mathcal{I}$ .

**Assumption 3.**  $\mathcal{G}$  is a connected and undirected graph.

#### 3.2. Algorithm Development

**Assumption 4.**  $\mathbb{F}(\mathbf{x})$  is a  $\mu$ -strongly monotone function, meaning that  $\langle \mathbb{F}(\mathbf{x}) - \mathbb{F}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \mu \|\mathbf{x} - \mathbf{y}\|_2^2, \forall \mathbf{x}, \mathbf{y} \in \Omega$ , and a  $\theta_0$ -Lipschitz continuous function, implying that  $\|\mathbb{F}(\mathbf{x}) - \mathbb{F}(\mathbf{y})\|_2 \leq \theta_0 \|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \Omega$ .

The strong monotonicity of  $\mathbb{F}$  implies that a v-GNE exists and is unique. We can write the variational problem for the original game problem as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} & \langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} \rangle + g(\mathbf{x}), \\ \text{s.t. } & \sum_{i=1}^N A_i x_i \leq \sum_{i=1}^N b_i. \end{aligned} \quad (9)$$

The objective function is defined by the following Lagrangian function:

$$\tilde{\mathcal{L}}(\mathbf{x}, \lambda_g) := \langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} \rangle + g(\mathbf{x}) + \lambda_g^T \left( \sum_{i=1}^N (A_i x_i - b_i) \right). \quad (10)$$

Let  $\lambda_g \in \mathbb{R}^m$  be a global Lagrangian multiplier. To design distributed optimization algorithms, we impose the consistency constraints  $\lambda_i = \lambda_j, \forall (i, j) \in \mathcal{E}$ , i.e.,  $(L \otimes I_m) \lambda = 0$ , where  $I_m$  is an identity matrix of order  $m$ . Define  $\mathbb{G}(\mathbf{x}) = \text{col}(\partial g_i(x_i))_{i \in \mathcal{I}} \in \mathbb{R}^n$ ,  $\lambda = \text{col}(\lambda_1, \dots, \lambda_N) = \text{col}(\lambda_i) \in \mathbb{R}^m$ ,  $\mathbf{A} = [A_1, \dots, A_N] \in \mathbb{R}^{m \times n}$ , and  $\mathbf{b} = \text{col}(b_i) \in \mathbb{R}^m$ . Then,

$$\mathcal{L}_1(\mathbf{x}, \lambda) := \langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} \rangle + g(\mathbf{x}) + \lambda^T (\mathbf{A} \mathbf{x} - \mathbf{b}). \quad (11)$$

Its corresponding saddle point problem would be

$$\begin{aligned} \max_{\lambda} \min_x \quad & \langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} \rangle + g(\mathbf{x}) + \lambda^T (\mathbf{A}\mathbf{x} - \mathbf{b}), \\ \text{s.t.} \quad & (\mathbf{L} \otimes \mathbf{I}_m) \lambda = 0. \end{aligned} \quad (12)$$

Let  $\mathbf{L} = (\mathbf{L} \otimes \mathbf{I}_m)$ ,  $\mathbf{W} = (\mathbf{W} \otimes \mathbf{I}_m)$  and  $\mathbf{D} = (\mathbf{D} \otimes \mathbf{I}_m)$ , the Lagrangian function that defines this saddle point problem is  $\mathcal{L}_2(\mathbf{x}, \lambda, \mathbf{z}) = \langle \mathbb{F}(\mathbf{x}^*), \mathbf{x} \rangle + g(\mathbf{x}) + \lambda^T (\mathbf{A}\mathbf{x} - \mathbf{b}) + \mathbf{z}^T \mathbf{L}\lambda$ . The optimal conditions for this problem are obtained by sequentially finding the partial derivatives of the variables  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  as follows:

$$\begin{cases} 0 \in \mathbb{F}(\mathbf{x}) + \mathbb{G}(\mathbf{x}) + \mathbf{A}^T \lambda \\ 0 = \mathbf{L}\lambda \\ 0 \in \mathbf{b} + \mathbf{L}\lambda + \mathbf{N}_{\mathbb{R}_{\geq 0}^{Nm}}(\lambda) - \mathbf{A}\mathbf{x} - \mathbf{L}\mathbf{z}. \end{cases} \quad (13)$$

It follows from Lagrange's dual theorem that the variational problem has an optimal solution  $\mathbf{x}^* = (x_i^*, \mathbf{x}_{-i}^*)$  only when  $\lambda_g^* \in \mathbb{R}^m$  satisfies the KKT conditions (13).

**Lemma 1.** *If Assumptions 1–3 hold, then (13) implies (7).*

Let  $\mathbf{v} = \text{col}(\mathbf{x}, \mathbf{z}, \lambda)$  and consider the following operator:

$$\mathcal{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbb{F}(\mathbf{x}) \\ \mathbf{0} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} \mathbb{G}(\mathbf{x}) \\ \mathbf{0} \\ \mathbf{N}_{\mathbb{R}_{\geq 0}^{Nm}}(\lambda) \end{bmatrix} + \begin{bmatrix} \mathbf{A}^T \lambda \\ \mathbf{L}\lambda \\ -\mathbf{A}\mathbf{x} - \mathbf{L}\mathbf{z} \end{bmatrix}. \quad (14)$$

We regard the iterative algorithm as a special case of proximal-point algorithms (PPA) [16] for finding a zero of  $\mathcal{A}$ . The general form of PPA can be written as

$$\mathbf{v}^{k+1} \in \mathbf{J}_{\mathcal{A}}(\mathbf{v}^k), \quad (15)$$

where  $\mathbf{J}_{\mathcal{A}}(\mathbf{v}^k) \neq \emptyset$ .

We apply the interaction rule (9) to the operator  $\Phi^{-1}\mathcal{A}$ , where  $\Phi$  is defined as the following:

$$\Phi = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{A}^T \\ \mathbf{0} & \mathbf{0} & -\mathbf{L} \\ -\mathbf{A} & -\mathbf{L} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \alpha^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \gamma^{-1} \end{bmatrix}. \quad (16)$$

We choose the step sizes  $\alpha = \text{diag}(\alpha_i \mathbf{I}_{n_i})$ ,  $\tau = \text{diag}(\tau_i \mathbf{I}_m)$  and  $\gamma = \text{diag}(\gamma_i \mathbf{I}_m)$  such that  $\Phi > 0$ . The next lemma provides sufficient conditions for  $\Phi > 0$  based on Gershgorin's circle theorem.

**Lemma 2.** *For any agent  $i \in \mathcal{I}$  and any  $\delta > 0$ , the preconditioning matrix  $\Phi$  in (16) is positive definite if*

$$\begin{aligned} 0 < \alpha_i &\leq \left( \max_{j \in \{1, \dots, n_i\}} \sum_{k=1}^m |[\mathbf{A}_i^T]_{jk}| + \delta \right)^{-1}, \\ 0 < \tau_i &\leq (2d_i + \delta)^{-1}, \\ 0 < \gamma_i &\leq \left( \max_{j \in \{1, \dots, m\}} \sum_{k=1}^{n_i} |[\mathbf{A}_i]_{jk}| + 2d_i + \delta \right)^{-1}. \end{aligned} \quad (17)$$

Assuming that  $\Phi > 0$  for all chosen step sizes, we can state the following results.

