applied sciences

MDPI

Article

Distributed GNE Seeking under Global-Decision and Partial-Decision Information over Douglas-Rachford Splitting Method

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

Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China
* Correspondence: menggang_chen@163.com

Abstract: This paper develops an algorithm for solving the generalized Nash equilibrium problem (GNEP) in non-cooperative games. The problem involves a set of players, each with a cost function that depends on their own decision as well as the decisions of other players. The goal is to find a decision vector that minimizes the cost for each player. Unlike most of the existing algorithms for GNEP, which require full information exchange among all players, this paper considers a more realistic scenario where players can only communicate with a subset of players through a connectivity graph. The proposed algorithm enables each player to estimate the decisions of other players and update their own and others' estimates through local communication with their neighbors. By introducing a network Lagrangian function and applying the Douglas-Rachford splitting method (DR), the GNEP is reformulated as a zero-finding problem. It is shown that the DR method can find the generalized Nash equilibrium (GNE) of the original problem under some mild conditions.

Keywords: generalized Nash equilibrium (GNE); Douglas-Rachford; global decision information; partial decision information

Citation: Cheng, J.; Chen, M.; Li, H.; Shi, Y.; Wang, Z.; Tang, J. Distributed GNE Seeking under Global-Decision and Partial-Decision Information over Douglas-Rachford Splitting Method. *Appl. Sci.* **2023**, *13*, 7058. https://doi.org/10.3390/app13127058

Academic Editor: Radu Godina

Received: 12 April 2023
Revised: 24 May 2023
Accepted: 9 June 2023
Published: 12 June 2023

1. Introduction

In recent years, the generalized Nash equilibrium (GNE) [1,2], has been increasingly applied to solve practical problems. Examples include electricity market scheduling [3], peer-to-peer electricity market analysis [4] and energy trading in combined heat and power market [5]. In a noncooperative game, there are both shared and unshared constraints. These constraints cause players' private costs and feasible decision sets to interact with each other. In such a situation, it is necessary to establish rules for the players. The players make decisions by exchanging information according to the rules. A Nash equilibrium can be found as long as the rules are appropriate.

Algorithms for solving Nash equilibrium problems can be classified as centralized optimization algorithms and distributed optimization algorithms based on the way they solve optimization problems. In a centralized optimization algorithm, the entire problem is solved by a single entity with access to all the necessary information. In contrast, distributed optimization algorithms solve the problem by decomposing it into smaller sub-problems that are cooperatively solved by multiple players. This paper mainly studies distributed optimization algorithms. The information setting of distributed optimization algorithms is also divided into two types. The first type requires each player to publicly disclose their decision information. Another type only requires players to disclose information to their neighbors. In this paper, these rules are referred to as "global information setting" and "partial information setting", respectively.

Since the 1950s, research on GNEP has gradually become a hot topic [6–9]. For the solving of Nash equilibrium problems, many distributed algorithms have been studied in recent years, such as an algorithm based on the forward-backward operator partitioning

method [10], a triangular preconditioned proto-dual algorithm [11] and an algorithm fully adapted to monotone variational inequalities [12]. In [13], Lacra Pavela proposes a distributed primal-dual algorithm for the computation of generalized Nash equilibria (GNE) in noncooperative games in networked systems. The algorithm handles games with shared affine coupling constraints via operator splitting methods. In contrast, the Algorithm 1 proposed in this paper has a higher convergence speed. [14] reformulates the GNE problem as a variational inequality problem and solves it through the doubly augmented operator splitting method (DAOS). This method can handle nonsmooth and nonconvex objective functions and constraints. In contrast, the Algorithm 1 proposed in this paper solves the optimization problem by using unconstrained optimization and linear mapping, which has higher computational efficiency. Moreover, some studies have combined the ideas of machine learning to solve optimization problems. By combining extremum-seeking control (ESC) with learning algorithms, Feng Xiao, Xin Cai and Bo Wei proposed a GNE-seeking algorithm in [15]. This algorithm can be executed in a partial information setting and does not require knowledge of the explicit expression of the cost function. In most of the current research, only algorithms under the local information setting are studied. This paper discusses two forms of algorithms under two information settings. It also shows intuitively through experiments that the algorithm under the local information setting has higher efficiency.

Algorithm 1 GNE seeking under partial-decision

Initialize: For each player $i \in \mathcal{N}$, let $x_{i,0}^{[i]} \in \mathcal{R}^{n_i}$, $\boldsymbol{x}_{-i,0}^{[i]} \in \mathcal{R}^{n-n_i}$, $z_{i,0} \in \mathcal{R}^m$ and $\lambda_{i,0} \in \mathcal{R}^m$.

for $k = 1, 2, \ldots$ **do**

$\breve{\boldsymbol{x}}_{-i,k}^{[i]} = \boldsymbol{x}_{-i,k}^{[i]} + \frac{1}{2}c\alpha_i \sum_{j=1}^{N} w_{ij}(\boldsymbol{x}_{-i,k}^{[j]} - \boldsymbol{x}_{-i,k}^{[i]})$