**Lemma 3.** *The iterative algorithm is equivalent to the following:*

$$(\forall k \in \mathbb{N}) \quad \mathbf{v}^{k+1} \in \mathbf{J}_{\Phi^{-1}\mathcal{A}}(\mathbf{v}^k), \quad (18)$$



where  $\mathcal{A}$  is given by (14),  $\Phi$  is given by (16).

**Proof of Lemma 3.** We use the definition of the inverse operator and obtain that

$$\begin{aligned} v^{k+1} &\in \left(\text{Id} + \Phi^{-1}\mathcal{A}\right)^{-1}(v^k) \\ \Leftrightarrow \mathbf{0} &\in \Phi^{-1}\mathcal{A}(v^{k+1}) - v^k + v^{k+1} \\ \Leftrightarrow \mathbf{0} &\in \Phi(v^{k+1} - v^k) + \mathcal{A}(v^{k+1}). \end{aligned} \quad (19)$$

Substituting  $\mathcal{A}$  and  $\Phi$  into (19) and simplifying them, we obtain that

$$\begin{aligned} \mathbf{x}_{k+1} + \alpha \mathbb{F}(\mathbf{x}_{k+1}) + \alpha G(\mathbf{x}_{k+1}) &= -\alpha \mathbf{A}^T \boldsymbol{\lambda}_k + \mathbf{x}_k, \\ \mathbf{z}_{k+1} &= -\tau \mathbf{L} \boldsymbol{\lambda}_k + \mathbf{z}_k, \\ \boldsymbol{\lambda}_{k+1} + \mathbf{N}_{\mathbb{R}_{\geq 0}^{Nm}}(\boldsymbol{\lambda}_{k+1}) &= -\gamma \mathbf{A} \mathbf{x}_k - \gamma \mathbf{L} \mathbf{z}_k + \boldsymbol{\lambda}_k + 2\gamma \mathbf{A} \mathbf{x}_{k+1} + 2\gamma \mathbf{L} \mathbf{z}_{k+1} - \gamma \mathbf{b}. \end{aligned} \quad (20)$$

Using the definition of the proximal operator [16], we can write (20) as

$$\begin{aligned} \mathbf{x}_{k+1} &= \arg \min_{\mathbf{x}} \left( \frac{1}{2} \|\mathbf{x} - \mathbf{x}_k\|^2 + \langle \mathbf{x}, \alpha \mathbf{A}^T \boldsymbol{\lambda}_k \rangle + \alpha f(\mathbf{x}) + \alpha g(\mathbf{x}) \right), \\ \mathbf{z}_{k+1} &= -\tau \mathbf{L} \boldsymbol{\lambda}_k + \mathbf{z}_k, \\ \boldsymbol{\lambda}_{k+1} &= \text{proj}_{\mathbb{R}_{\geq 0}^{Nm}}(\boldsymbol{\lambda}_k - \gamma \mathbf{A} \mathbf{x}_k - \gamma \mathbf{L} \mathbf{z}_k + 2\gamma \mathbf{A} \mathbf{x}_{k+1} + 2\gamma \mathbf{L} \mathbf{z}_{k+1} - \gamma \mathbf{b}). \end{aligned} \quad (21)$$

□

#### 4. Distributed Algorithm with Partial Information

In a distributed setting, it is challenging to access the decisions of all other agents requiring a central coordinator to collect and deliver information from all participants, as assumed in the previous section. The global information assumptions in Section 3 are relaxed in this section, allowing each participant to iteratively update their decisions, multipliers, and auxiliary variables using only their own local information and the estimates exchanged with their neighbors. Therefore, this section proposes a distributed GNE-seeking algorithm that uses partial information based on the preconditioned proximal-point algorithm.

##### 4.1. Algorithm Development

This section presents an algorithm for GNE-seeking in game (3) in a fully distributed manner.

Each agent  $i$  has a cost function  $J_i$  and a feasible set  $\Omega_i$  but does not know the full state  $\mathbf{x}_{-i}$  of other agents. Agent  $i$  can only exchange information with its neighboring agents over a network  $\mathcal{G}(\mathcal{I}, \mathcal{E})$ . The edge  $(i, j)$  belongs to  $\mathcal{E}$  if agents  $i$  and  $j$  can exchange information mutually.  $\mathbf{x}^i := \text{col}\left(\left(\mathbf{x}_j^i\right)_{j \in \mathcal{I}}\right) \in \mathbb{R}^n$  is defined as agent  $i$ 's estimate of agent  $j$ 's decision. Then  $\mathbf{x}_i^i := \mathbf{x}_i$  and  $\mathbf{x}_{-i}^j := \text{col}\left(\left(\mathbf{x}_\ell^j\right)_{\ell \in \mathcal{I} \setminus \{i\}}\right)$  denotes  $j$ 's estimate of all other agents except  $i$ . If  $\mathbf{x}^i = \mathbf{x}^j$ , we can replace the cost function of agent  $i$  with  $J_i(\mathbf{x}_i^i, \mathbf{x}_{-i}^i)$ . Then, we equivalently transform the game (3) as the following:

$$\begin{aligned} \min_{\mathbf{x}^i} \quad & J_i(\mathbf{x}_i^i, \mathbf{x}_{-i}^i) = f_i(\mathbf{x}_i^i, \mathbf{x}_{-i}^i) + g_i(\mathbf{x}_i^i), \\ \text{s.t.} \quad & \mathbf{x}_i^i \in \tilde{\mathcal{X}}_i, \\ & \mathbf{x}^i = \mathbf{x}^j, \forall j \in \mathcal{I}. \end{aligned} \quad (22)$$

It is worth noting that problems (3) and (22) are equivalent under a certain condition. We will explain this point below.

Let  $\tilde{\mathcal{X}}_i = \left\{ x \in \Omega_i \mid A_i x_i^i + \sum_{j \neq i} A_j x_j^i \leq \sum_{i=1}^m b_i \right\}$  be the set of feasible solutions for agent  $i$  under a consistency constraint. Then, agent  $i$ 's cost function  $J_i$  only depends on its own local information  $x^i$ .

We introduce matrices  $\mathcal{R}_i$  and  $\mathcal{S}_i$  to develop a distributed algorithm that uses partial decision information for game (22).

$$\mathcal{R}_i := \begin{bmatrix} \mathbf{0}_{n_i \times n_{<i}} & I_{n_i} & \mathbf{0}_{n_i \times n_{>i}} \end{bmatrix}, \quad (23)$$

$$\mathcal{S}_i := \begin{bmatrix} I_{n_{<i}} & \mathbf{0}_{n_{<i} \times n_i} & \mathbf{0}_{n_{<i} \times n_{>i}} \\ \mathbf{0}_{n_{>i} \times n_{<i}} & \mathbf{0}_{n_{>i} \times n_i} & I_{n_{>i}} \end{bmatrix}. \quad (24)$$

where  $n_{<i} := \sum_{j=1}^{i-1} n_j$ ,  $n_{>i} := \sum_{j=i+1}^N n_j$ . Then,  $\mathcal{R}_i x_i = x_i^i = x_i$  and  $\mathcal{S}_i x_i = x_{-i}^i$ . Let  $\mathcal{R} := \text{diag}((\mathcal{R}_i)_{i \in \mathcal{I}})$ ,  $\mathcal{S} := \text{diag}((\mathcal{S}_i)_{i \in \mathcal{I}})$ . Hence  $\mathcal{R}x = x$  and  $\mathcal{R}x = \text{col}((x_{-i}^i)_{i \in \mathcal{I}}) \in \mathbb{R}^{(N-1)n}$ . Moreover,  $x = \mathcal{R}^\top x + \mathcal{S}^\top \mathcal{S}x$ .

Similar to (13), we design sufficient conditions of (22) as follows:

$$\begin{cases} 0 \in \mathcal{R}^\top \hat{\mathbb{F}}(\hat{x}^*) + \mathcal{R}^\top \mathbb{G}(\mathcal{R}^* \hat{x}^*) + \mathcal{R}^\top A^\top \lambda^* + c \hat{\mathbb{L}} \hat{x}^* \\ 0 = \mathbf{L} \lambda^* \\ 0 \in A \mathcal{R} \hat{x}^* - b + \mathbf{L} z^* - N_{\mathbb{R}_{\geq 0}^{Nm}}(\lambda^*), \end{cases} \quad (25)$$

where  $\hat{\mathbb{L}} = (L \otimes I_n) \in \mathbb{R}^{Nn \times Nn}$ ,  $\mathcal{R} = \text{diag}\{\mathcal{R}_i\}$ ,  $\mathbb{F}(x) = \text{col}(\nabla_{x_i^i} f_i(x_i^i, x_{-i}^i))_{i \in \mathcal{I}} \in \mathbb{R}^n$ ,  $\mathbb{G}(x) = \text{col}(\nabla g_i(x_i^i))_{i \in \mathcal{I}} \in \mathbb{R}^n$ . Moreover,  $c$  is a parameter about the dual variable associated with the constraint  $0 = -\hat{\mathbb{L}}x^*$ .

Define the following operator:

$$\bar{\mathbb{A}} : \begin{bmatrix} \hat{x} \\ z \\ \lambda \end{bmatrix} \mapsto \begin{bmatrix} \mathcal{R}^\top \hat{\mathbb{F}}(\hat{x}) + \mathcal{R}^\top \mathbb{G}(\mathcal{R} \hat{x}) + \mathcal{R}^\top A^\top \lambda + c \hat{\mathbb{L}} \hat{x} \\ \mathbf{L} \lambda \\ -A \mathcal{R} \hat{x} + b - \mathbf{L} z + N_{\mathbb{R}_{\geq 0}^{Nm}}(\lambda) \end{bmatrix}, \quad (26)$$

$$\Phi = \begin{bmatrix} c \hat{\mathbb{W}} & \mathbf{0} & -\mathcal{R}^\top A^\top \\ \mathbf{0} & \mathbf{0} & -\mathbf{L} \\ -A \mathcal{R} & -\mathbf{L} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \alpha^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \gamma^{-1} \end{bmatrix}. \quad (27)$$

The variables  $x, z$  and  $\lambda$  satisfy condition (25) if  $\mathbf{0} \in \bar{\mathbb{A}}v$  where  $v = [\hat{x}^\top, z^\top, \lambda^\top]^\top$ .

**Lemma 4.** Algorithm 1 is equivalent to the following:

$$(\forall k \in \mathbb{N}) \quad v^{k+1} \in J_{\Phi^{-1} \bar{\mathbb{A}}}(v^k). \quad (28)$$

Let  $\bar{\mathbb{A}}$  be defined as in (26) and  $\Phi$  as in (27). Then, Algorithm 1 generates a sequence  $(x^k, z^k, \lambda^k)$  for  $k \in \mathbb{N}$  and any initial condition  $v^0 = \text{col}(x^0, z^0, \lambda^0)$ .



**Proof of Lemma 4.** By applying the definition of inverse operations, we obtain that

$$\begin{aligned}
 & v^{k+1} \in \left( \text{Id} + \Phi^{-1} \overline{\mathbb{A}} \right)^{-1} \left( v^k \right) \\
 \Leftrightarrow & 0 \in \Phi^{-1} \overline{\mathbb{A}} \left( v^{k+1} \right) - v^k + v^{k+1} \\
 \Leftrightarrow & 0 \in \Phi \left( v^{k+1} - v^k \right) + \overline{\mathbb{A}} \left( v^{k+1} \right) \\
 \Leftrightarrow & \begin{cases} 0 \in \alpha^{-1} (\hat{x}_{k+1} - \hat{x}_k) + c \hat{\mathbf{W}} \hat{x}_{k+1} - c \hat{\mathbf{W}} \hat{x}_k + c \hat{\mathbf{L}} \hat{x}_{k+1} - \mathcal{R}^\top A^\top \lambda_{k+1} + \mathcal{R}^\top A^\top \lambda_k \\ \quad + \mathcal{R}^\top \hat{\mathbb{F}} (\hat{x}_{k+1}) + \mathcal{R}^\top G (\mathcal{R} \hat{x}_{k+1}) - \mathcal{R}^\top A^\top \lambda_{k+1} \\ 0 \in \tau^{-1} (z^{k+1} - z^k) + \mathbf{L} \lambda_{k+1} + \mathbf{L} \lambda_k - \mathbf{L} \lambda_{k+1} \\ 0 \in \gamma^{-1} (\lambda_{k+1} - \lambda_k) + \mathbf{N}_{\mathbb{R}_{\geq 0}^{mN}} (\lambda^{k+1}) + \mathbf{b} - A \mathcal{R} (2 \hat{x}_{k+1} - \hat{x}_k) + \mathbf{L} (2 z_{k+1} - z_k). \end{cases} \quad (29)
 \end{aligned}$$

By  $\hat{\mathbf{L}} = \hat{\mathbf{D}} - \hat{\mathbf{W}}$ ,  $\mathcal{R} \alpha \mathcal{R}^\top = \tilde{\alpha}$ ,  $S \alpha \mathcal{R}^\top = \mathbf{0}$ ,  $\mathcal{R} \alpha \hat{\mathbf{L}} = \tilde{\alpha} \mathcal{R} \hat{\mathbf{L}}$ , and  $S \alpha \hat{\mathbf{L}} = \hat{\alpha} \hat{\mathbf{S}}$ , we have

$$\begin{aligned}
 & \begin{cases} 0 \in \mathcal{S} \left( (I + c \hat{\alpha} \hat{\mathbf{D}}) x_{k+1} - x_k - c \hat{\alpha} \hat{\mathbf{W}} x_k \right) \\ 0 \in \mathcal{R} \left( (I + c \tilde{\alpha} \hat{\mathbf{D}}) x_{k+1} - x_k - c \tilde{\alpha} \hat{\mathbf{W}} x_k \right) + \tilde{\alpha} \hat{\mathbb{F}} (\hat{x}_{k+1}) + \tilde{\alpha} G (\mathcal{R} x_{k+1}) + \tilde{\alpha} A^\top \lambda_k \end{cases} \\
 \Leftrightarrow & \begin{cases} x_{-i,k+1}^i = \frac{1}{1 + c \alpha_i d_i} \left( x_{-i,k}^i + \alpha_i c \sum_{j=1}^N w_{ij} x_{-i,k}^j \right) \\ \forall i \in \mathcal{I} : 0_{n_i} \in \partial_{x_i^{k+1}} \left( \frac{1}{2} \|x - x_{i,k}\|^2 + \alpha_i x^\top A_i^\top \lambda_{i,k} + \frac{c}{2} \alpha_i d_i \left\| x - \frac{1}{d_i} \sum_{j=1}^N w_{ij} x_{i,k+1}^j \right\|^2 \right. \\ \quad \left. + \alpha_i f_i (x, x_{-i,k+1}^i) + \alpha_i g_i (x) \right). \end{cases} \quad (30)
 \end{aligned}$$