$\breve{x}_{i,k} = prox_{\alpha_i f_i}\{x_{i,k} + \frac{1}{2}c\alpha_i \sum_{j=1}^{N} w_{ij}(x_{i,k}^{[j]} - \boldsymbol{x}_{i,k}) - \frac{1}{2}\alpha_i A_i^T \lambda_{i,k})\}$

$\breve{z}_{i,k} = z_{i,k} + \frac{1}{2}\tau_i \sum_{j=1}^{N} w_{ij}(\lambda_{j,k} - \lambda_{i,k})$

$\breve{\lambda}_{i,k} = \lambda_{i,k} - \frac{1}{2}\gamma_i A_i x_{i,k} - \frac{1}{2}\gamma_i \sum_{j=1}^{N} w_{ij}(z_{i,k} - z_{j,k}) - \gamma_i b + \gamma_i A_i \breve{x}_{i,k} - \gamma_i \sum_{j=1}^{N} w_{ij}(\breve{z}_{j,k} - \breve{z}_{i,k})$

$\widetilde{x}_{i,k} = 2\breve{x}_{i,k} - x_{i,k}$

$\widetilde{\boldsymbol{x}}_{-i,k}^{[i]} = 2\breve{\boldsymbol{x}}_{-i,k}^{[i]} - \boldsymbol{x}_{-i,k}^{[i]}$

$\widetilde{z}_{i,k} = 2\breve{z}_{i,k} - z_{i,k}$

$\widetilde{\lambda}_{i,k} = 2\breve{\lambda}_{i,k} - \lambda_{i,k}$

$\overline{\boldsymbol{x}}_{-i,k}^{[i]} = \widetilde{\boldsymbol{x}}_{-i,k}^{[i]} + \frac{1}{2}c\alpha_i \sum_{j=1}^{N} w_{ij}(\widetilde{\boldsymbol{x}}_{-i,k}^{[j]} - \widetilde{\boldsymbol{x}}_{-i,k}^{[i]})$

$\overline{x}_{i,k} = prox_{\alpha_i f_i}\{x_{i,k} + \frac{1}{2}c\alpha_i \sum_{j=1}^{N} w_{ij}(\widetilde{x}_{i,k}^{[j]} - \widetilde{\boldsymbol{x}}_{i,k}) - \frac{1}{2}\alpha_i A_i^T \widetilde{\lambda}_{i,k}\}$

$\overline{z}_{i,k} = \widetilde{z}_{i,k} + \frac{1}{2}\tau_i \sum_{j=1}^{N} w_{ij}(\widetilde{\lambda}_{j,k} - \widetilde{\lambda}_{i,k})$

$\overline{\lambda}_{i,k} = \lambda_{i,k} - \frac{1}{2}\gamma_i A_i \widetilde{x}_{i,k} - \frac{1}{2}\gamma_i \sum_{j=1}^{N} w_{ij}(\widetilde{z}_{i,k} - \widetilde{z}_{j,k}) - \gamma_i b + \gamma_i A_i \overline{x}_{i,k} + \gamma_i \sum_{j=1}^{N} w_{ij}(\overline{z}_{i,k} - \overline{z}_{j,k})$

$x_{i,k+1} = x_{i,k} + 2r_k \overline{x}_{i,k} - 2r_k \breve{x}_{i,k}$

$\boldsymbol{x}_{-i,k+1}^{[i]} = \boldsymbol{x}_{i,k} + 2r_k \overline{\boldsymbol{x}}_{-i,k}^{[i]} - 2r_k \breve{\boldsymbol{x}}_{-i,k}^{[i]}$

$z_{i,k+1} = z_{i,k} + 2r_k \overline{z}_{i,k} - 2r_k \breve{z}_{i,k}$

$\lambda_{i,k+1} = \lambda_{i,k} + 2r_k \overline{\lambda}_{i,k} - 2r_k \breve{\lambda}_{i,k}$

end for

Retuen: The sequence $(x_{i,k}^{[i]}, \boldsymbol{x}_{-i,k}^{[i]})_{k=1}^{\infty}$ will eventually approximate the optimal solution.

The Douglas-Rachford splitting method used in this paper can handle nonsmooth convex functions and even non-convex functions, in addition to being able to use random block coordinate strategies and asynchronous execution to improve efficiency and robustness. This approach has been widely used to study graph coloring [16], compression perception [17], and image denoising [18].

The contributions of this paper can be summarized as follows:

- Two new variants of the Douglas-Rachford algorithm are proposed in this paper, denoted as Algorithms 1 and 2, which can solve Nash equilibrium problems.
- Two pairs of new splitting operators are introduced for Algorithms 1 and 2, respectively, which can ensure the convergence and efficiency of the algorithms.
- Algorithm 2 is proved in this paper, and it has a linear convergence rate under the global information setting, while Algorithm 1 has a faster convergence rate under the local information setting.
- The theoretical results in this paper are verified by numerical experiments and the advantages of Algorithm 1 are shown by comparison with the existing method.