Hence, by the property that the subdifferential of a convex function contains only zeros at its minima [16], we can reformulate (30) as follows.

$$\begin{aligned}
 \forall i \in \mathcal{I} : & \begin{cases} x_{-i,k+1}^i = \frac{1}{1 + \tau_i d_i} \left( x_{-i,k}^i + \tau_i \sum_{j=1}^N w_{ij} x_{-i,k}^j \right) \\ x_{i,k+1} = \underset{x}{\operatorname{argmin}} \left( \frac{1}{2} \|x - x_{i,k}\|^2 + \alpha_i x^\top A_i^\top \lambda_{i,k} + \alpha_i f_i (x, x_{-i,k+1}^i) \right. \\ \quad \left. + \alpha_i g_i (x) + \frac{c}{2} \alpha_i d_i \left\| x - \frac{1}{d_i} \sum_{j=1}^N w_{ij} x_{i,k+1}^j \right\|^2 \right), \end{cases} \quad (31) \\
 & z_{k+1} = -\tau \mathbf{L} \lambda_k + z_k, \\
 & \lambda_{k+1} = \operatorname{proj}_{\mathbb{R}_{\geq 0}^{mN}} \left( \lambda_k + \gamma A \mathcal{R} (2 \hat{x}_{k+1} - \hat{x}_k) + 2 \gamma \mathbf{L} z_{k+1} - \gamma \mathbf{L} z_k - \gamma \mathbf{b} \right).
 \end{aligned}$$

□

Write (31) as a distributed algorithm as Algorithm 1.

---

**Algorithm 1** Distributed Algorithm with Partial Information

---

**Initialize:** For all  $i \in \mathcal{I}$ , set  $x_{i,0} \in \Omega_i$ ,  $x_{-i,0}^i \in \mathbb{R}^{n-n_i}$ ,  $\lambda_{i,0} \in \mathbb{R}_+^m$ ,  $z_{i,0} \in \mathbb{R}^m$

**for**  $k = 1, 2, 3, \dots$  **do**

$$x_{-i,k+1}^i = \frac{1}{1 + \tau_i d_i} \left( x_{-i,k}^i + \tau_i \sum_{j=1}^N w_{ij} x_{-i,k}^j \right)$$

$$\begin{aligned}
 x_{i,k+1} = \underset{x}{\operatorname{argmin}} & \left( \frac{1}{2} \|x - x_{i,k}\|^2 + \alpha_i x^\top A_i^\top \lambda_{i,k} + \frac{c}{2} \alpha_i d_i \left\| x - \frac{1}{d_i} \sum_{j=1}^N w_{ij} x_{i,k+1}^j \right\|^2 \right. \\
 & \left. + \alpha_i f_i (x, x_{-i,k+1}^i) + \alpha_i g_i (x) \right)
 \end{aligned}$$

$$z_{i,k+1} = z_{i,k} - \tau_i \sum_{j=1}^N w_{ij} (\lambda_{i,k} - \lambda_{j,k})$$

$$\lambda_{i,k+1} = \operatorname{prox}_{\mathbb{R}_{\geq 0}^m} \left( \lambda_{i,k} + 2 \gamma_i A_i x_{i,k+1} - \gamma_i A_i x_{i,k} + 2 \gamma_i \sum_{j=1}^N w_{ij} (z_{i,k+1} - z_{j,k+1}) - \gamma_i b_i \right)$$

**end for**

**Return:** The sequence  $(x_{i,k})_{k=1}^\infty$  will eventually approximate the optimal solution.

---

#### 4.2. Convergence Analysis

In this section, we will prove that Algorithm 1 applying the game (22) converges to the variational GNE through a rigorous mathematical analysis. We show that any limit point of Algorithm 1 satisfies  $\mathbf{0} \in \Phi^{-1}\bar{\mathbb{A}}$  and has certain properties. Moreover, we demonstrate that every such zero is in the consensus subspace and solves  $\text{VI}(\mathbb{F}, \mathcal{X})$ .

**Assumption 5.**  $\mathbb{F}$  is a Lipschitz continuous mapping, which implies that there exists a positive constant  $\theta$  such that for any  $\mathbf{x}$  and  $\mathbf{y}$ , we have  $|\mathbb{F}(\mathbf{x}) - \mathbb{F}(\mathbf{y})| \leq \theta|\mathbf{x} - \mathbf{y}|$ .

Define the operator:

$$F_c(\mathbf{x}) := \mathcal{R}^T \hat{\mathbb{F}}(\mathbf{x}) + \mathcal{R}^T \mathbb{G}(\mathcal{R}\mathbf{x}) + c\mathbb{L}\mathbf{x}. \quad (32)$$

From (26), we know

$$\bar{\mathbb{A}} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} F_c(\hat{\mathbf{x}}) \\ \mathbf{0} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} \mathcal{R}^T A^T \boldsymbol{\lambda} \\ \mathbf{L}\boldsymbol{\lambda} \\ -A\mathcal{R}\hat{\mathbf{x}} - \mathbf{L}\mathbf{z} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ N_{\mathcal{R}_{\geq 0}^{N_m}}(\boldsymbol{\lambda}) \end{bmatrix}. \quad (33)$$

**Lemma 5.** Define  $c_{\min} = \frac{(\theta + \theta_0)^2}{4\mu s_2(L)} + \frac{\theta}{s_2(L)}$ .

$$\Psi = \begin{bmatrix} \frac{\mu}{N} & -\frac{\theta + \theta_0}{2\sqrt{N}} \\ -\frac{\theta + \theta_0}{2\sqrt{N}} & cs_2(L) - \theta \end{bmatrix}, \mu_{F_c} = s_{\min}(\Psi). \quad (34)$$

When  $c > c_{\min}$ , we have  $\mu_{F_c} = s_{\min}(\Psi) > 0$  and  $\bar{\mathbb{A}}$  is restricted monotone if Assumptions 1–5 hold.

**Proof of Lemma 5.** Let  $\mathbf{v}^* = \text{col}(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*) \in \text{zer}(\bar{\mathbb{A}})$  be any zero of  $\bar{\mathbb{A}}$ , which exists by Lemma 4. From (33), we can decompose  $\bar{\mathbb{A}}$  as a sum of three operators, two of which are monotone. Moreover, for any  $c > c_{\min}$ , we have  $\Psi \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^{N_n}$  by Lemma 1. Then,

$$\langle \mathbf{x} - \mathbf{x}^* | F_c(\mathbf{x}) - F_c(\mathbf{x}^*) \rangle \geq \mu_{F_c} \|\mathbf{x} - \mathbf{x}^*\|^2. \quad (35)$$

Hence, for any  $(v, u) \in \text{gra}(\bar{\mathbb{A}})$ , with  $v = \text{col}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda})$ , we obtain  $\langle v - v^* | u - \mathbf{0} \rangle \geq \mu_{F_c} \|\mathbf{x} - \mathbf{x}^*\|^2 \geq 0$ .  $\square$

**Lemma 6.** Define  $\mathcal{B} : \mathbb{R}^q \rightrightarrows \mathbb{R}^q$  as a restricted monotone in  $\mathcal{H}_\Phi$  and  $J_{\mathcal{B}}$  is firmly quasinonexpansive in  $\mathcal{H}_\Phi$ , where  $\mathcal{H}_\Phi$  is the Hilbert space induced by the inner product  $\langle \cdot, \cdot \rangle_\Phi$ . Then for any  $(v, u) \in \text{gra}(J_{\mathcal{B}})$ ,  $v^* \in \text{zer}(\mathcal{B}) = \text{fix}(J_{\mathcal{B}})$ , it holds that

$$\langle v - u | v - v^* \rangle_\Phi - \|u - v\|_\Phi^2 = \langle v - u | u - v^* \rangle_\Phi \geq 0. \quad (36)$$

**Proof of Lemma 6.** We use the definition of resolvent:  $v^* \in J_{\mathcal{B}}(v^*) \Leftrightarrow v^* + \mathcal{B}v^* \ni v^* \Leftrightarrow \mathbf{0} \in \mathcal{B}(v^*)$ . Moreover, for any  $(v, u) \in \text{gra}(J_{\mathcal{B}})$ , we have  $v - u \in \mathcal{B}(u)$ . Therefore, (36) follows from the restricted monotonicity of  $\mathcal{B}$  and some simple algebra. Finally, setting  $v = v^*$  in (36), we obtain that  $J_{\mathcal{B}}$  is single-valued.  $\square$

**Lemma 7.** Let  $\text{zer}(\mathcal{B}) \neq \emptyset$  by Lemma 6. For  $k \in \mathbb{N}$ , we assume that  $\beta^k \in [0, 2]$  and  $e^k \in \mathbb{R}^q$  such that  $\beta^k \|e^k\|_P \in \ell^1$ . If  $v^0 \in \mathbb{R}^q$ ,

$$(\forall k \in \mathbb{N}) \quad v^{k+1} = v^k + \beta^k (u^k - v^k + e^k), u^k \in J_{\mathcal{B}}(v^k). \quad (37)$$

Then we obtain Algorithm 1 by applying (13) to the operator  $\Phi^{-1}\bar{\mathbb{A}}$ , where

$$\Phi = \begin{bmatrix} c\hat{W} & \mathbf{0} & -\mathcal{R}^T A^T \\ \mathbf{0} & \mathbf{0} & -\mathbf{L} \\ -A\mathcal{R} & -\mathbf{L} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \alpha^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \gamma^{-1} \end{bmatrix} \quad (38)$$

is called a preconditioning matrix. It ensures that the agents will be able to compute the resulting iteration in a fully distributed manner.

We choose the step sizes  $\alpha = \text{diag}(\alpha_i I_{n_i})$ ,  $\tau = \text{diag}(\tau_i I_m)$  and  $\gamma = \text{diag}(\gamma_i I_m)$  such that  $\Phi > 0$ . This implies that  $\text{zer}(\Phi^{-1}\bar{\mathbb{A}}) = \text{zer}(\bar{\mathbb{A}})$ . The next lemma provides sufficient conditions for  $\Phi > 0$  based on Gershgorin's circle theorem.

**Lemma 8.** For any agent  $i \in \mathcal{I}$  and any  $\delta > 0$ , the preconditioning matrix  $\Phi$  in (38) is positive definite if

$$\begin{aligned} 0 < \alpha_i &\leq \left( \max_{j \in \{1, \dots, n_i\}} \sum_{k=1}^m | [A_i^T]_{jk} | + \delta \right)^{-1}, \\ 0 < \tau_i &\leq (2d_i + \delta)^{-1}, \\ 0 < \gamma_i &\leq \left( \max_{j \in \{1, \dots, m\}} \sum_{k=1}^{n_i} | [A_i]_{jk} | + 2d_i + \delta \right)^{-1}. \end{aligned} \quad (39)$$

Using Lemma 4 and Assumption 4, we can show that  $J_{\Phi^{-1}\bar{\mathbb{A}}}$  is single-valued by applying Equation (28) to  $\Phi^{-1}\bar{\mathbb{A}}$ .

**Lemma 9.** Let  $c > c_{\min}, c_{\min}$  as in Lemma 3. Then  $\Phi^{-1}\bar{\mathbb{A}}$  is restricted monotone in  $\mathcal{H}_\Phi$ .

It means that finding a zero point of  $\bar{\mathbb{A}}$  is equivalent to finding a variational GNE for problem (3). Moreover, since  $\Phi > 0$ , we have  $\text{zer}(\bar{\mathbb{A}}) = \text{zer}(\Phi^{-1}\bar{\mathbb{A}})$  by Lemma 9. Therefore, finding a zero point of  $\Phi^{-1}\bar{\mathbb{A}}$  is also equivalent to finding a variational GNE for problem (3). This establishes the equivalence between problems (3) and (22) under the condition that  $c > c_{\min}, c_{\min}$  as in Lemma 3.

**Theorem 1.** Assume that  $c > c_{\min}, c_{\min}$  are as defined in Lemma 8 and that the step sizes  $\alpha, \tau, \gamma$  satisfy Lemma 5. Then, Algorithm 1 generates a sequence  $(\mathbf{x}^k, \mathbf{z}^k, \boldsymbol{\lambda}^k)_{k \in \mathbb{N}}$  that converges to an equilibrium  $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*)$ .

**Proof of Theorem 1.** The set of inequalities (29) is equivalent to Equation (37) when  $\beta^k = 1$ ,  $e^k = 0$  for all  $k \in \mathbb{N}$ . By applying Lemma 9, we deduce that  $\Phi^{-1}\bar{\mathbb{A}}$  is restricted monotone on  $\mathcal{H}_\Phi$ . Then, we define

$$u^k = J_{\Phi^{-1}\bar{\mathbb{A}}}(v^k). \quad (40)$$

By Lemma 9,  $\Phi^{-1}\bar{\mathbb{A}}$  is restricted monotone in  $\mathcal{H}_\Phi$ , which implies  $u^k = v^{k+1}$ .

Let  $h^k := v^k + \beta^k(u^k - v^k)$ , so that  $v^{k+1} = h^k + \beta^k e^k$ . For all  $k \in \mathbb{N}$ ,

$$\|h^k - v^*\|_\Phi^2 = \|v^k - v^*\|_\Phi^2 - 2\beta^k \langle v^k - u^k | v^k - v^* \rangle_\Phi + (\beta^k)^2 \|u^k - v^k\|_\Phi^2. \quad (41)$$

Then by Lemma 7, we have

$$\|h^k - v^*\|_\Phi^2 \leq \|v^k - v^*\|_\Phi^2 - \gamma^k (2 - \beta^k) \|u^k - v^k\|_\Phi^2. \quad (42)$$

By the Cauchy–Schwarz inequality, it holds that

$$\|h^k - v^*\|_{\Phi} \leq \|v^k - v^*\|_{\Phi}. \quad (43)$$

By (43), the sequence  $(v^k)_{k \in \mathbb{N}}$  is bounded so that there is at least one cluster point such as  $\bar{v}$ . Define:

$$\overline{\mathbb{A}}_1 \begin{bmatrix} x \\ z \\ \lambda \end{bmatrix} = \begin{bmatrix} F_c(\hat{x}) \\ 0 \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} R^T A^T \lambda \\ L \lambda \\ -AR\hat{x} - Lz \end{bmatrix}. \quad (44)$$