Algorithm 2 GNE seeking under global-decision

Initialize: For each player $i \in \mathcal{N}$, let $x_{i,0} \in \mathcal{R}^{n_i}$, $z_{i,0} \in \mathcal{R}^m$ and $\lambda_{i,0} \in \mathcal{R}^m$

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

$\check{x}_{i,k} = \arg\min_x(\frac{1}{2}\|x - (x_{i,k} - \alpha_i A_i^T \lambda_{i,k})\|^2 + \alpha_i f_i(x_i, x_{-i}))$

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

$\check{\lambda}_{i,k} = \lambda_{i,k} - \frac{1}{2}\gamma_i A_i x_{i,k} - \frac{1}{2}\gamma_i \sum_{j=1}^N w_{ij}(z_{i,k} - z_{j,k}) - \gamma_i b_i + \gamma_i A_i \check{x}_{i,k} + \gamma_i \sum_{j=1}^N w_{ij}(\check{z}_{i,k} - \check{z}_{j,k})$

$\widetilde{x}_{i,k} = 2\check{x}_{i,k} - x_{i,k}$

$\widetilde{z}_{i,k} = 2\check{z}_{i,k} - z_{i,k}$

$\widetilde{\lambda}_{i,k} = 2\check{\lambda}_{i,k} - \lambda_{i,k}$

$\overline{x}_{i,k} = prox_{a_i g_i}\{\widetilde{x}_{i,k} - \frac{1}{2}\alpha_i A_i^T \widetilde{\lambda}_{i,k}\}$

$\overline{z}_{i,k} = \frac{1}{2}\gamma_i \sum_{j=1}^N w_{ij}(\widetilde{\lambda}_{j,k} - \widetilde{\lambda}_{i,k}) + \widetilde{z}_{i,k}$

$\overline{\lambda}_{i,k} = proj_{\mathcal{R}^m_{\geq 0}}(\widetilde{\lambda}_{i,k} - \frac{1}{2}\gamma_i A_i \widetilde{x}_{i,k} - \frac{1}{2}\gamma_i \sum_{j=1}^N w_{ij}(\widetilde{z}_{i,k} - \widetilde{z}_{j,k}) + \gamma_i A_i \overline{x}_{i,k} + \gamma_i \sum_{j=1}^N w_{ij}(\overline{z}_{i,k} - \overline{z}_{j,k}))$

$x_{i,k+1} = x_{i,k} + 2r_k \overline{x}_{i,k} - 2r_k \widetilde{x}_{i,k}$

$z_{i,k+1} = z_{i,k} + 2r_k \overline{z}_{i,k} - 2r_k \widetilde{z}_{i,k}$

$\lambda_{i,k+1} = \lambda_{i,k} + 2r_k \overline{\lambda}_{i,k} - 2r_k \widetilde{\lambda}_{i,k}$

end for

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

In this paper, Section 2 derives the KKT conditions of the variational problem of the original game using dual analysis method; Section 3 proposes distributed Algorithms 1 and 2 based on DR; Section 4 analyzes the convergence of the two algorithms; Section 5 presents the simulation results of these two algorithms; and Section 6 summarizes the whole paper and looks forward to the future.

2. Preliminary Acquaintance

2.1. Graph Theory

The graph $\mathcal{G}$ is mainly represented by two sets, namely the set of vertices $\mathcal{N}$ and the set of edges ε, i.e., $\mathcal{G} = \{\mathcal{N}, \varepsilon\}$, where $\mathcal{N} = \{1, \ldots, N\}$ and $\varepsilon \subset \mathcal{N} \times \mathcal{N}$. A player i and a player j are said to be neighbors of i if they can exchange information, and this relationship is represented in the connectivity graph as $(i, j) \in \varepsilon$. Moreover, the set of neighbors of player i is $\mathcal{N}_{-i} = \{j | (i, j) \in \varepsilon\}$. A path in a graph $\mathcal{G}$ is an interleaved sequence of vertices in $\mathcal{N}$ and edges in ε, where the end point of each edge is the start point of the next edge. If $\forall i, j \in \mathcal{N}, (i, j) \in \varepsilon$, then the graph $\mathcal{G}$ can be called a connected graph. Define $\mathcal{R}$ as the set of real numbers then the adjacency matrix of the graph $\mathcal{G}$ is $W = [w_{ij}] \in \mathcal{R}^{N \times N}$, where $\mathcal{R}$ denotes the set of real numbers, and whether the value of w_{ij} is greater than zero indicates whether there exists a path between i and j. Let $d_i = \sum_{j=1}^N w_{ij}$, and the weighted degree matrix of the adjacency matrix W is $WD = \text{diag}(d_i)_{i \in \mathcal{N}}$. Then $L = WD - W$ is the Laplace matrix of the graph $\mathcal{G}$.