$\overline{\mathbb{A}}_1$  consists of the sum of the first two equations of  $\overline{\mathbb{A}}$ . By (29) and (44), it holds that

$$\langle \overline{\mathbb{A}}_1(u^k) + \Phi(u^k - v^k) | v - u^k \rangle \geq 0. \quad (45)$$

Let  $\eta := \sup_{k \in \mathbb{N}} \|v^k - v^*\|_{\Phi} < \infty$  and  $\epsilon^k := 2\eta(\beta^k \|e^k\|_{\Phi}) + (\beta^k \|e^k\|_{\Phi})^2$ , for all  $k \in \mathbb{N}$ . Then,  $(\epsilon^k)_{k \in \mathbb{N}} \in \ell^1$ . Moreover, for all  $k \in \mathbb{N}$  we have

$$\begin{aligned} & \|v^{k+1} - v^*\|_{\Phi}^2 \\ & \leq (\|h^k - v^*\|_{\Phi} + \beta^k \|e^k\|_{\Phi})^2 \\ & \leq \|v^k - v^*\|_{\Phi}^2 - \beta^k (2 - \beta^k) \|u^k - v^k\|_{\Phi}^2 + \epsilon^k. \end{aligned} \quad (46)$$

Followed by recursion, it holds that

$$(\beta^k (2 - \beta^k) \|u^k - v^k\|_{\Phi}^2)_{k \in \mathbb{N}} \in \ell^1. \quad (47)$$

By (47),  $u^k - v^k$  converges to 0 as  $k \rightarrow \infty$ . Let  $l_k$  be a divergent subsequence such that  $v^{l_k} \rightarrow \bar{v}$ . Then, for any  $v \in \Omega \times \mathbb{R}^m \times \mathbb{R}_{\geq 0}^m$ , we have  $\langle \overline{\mathbb{A}}_1(\bar{v}) | v - \bar{v} \rangle \geq 0$  by continuity of  $\overline{\mathbb{A}}_1$ , which means  $\bar{v} \in \text{zer}(\overline{\mathbb{A}}_1) = \text{fix}(J_{\Phi^{-1}\overline{\mathbb{A}}_1})$ . By Lemma 6, any cluster point of  $v^k$  belongs to  $C$ . Hence,  $v^k$  converges to an equilibrium of (29).  $\square$

## 5. Numerical Studies

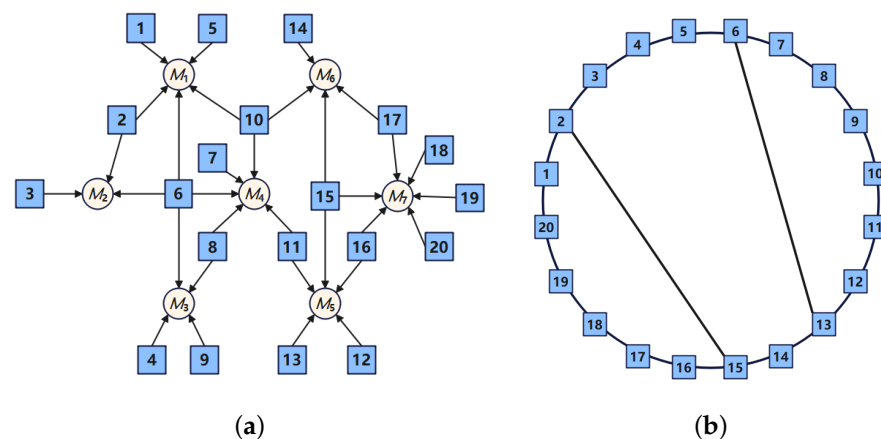
In this section, we will explore a network-based Nash–Cournot game [21] that is used to model the competition between  $N$  companies in  $m$  markets. This game is of particular interest because it has been shown to be an effective tool for analyzing competition between companies. The markets have shared affine constraints or equivalent global coupling affine constraints, which makes it possible to model the interactions between the companies in a realistic way. It is important to note that in this game, the agents can only communicate with their neighbors, and there is no central node with bidirectional communication to all participants. This makes the game even more challenging, as the agents must rely on their own resources to make decisions.

Although this game has been applied in networked Cournot games [15], it is worth noting that it assumes that the agents' decision information is global. However, in contrast, our work considers network structure and partial decision information, which leads to a more accurate representation of the real-world dynamics of competition between companies. This means that our research is more relevant to the real world, and can help us gain a deeper understanding of the complex interactions between the agents. By incorporating the network structure and partial decision information, we are able to gain a deeper understanding of the complex interactions between the agents, which can have significant implications for the overall performance of the companies in the markets. Additionally, our research can help to inform future policy decisions, as it provides a more accurate picture of the dynamics of competition between companies in networked markets.

### 5.1. Cournot Market Competition

Each firm  $i$  decides the quantity  $x_i \in \mathbb{R}^{n_i}$  of the commodity for  $n_i \leq m$  markets, subject to  $0^{n_i} \leq x_i \leq X_i$ . The maximum capacity of each market  $l = 1, \dots, m$  is  $r_l$ . Hence, we have a shared affine constraint  $Ax \leq r$ , where  $r = \text{col}((r_k)_{k=1, \dots, m})$  and  $A = [A_1 \dots A_N]$ . The matrix  $A_i \in \mathbb{R}^{m \times n_i}$  indicates which markets firm  $i$  enters. Specifically,  $[A_i]_{l,j} = 1$  if firm  $i$  enters market  $l$ , and  $[A_i]_{l,j} = 0$  otherwise, for all  $j = 1, \dots, n_i$  and  $l = 1, \dots, m$ . The profit function of each firm  $i$  is  $J_i(x_i, x_{-i}) = f_i(x_i, x_{-i}) + g_i(x_i)$ , where  $f_i(x_i, x_{-i}) = p(Ax)^T A_i x_i$  and  $g_i(x_i) = -x_i^T Q_i x_i - q_i^T x_i$ .  $g_i(x_i)$  is the production cost of firm  $i$ , and  $Q_i \in \mathbb{R}^{n_i \times n_i}$ ,  $Q_i > 0$ ,  $q_i \in \mathbb{R}^{n_i}$  are given parameters. The market price for each market and  $l = 1, \dots, m$  is given by  $[p(x)]_l = \bar{P}_l - \chi_l [Ax]_l$ , where  $\bar{P}_l, \chi_l > 0$  are constants.

In this simulation, we set the total number of participants to 20 and the number of markets to 7, i.e.,  $N = 20, m = 7$ . We defined the market structure using Figure 1a, which does not show the actual spatial relationships and distances among markets and firms. We also set  $n_i = 1$  for all  $i \in \mathcal{I}$ . Then the arrows in Figure 1a represent only the participation of firms in the markets; therefore, we have  $x = \text{col}((x_i))_{i \in \mathcal{I}} \in \mathbb{R}^n$ .

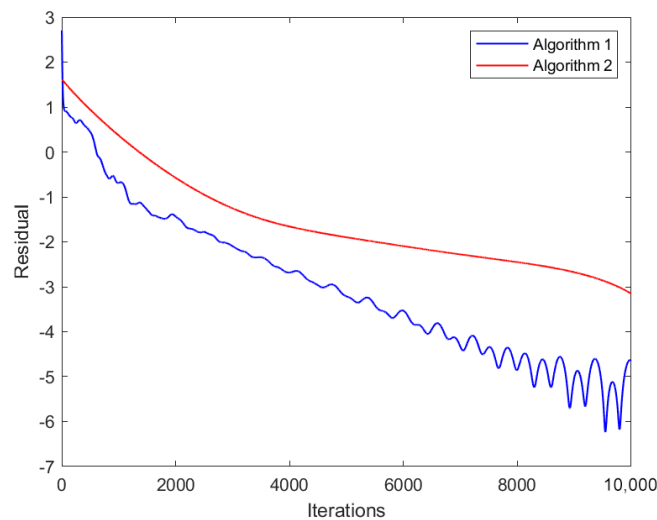


**Figure 1.** (a) Network Nash–Cournot game. (b) Communication graph  $G_c$ .

We considered  $m$  markets distributed across seven continents. Individual firms could not communicate with all other firms because of geographical location, communication technology, or company systems. The firms could communicate only with their neighbors on the communication graph  $G_c$ , shown in Figure 1b. Only connected firms  $i$  and  $j$  in the undirected communication graph  $G_c$  could exchange their information. We randomly selected  $r_l \in [1, 2]$ ,  $Q$  with diagonal elements in  $[1, 8]$ ,  $q_i \in [1, 2]$ ,  $\bar{P}_l \in [10, 20]$ ,  $\chi_l \in [1, 3]$ , and  $X_i \in [5, 10]$  for all  $i \in \mathcal{I}$  and  $l = 1, \dots, m$ .

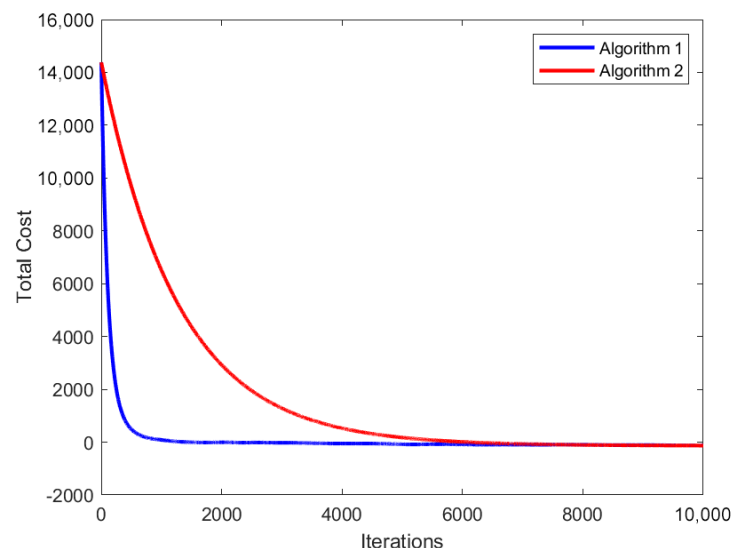
### 5.2. Numerical Results

The experimental settings described above satisfy all assumptions presented in [21]. We selected the step size in Lemma 8 to fulfill all conditions required by Theorem 1. To compare the performance of Algorithm 1 and the algorithm in [21], we conducted experiments using the same random initial condition for both algorithms. As shown in Figures 2 and 3, Algorithm 1 proposed in this paper converges faster than the algorithm in [21], which is referred to as Algorithm 2 in the following text.



**Figure 2.** Relative error  $\|x_k - x^*\|/\|x^*\|$  plot generated by Algorithm 1 and Algorithm 2 (the algorithm in [21]).

Figure 2 compares the convergence of the two algorithms under the partial decision setting by plotting the relative error of their decisions. The results show that Algorithm 1 has smaller relative errors than Algorithm 2 (the algorithm in [21]) for the same number of iterations, indicating a faster convergence rate for Algorithm 1.

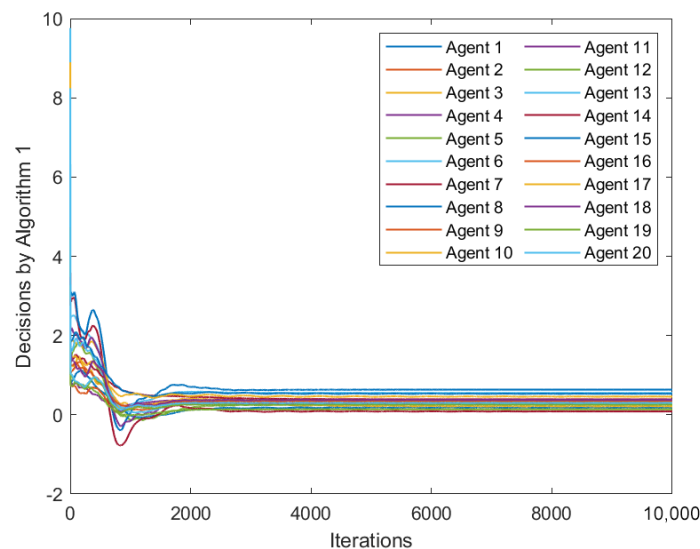


**Figure 3.** The total cost of all agents generated by Algorithm 1 and Algorithm 2 (the algorithm in [21]).

Figure 3 illustrates the trajectory of the total cost of all agents in the market corresponding to Algorithm 1 and Algorithm 2 (the algorithm in [21]), where the total cost is generated by  $\sum_{i=1}^N J_i(x_i, x_{-i})$ . The trajectory eventually converges to the same minimum value, indi-

cating the correctness and accuracy of Algorithm 1. Furthermore, the convergence of the trajectory represents the effectiveness of the algorithm in terms of optimizing the total cost of all agents in the market.

In Figure 4, we can observe the trajectory of each agent's decision in the Cournot market game, which has been solved by Algorithm 1. As we can see from the graph, the decision trajectories of all agents converge, which is a strong indication that the GNE obtained by executing Algorithm 1 can effectively minimize the cost incurred by each agent. It is worth noting that the convergence of decision trajectories is a critical aspect of the game theory that is often used to evaluate the effectiveness of a given algorithm. Therefore, the observed convergence in Figure 4 is an encouraging sign that Algorithm 1 can be a viable approach for solving similar problems in the future.



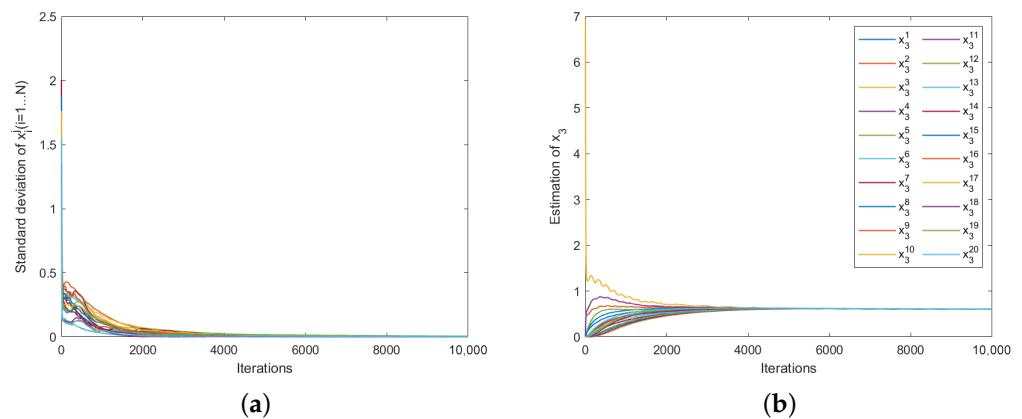
**Figure 4.** Trajectories of every agent's decision  $x_{i,k}$  generated by Algorithm 1.