Suppose $\mathcal{G}$ is a connected graph and is undirected, i.e., $W = W^T$; then L has a simple eigenvalue $s_1 = 0$ and all other eigenvalues are greater than zero. Arrange all eigenvalues in descending order as: $s_N > s_{N-1} > ... > s_2 > 0$. Let $d^* = \max\{d_1, ..., d_N\}$, then $d^* \leq s_N \leq 2d^*$ [19].

2.2. Game Model

This subsection studies the noncooperative game model on an undirected connected graph $\mathcal{G}$. Player i in the set $\mathcal{N}$ learns other players' decisions x_{-i} through paths in the graph and then controls the local decision x_i. Then, the global decision set x is $\mathrm{col}(x_1, ...x_N)$. Denote $\sum_{i=1}^{N} A_i x_i \leq \sum_{i=1}^{N} b_i$ as global coupling constraint, then we get the cost function:

$$\forall i \in \mathcal{N}, \quad \begin{cases} \underset{x_i \in R^{n_i}}{\arg\min} & \mathbb{J}_i(x_i, x_{-i}) = f_i(x_i, x_{-i}) + g_i(x_i), \\ \text{s.t.} & \sum_{i=1}^{N} A_i x_i \leq \sum_{i=1}^{N} b_i, \end{cases} \tag{1}$$

where $i \in \mathcal{N}$, $A_i \in \mathcal{R}^{m \times n_i}$, $b_i \in \mathcal{R}^m$, $x_i \in \Omega_i$. $\Omega_i \subseteq \mathcal{R}^{n_i}$ is private data of player i, and $x_{-i} = \mathrm{col}(x_j)_{j \in \mathcal{N}, j \neq i}$ denotes the decisions of all players except player i, $x_{-i} \in \mathcal{R}^{n-n_i}$, $n = \sum_{i=1}^{N} n_i$.

Assumption 1.

1. *The domain of the function $dom(g_i) = \Omega_i \subseteq \mathcal{R}^{n_i}$ is nonempty, compact and convex.*
2. *g_i is lower semicontinuous on its domain.*
3. *The function satisfies the convexity inequality on its domain.*

Denote

$$\mathcal{X} := \prod_{i=1}^{N} \Omega_i \cap \{x | x \in R^n, \sum_{i=1}^{N} A_i x_i - b_i \leq 0\}, \tag{2}$$

which is the set of feasible values of the global decision vector x. The feasible decision set of agent i is $\mathcal{X}_i(x_{-i}) := \{x_i \in \Omega_i | (x_i, x_{-i}) \in \mathcal{X}\}$.

Assumption 2.

1. *Every locally feasible set $\mathcal{X}_i$ is a non-empty closed convex set.*
2. *Globally shared feasible set $\mathcal{X}$ is nonempty.*
3. *For any feasible point, there exists a nonzero vector that is orthogonal to all gradients of the constraint functions, i.e., (MFCQ) holds.*

Let $A = [A_1, ..., A_N] \in \mathcal{R}^{m \times N}$ and $b = \sum_{i=1}^{N} b_i$, then the objective function is equal to

$$\forall i \in \mathcal{N}, \quad \begin{cases} \underset{x_i \in R^{n_i}}{\arg\min} & \mathbb{J}_i(x_i, x_{-i}) = f_i(x_i, x_{-i}) + g_i(x_i), \\ \text{s.t.} & Ax \leq b, \end{cases} \tag{3}$$

which consists of two parts, i.e., $f_i(x_i, x_{-i})$ and $g_i(x_i)$, the former representing the local impact of the global decision set and the latter representing the local cost. A GNE of game (3) is (4)

$$\forall i \in \mathcal{N}, \quad \begin{cases} x_i^* \in \underset{x_i \in R^{n_i}}{\arg\min} \mathbb{J}_i(x_i, x_{-i}^*), \\ \text{s.t. } x_i \in \mathcal{X}_i(x_{-i}^*). \end{cases} \tag{4}$$

Assumption 3.

1. *When given x_{-i}, Ω_i is a non-empty compact convex set and $\forall i \in \mathcal{N}$, $\mathbb{J}_i(x_i, x_{-i})$ is*
2. *continuously differentiable at x_i.*
3. *$\mathbb{J}_i(x_i, x_{-i})$ is a convex function and $\mathcal{X}$ is a nonempty set satisfying Slater-constraint-qualification.*

2.3. Nash Equilibrium for Variational Problems(V-GNE) and KKT Condition

If the original problem is a convex optimization problem, then v-GNE and GNE are equivalent. If the original problem is a nonconvex optimization problem, then v-GNE and GNE may be different, but v-GNE can be used as an approximate solution or a lower bound for GNE [8].