During the iterative process of Algorithm 1, the estimated value of each agent's decision for all other agents is first obtained according to the formula  $x_{-i,k+1}^i = \frac{1}{1+\tau_i d_i} \left( x_{-i,k}^i + \tau_i \sum_{j=1}^N w_{i,j} x_{-i,k}^j \right)$ . This formula is used to compute the value of  $x_{-i,k+1}^i$ . Subsequently, the estimated decision value is used for the iteration of  $(x, z, \lambda)$ , which is a mathematical notation used to represent the values of the decision variables. When the algorithm converges, i.e., when the GNE is obtained, the estimated decision value of each agent for  $x_i$  is equal to the actual decision value of  $x_i$ , i.e.,  $x_i^j = x_i^i$  for  $j = 1, \dots, N$ . Figure 5a shows the trajectory of the standard deviation of each agent's estimated decision set for  $x_i$  with increasing iteration

times, which is generated by  $\sqrt{\frac{1}{N} \sum_{j=1}^N (x_i^j - \sum_{j=1}^N x_i^j)^2}$  for  $j = 1, \dots, N$ . It can be seen that the

standard deviation of the estimated decision set for each agent eventually converges to 0. This indicates that the results satisfy the condition that the estimated value of each agent's decision for  $x_i$  is equal to the actual decision value of  $x_i$ . As shown in Figure 5b, taking agent 3 as an example, the trajectory plot of all agents' estimated decision values for agent 3 is displayed, where  $x_3^3$  represents the actual value of agent 3's decision. The trajectory plot eventually converges to the same value, visually confirming the accuracy of Algorithm 1.





**Figure 5.** (a) Trajectories of the standard deviation of agents' estimations of  $x_{i,k}$  generated by Algorithm 1. (b) Trajectories of agents' estimations of  $x_{3,k}$  generated by Algorithm 1.

## 6. Conclusions

In this paper, we propose a distributed algorithm for games with shared coupling constraints based on the preconditioned proximal-point algorithm under partial decision information, which can converge with a fixed step size on arbitrarily connected graphs and is successfully applied to the GNE computation of the Cournot Market competition under partial decision information with a relatively fast convergence rate. A possible direction for future work is to study partial decision information sets and their impact on the memory efficiency of the algorithm. We will also explore how to predict only a subset of agents' decisions rather than all agents' decisions and study the convergence properties of the algorithm. Future work will further investigate this direction.

**Author Contributions:** Methodology, H.L. and Y.S.; Software, Z.W., M.C. and J.C.; Formal analysis, Z.W., H.L. and M.C.; Investigation, Z.W. and Y.S.; Resources, Z.W. and Y.S.; Data curation, Z.W., H.L. and J.T.; Writing—original draft, Z.W.; Supervision, Y.S.; Project administration, Z.W. and Y.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Saad, W.; Han, Z.; Poor, H.V.; Basar, T. Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications. *IEEE Signal Process. Mag.* **2012**, *29*, 86–105. [\[CrossRef\]](#)
2. Li, N.; Chen, L.; Dahleh, M.A. Demand response using linear supply function bidding. *IEEE Trans. Smart Grid* **2015**, *6*, 1827–1838. [\[CrossRef\]](#)
3. Grammatico, S. Dynamic control of agents playing aggregative games with coupling constraints. *IEEE Trans. Autom. Control* **2017**, *62*, 4537–4548. [\[CrossRef\]](#)
4. Debreu, G. A social equilibrium existence theorem. *Proc. Natl. Acad. Sci. USA* **1952**, *38*, 886–893. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Rosen, J.B. Existence and uniqueness of equilibrium points for concave n-person games. *Econom. J. Econom. Soc.* **1965**, *33*, 520–534. [\[CrossRef\]](#)
6. Facchinei, F.; Kanzow, C. Generalized Nash equilibrium problems. *Ann. Oper. Res.* **2010**, *175*, 177–211. [\[CrossRef\]](#)
7. Mastroeni, G.; Pappalardo, M.; Raciti, F. Generalized Nash equilibrium problems and variational inequalities in Lebesgue spaces. *Minimax Theory Appl.* **2020**, *5*, 47–64.
8. Yin, H.; Shanbhag, U.V.; Mehta, P.G. Nash equilibrium problems with congestion costs and shared constraints. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 4649–4654.

9. Zhu, M.; Frazzoli, E. Distributed robust adaptive equilibrium computation for generalized convex games. *Automatica* **2016**, *63*, 82–91. [\[CrossRef\]](#)
10. Tatarenko, T.; Kamgarpour, M. Learning generalized Nash equilibria in a class of convex games. *IEEE Trans. Autom. Control* **2018**, *64*, 1426–1439. [\[CrossRef\]](#)
11. Paccagnan, D.; Gentile, B.; Parise, F.; Kamgarpour, M.; Lygeros, J. Distributed computation of generalized Nash equilibria in quadratic aggregative games with affine coupling constraints. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 6123–6128.
12. Belgioioso, G.; Grammatico, S. Semi-decentralized Nash equilibrium seeking in aggregative games with separable coupling constraints and non-differentiable cost functions. *IEEE Control. Syst. Lett.* **2017**, *1*, 400–405. [\[CrossRef\]](#)
13. Belgioioso, G.; Grammatico, S. Projected-gradient algorithms for generalized equilibrium seeking in aggregative games are preconditioned forward-backward methods. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2188–2193.
14. Parise, F.; Gentile, B.; Lygeros, J. A distributed algorithm for average aggregative games with coupling constraints. *IEEE Trans. Control. Netw. Syst.* **2020**, *7*, 770–782. [\[CrossRef\]](#)
15. Yi, P.; Pavel, L. An operator splitting approach for distributed generalized Nash equilibria computation. *Automatica* **2019**, *102*, 111–121. [\[CrossRef\]](#)
16. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 408.
17. Yi, P.; Pavel, L. Asynchronous distributed algorithms for seeking generalized Nash equilibria under full and partial-decision information. *IEEE Trans. Cybern.* **2019**, *50*, 2514–2526. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Passacantando, M.; Raciti, F. A note on generalized Nash games played on networks. In *Nonlinear Analysis, Differential Equations, and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 365–380.
19. Bianchi, M.; Belgioioso, G.; Grammatico, S. Fast generalized Nash equilibrium seeking under partial-decision information. *Automatica* **2022**, *136*, 110080. [\[CrossRef\]](#)
20. Yi, P.; Pavel, L. Distributed generalized Nash equilibria computation of monotone games via double-layer preconditioned proximal-point algorithms. *IEEE Trans. Control. Netw. Syst.* **2018**, *6*, 299–311. [\[CrossRef\]](#)
21. Pavel, L. Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach. *IEEE Trans. Autom. Control* **2019**, *65*, 1584–1597. [\[CrossRef\]](#)
22. Gadjov, D.; Pavel, L. A passivity-based approach to Nash equilibrium seeking over networks. *IEEE Trans. Autom. Control* **2018**, *64*, 1077–1092. [\[CrossRef\]](#)
23. Salehisadaghiani, F.; Shi, W.; Pavel, L. Distributed Nash equilibrium seeking under partial-decision information via the alternating direction method of multipliers. *Automatica* **2019**, *103*, 27–35. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.