Define a local Lagrangian function for agent i as $\mathbb{L}_i(x_i, x_{-i}, \lambda_i) = \mathbb{J}_i(x_i, x_{-i}) + \lambda_i^T(Ax - b)$. Let $\boldsymbol{\lambda} = \mathrm{col}((\lambda_i)_{i\in\mathcal{N}}) \in \mathcal{R}^{Nm}$, $\boldsymbol{L} = L \otimes I_m \in \mathcal{R}^{Nm\times Nm}$, where L is the Laplacian matrix and I_m is a unit matrix of order m. In order to solve for v-GNEs, it is necessary to add the equality constraint $\boldsymbol{L\lambda} = \boldsymbol{0}$, and the augmented Lagrangian function is $\mathbb{L}_i(x_i, x_{-i}, \lambda_i, z_i) = \mathbb{J}_i(x_i, x_{-i}) + \lambda_i^T(Ax - b) + \sum_{i=1}^N \sum_{j=1}^N z_i^T L_{ij}\lambda_j$, where $z_i \in \mathcal{R}^m$ serves to satisfy the coupling constraints and coordinate the local multipliers λ_i to bring them to consensus. Let $\boldsymbol{z} = \mathrm{col}(z_1, z_2, ..., z_N) \in \mathcal{R}^{Nm}$, there have

$$\mathbb{L}_i(x_i, x_{-i}, \lambda_i, z_i) = \mathbb{J}_i(x_i, x_{-i}) + \lambda_i^T(Ax - b) + \boldsymbol{z}^T\boldsymbol{L\lambda}, \quad i \in \mathcal{N}. \tag{5}$$

Suppose (x_i^*, x_{-i}^*) is the optimal solution of (5) and it follows from the definition of the generalized Lagrangian function that (x_i^*, x_{-i}^*) is also the variational GNE of the game (3), then x^* satisfies Formula (4). Let $\boldsymbol{A} = \mathrm{diag}(A_i)_{i\in\mathcal{N}}$, $\boldsymbol{b} = \mathrm{col}(b_i)_{i\in\mathcal{N}}$, when $\lambda_1 = \lambda_2 = ... = \lambda_N = \lambda^*$, the KKT condition for a v-GNE is

$$\begin{cases} 0 \in \mathbb{F}(x) + \mathbb{G}(x) + A^T\boldsymbol{\lambda}, \\ 0 = \boldsymbol{L\lambda}, \\ 0 \in \boldsymbol{b} + \boldsymbol{L\lambda} + N_{\mathbb{R}^{Nm}_{\geq 0}}(\boldsymbol{\lambda}) - \boldsymbol{A}x - \boldsymbol{L}z, \end{cases} \tag{6}$$

where $\mathbb{F}(x) = \mathrm{col}(\nabla_{x_i} f_i(x_i, x_{-i}))_{i\in\mathcal{N}}$, $\mathbb{G}(x) = \mathrm{col}(\partial_{x_i} g_i(x_i))_{i\in\mathcal{N}}$.

3. Distributed Algorithm

3.1. Global Information Setting

In this subsection, the derivation of Algorithm 2 is given based on the assumption that the global decision set of the game model is public. Next, we need to use the KKT condition obtained in the previous section to derive two operators. Using these two operators and a positive definite matrix, we can apply the DR operator splitting method to obtain Algorithm 2. Equation (10) is the standard form of the DR operator splitting method.

Let $\boldsymbol{v} = [x^T, z^T, \lambda^T]^T$ and define $\mathbb{A}$, $\mathbb{B}$ as

$$\mathbb{A} : \boldsymbol{v} \longmapsto \begin{bmatrix} \mathbb{F}(x) \\ \boldsymbol{0} \\ b \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} \\ -\boldsymbol{A} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix}\boldsymbol{v}, \tag{7}$$

$$\mathbb{B} : \boldsymbol{v} \longmapsto \begin{bmatrix} \mathbb{G}(x) \\ \boldsymbol{0} \\ N_{\mathbb{R}^{Nm}_{\geq 0}}(\boldsymbol{\lambda}) \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} \\ -\boldsymbol{A} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix}\boldsymbol{v}, \tag{8}$$

then the KKT condition (6) is equal to $0 \in \mathbb{A}(\boldsymbol{v}) + \mathbb{B}(\boldsymbol{v})$. Denote Φ as

$$\Phi = \frac{1}{2}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{L} \\ -\boldsymbol{A} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \alpha^{-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tau^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \gamma^{-1} \end{bmatrix}, \tag{9}$$

where $\alpha = \mathrm{diag}(\alpha_i\boldsymbol{I})$, $\tau = \mathrm{diag}(\tau_i\boldsymbol{I})$, $\gamma = \mathrm{diag}(\gamma_i\boldsymbol{I})$ are fixed step in the iteration, and r_k meets $\sum_{k\in\mathcal{N}} r_k(1 - r_k) = \infty$, $(0 < r_k < 1)$. Then we can design the Algorithm 2:

$$\begin{aligned}
&\breve{v} = (Id + \Phi^{-1}\mathbb{A})^{-1}(\breve{v}_k),\\
&\widetilde{v}_k = 2\breve{v} - v_k,\\
&\overline{v}_k = (Id + \Phi^{-1}\mathbb{B})^{-1}(\breve{v}_k),\\
&v_{k+1} = v_k + 2r_k(\overline{v}_k - \breve{v}).
\end{aligned} \tag{10}$$

3.2. Partial Information Setting

In this subsection, players are reluctant to disclose their decisions due to privacy concerns, i.e., player i is not aware of other players' decisions x_{-i}. In this case, each player can only find the GNE by exchanging local information with their trusted neighbors through the connectivity graph $\mathcal{G}$ and minimizing their objective function with the obtained information. Next, we will obtain an equivalent form of the variational problem under certain assumptions. By following a similar process as in the previous subsection, we can derive Algorithm 1.

Assumption 4.
$\mathcal{G}$ is undirected and connected and it has no self-loops.

Define $\boldsymbol{x}^{[i]}$ to denote the estimate of agent i for GNE x^*, where $\boldsymbol{x}^{[i]} = \mathrm{col}(x_j^{[i]})_{j\in\mathcal{N}}$ and $\boldsymbol{x}_{-i}^{[i]} = \mathrm{col}(x_j^{[i]})_{j\in\mathcal{N},j\neq i}$. We next investigated under what conditions the player can use the estimator $\boldsymbol{x}^{[i]}$ to minimize the objective function and why this can be done.

Remark 1. *$x_j^{[i]}$ represent the state estimate of i versus j, and $x_i^{[i]} = x_i$. When $\forall(i,j)$, $\boldsymbol{x}^{[i]} = \boldsymbol{x}^{[j]}$, all the agents reach an agreement.*

The variational problem of the original game problem (3) can be written as the following optimization problem:

$$\begin{cases}
\underset{x\in\mathcal{R}^n}{\arg\min} & \langle \mathbb{F}(x^*), x\rangle + g(x),\\
\text{s.t.} & \boldsymbol{A}x \leq \boldsymbol{b},\\
& \boldsymbol{L}\boldsymbol{\lambda} = 0,
\end{cases} \tag{11}$$

where $\mathbb{F}(x) = \mathrm{col}(\nabla_{x_i} f_i(x_i, x_{-i}))_{i\in\mathcal{N}}$, $g(x) = \sum_{i=1}^{N}(g_i(x_i))$, $\boldsymbol{A} = \mathrm{diag}(A_1, A_2, ..., A_N)$, $\boldsymbol{b} = \mathrm{col}(b_1, ..., b_N)$. In order that the decision variable x_i and the estimation variable $x_{-i}^{[i]}$ of agent i can be selected from $\boldsymbol{x}^{[i]}$, define two linear mappings: let

$$R_i = \begin{bmatrix} \mathbf{0}_{n_i\times n_{<i}} & I_{n_i} & \mathbf{0}_{n_i\times n_{>i}} \end{bmatrix}, \tag{12}$$

$$S_i = \begin{bmatrix} I_{n_{<i}} & \mathbf{0}_{n_{<i}\times n_i} & \mathbf{0}_{n_{<i}\times n_{>i}} \\ \mathbf{0}_{n_{>i}\times n_{<i}} & \mathbf{0}_{n_{>i}\times n_i} & I_{n_{>i}} \end{bmatrix}, \tag{13}$$

where $n_{<i} = \sum_{j<i,j\in\mathcal{N}} n_j$, $n_{<i} = \sum_{j>i,j\in\mathcal{N}} n_j$, i.e.,$R_i \in \mathcal{R}^{n_i\times n} : x \longmapsto x_i$ and $S_i \in \mathcal{R}^{(n-n_i)\times n} : x \longmapsto x_{-i}$. Hence $R_i\boldsymbol{x}^{[i]} = x_i^{[i]} = x_i$ and $S_i\boldsymbol{x}^{[i]} = \boldsymbol{x}_{-i}^{[i]}$. Furthermore, these two operators satisfy the following formulas:

$$\begin{aligned}
&R_i S_i^T = \mathbf{0}_{n_i\times(n-n_i)},\\
&R_i R_i^T = I_{n_i},\\
&S_i S_i^T = I_{(n-n_i)\times(n-n_i)},\\
&S_i^T S_i + R_i R_i^T = I_n.
\end{aligned} \tag{14}$$

Define $\hat{\boldsymbol{x}} = \text{col}(\boldsymbol{x}^{[i]})_{i=1}^{N}$, $\hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*) = \text{col}(\nabla_{x_i} f_i(x_i, \boldsymbol{x}_{-i}^{[i]*}))_{i\in\mathcal{N}}$, if $\forall(i,j) \in \varepsilon, \boldsymbol{x}^{[i]} = \boldsymbol{x}^{[j]}$, then $\hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*) = \mathbb{F}(x^*)$. With $R_i \boldsymbol{x}^{[i]} = x_i$ and $\boldsymbol{R} = \text{diag}(R_i)$, the stacking decision x is equivalent to $\boldsymbol{R}\hat{\boldsymbol{x}}$. Since $\forall(i,j) \in \varepsilon, \boldsymbol{x}^{[i]} = \boldsymbol{x}^{[j]} \Leftrightarrow (L \bigotimes I_n)\hat{\boldsymbol{x}} = \boldsymbol{0}$, let $\hat{\boldsymbol{L}} = (L \bigotimes I_n) \in \mathcal{R}^{Nn\times Nn}$, then (11) is equivalent to

$$\begin{cases} \underset{x\in\mathcal{R}^n}{\arg\min} & \left\langle \hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*), \boldsymbol{R}\hat{\boldsymbol{x}} \right\rangle + g(\boldsymbol{R}\hat{\boldsymbol{x}}), \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{R}\hat{\boldsymbol{x}} \le \boldsymbol{b}, \\ & \boldsymbol{L}\boldsymbol{\lambda} = \boldsymbol{0}, \\ & \hat{\boldsymbol{L}}\hat{\boldsymbol{x}} = \boldsymbol{0}. \end{cases} \tag{15}$$

The Lagrangian function of the optimization problem (15) is defined as $\hat{\mathbb{L}} = \left\langle \hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*), \boldsymbol{R}\hat{\boldsymbol{x}} \right\rangle + g(\boldsymbol{R}\hat{\boldsymbol{x}}) + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{R}\hat{\boldsymbol{x}} - \boldsymbol{b}) + \boldsymbol{z}^T\boldsymbol{L}\boldsymbol{\lambda} + \frac{c}{2}\hat{\boldsymbol{x}}^T\hat{\boldsymbol{L}}\hat{\boldsymbol{x}}$. Let $\hat{\mathbb{G}}(\boldsymbol{R}\hat{\boldsymbol{x}}) = \text{col}(\partial_{\hat{\boldsymbol{x}}} g(\boldsymbol{R}\hat{\boldsymbol{x}}))_{i\in\mathcal{N}}$, the KKT condition is

$$\begin{cases} \boldsymbol{0} \in \boldsymbol{R}^T\hat{\mathbb{F}}(\hat{\boldsymbol{x}}) + \boldsymbol{R}^T\hat{\mathbb{G}}(\boldsymbol{R}\hat{\boldsymbol{x}}) + \boldsymbol{R}^T\boldsymbol{A}^T\boldsymbol{\lambda} + c\hat{\boldsymbol{L}}\hat{\boldsymbol{x}}, \\ \boldsymbol{0} = \boldsymbol{L}\boldsymbol{\lambda}, \\ \boldsymbol{0} \in \boldsymbol{A}\boldsymbol{R}\hat{\boldsymbol{x}} - \boldsymbol{b} + \boldsymbol{L}\boldsymbol{z} - N_{\mathbb{R}^{Nm}_{\ge 0}}(\boldsymbol{\lambda}). \end{cases} \tag{16}$$

Next, according to the KKT condition (16), the operator $\mathbb{T}$ can be defined as:

$$\mathbb{T} : \begin{bmatrix} \hat{\boldsymbol{x}} \\ \boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} \longrightarrow \begin{bmatrix} \boldsymbol{R}^T\hat{\mathbb{F}}(\hat{\boldsymbol{x}}) + \boldsymbol{R}^T\hat{\mathbb{G}}(\boldsymbol{R}\hat{\boldsymbol{x}}) + \boldsymbol{R}^T\boldsymbol{A}^T\boldsymbol{\lambda} + c\hat{\boldsymbol{L}}\hat{\boldsymbol{x}} \\ \boldsymbol{L}\boldsymbol{\lambda} \\ -\boldsymbol{A}\boldsymbol{R}\hat{\boldsymbol{x}} + \boldsymbol{b} + \boldsymbol{L}\boldsymbol{z} + N_{\mathbb{R}^{Nm}_{\ge 0}}(\boldsymbol{\lambda}) + \boldsymbol{L}\boldsymbol{\lambda} \end{bmatrix}. \tag{17}$$

Let $\boldsymbol{\omega} = \text{col}(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{\lambda})$; the KKT condition can be written as $0 \in \mathbb{T}(\boldsymbol{\omega})$, then the solution of optimization problem (15) is converted into the operator zero-seeking problem. If $\text{col}(\hat{\boldsymbol{x}}^*, \boldsymbol{z}^*, \boldsymbol{\lambda}^*)$ is a zero of the operator $\mathbb{T}$, then $\boldsymbol{R}\hat{\boldsymbol{x}}^*$ is a v-GNE of the original game problem (3). The operator $\mathbb{T}$ is partitioned into $\overline{\mathbb{A}}$, $\overline{\mathbb{B}}$ as follows :

$$\overline{\mathbb{A}} : \boldsymbol{\omega} \longrightarrow \begin{bmatrix} \boldsymbol{R}^T\hat{\mathbb{F}}(\hat{\boldsymbol{x}}) \\ \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} c\hat{\boldsymbol{L}} & \boldsymbol{0} & \boldsymbol{R}^T\boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} \\ -\boldsymbol{A}\boldsymbol{R} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\omega}, \tag{18}$$

$$\overline{\mathbb{B}} : \boldsymbol{\omega} \longrightarrow \begin{bmatrix} \boldsymbol{R}^T\hat{\mathbb{G}}(\boldsymbol{R}\hat{\boldsymbol{x}}) \\ \boldsymbol{0} \\ N_{\mathbb{R}^{Nm}_{\ge \boldsymbol{0}}}(\boldsymbol{\lambda}) \end{bmatrix} + \frac{1}{2}\begin{bmatrix} c\hat{\boldsymbol{L}} & \boldsymbol{0} & \boldsymbol{R}^T\boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} \\ -\boldsymbol{A}\boldsymbol{R} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\omega}. \tag{19}$$

In view of the partition operators $\overline{\mathbb{A}}$, $\overline{\mathbb{B}}$. To obtain the distributed algorithm, a matrix $\overline{\Phi}$ is defined, and this matrix is required to be positive definite. One choice is

$$\overline{\Phi} = \frac{1}{2}\begin{bmatrix} -c\hat{\boldsymbol{L}} & \boldsymbol{0} & -\boldsymbol{R}^T\boldsymbol{A}^T \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{L} \\ -\boldsymbol{A}\boldsymbol{R} & -\boldsymbol{L} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \overline{\alpha}^{-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \overline{\tau}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \overline{\gamma}^{-1} \end{bmatrix}, \tag{20}$$

where $\overline{\alpha} = diag(\overline{\alpha}_i I)$, $\overline{\tau} = diag(\overline{\tau}_i I)$, $\overline{\gamma} = diag(\overline{\gamma}_i I)$, with $\overline{\alpha}_i$, $\overline{\tau}_i$ and $\overline{\gamma}_i$ for $i \in \mathcal{N}$. Substituting $\mathbb{A} = \overline{\mathbb{A}}$, $\mathbb{B} = \overline{\mathbb{B}}$, $\Phi = \overline{\Phi}$ and $v = \boldsymbol{\omega}$ into (10), the distributed algorithm under the local information setting, i.e., Algorithm 1 can be derived.

In Algorithm 1, both the local decision x_i and the local estimate $\boldsymbol{x}_{-i}^{[i]}$ are updated by the Laplace matrix, which causes the global estimate $\boldsymbol{x}^{[i]}$ of agent i get closer to its neighbors, and the gradient descent causes the local cost function to decrease. The range of values of the step size parameter α_i, τ_i, γ_i depends on the positive definiteness of the matrix $\overline{\Phi}$, which can be found analytically using the Gershgorin. Note that the update step in the algorithm r_k needs to meet two conditions according to the requirements of the Douglas-Rachford method, i.e., $0 < r_k < 1$ and $\sum_{k=1}^{\infty} r_k(1 - r_k) = \infty$.

Remark 2. *In the pseudo-code, the input parameters are* $x_{i,0}^{[i]}$, $\boldsymbol{x}_{-i,0}^{[i]}$, $z_{i,0}$ *and* $\lambda_{i,0}$, *where* $(x_{i,0}^{[i]}, \boldsymbol{x}_{-i,0}^{[i]})$ *is player i's estimate of the global decision set at iteration 0 and the output parameter is* $(x_{j,k}^{[j]}, \boldsymbol{x}_{-j,k}^{[j]})$, *which is player i's final estimate of the global decision set.*

Observing the output parameters $(x_{j,k}^{[j]}, \boldsymbol{x}_{-j,k}^{[j]})$ after the algorithm iterations are completed reveals that eventually, all players have the same estimate of the global decision set, i.e., $\forall i, j \in \mathcal{N}, \quad (x_{i,k}^{[i]}, \boldsymbol{x}_{-i,k}^{[i]}) = (x_{j,k}^{[j]}, \boldsymbol{x}_{-j,k}^{[j]})$. As shown in Figure 1 in Section 3, the variance of the estimates of all players converges to zero during the iterations of the algorithm.

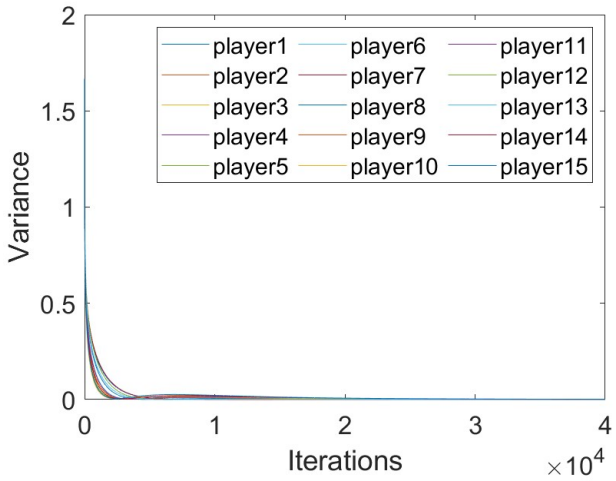

Figure 1. The figure shows the process by which all customers agree on the estimated value of GNE. The vertical coordinate is the variance, i.e.,$Var(x_i^{[1]}, ..., x_i^{[N]}), i \in \{1, 2, ...N\}$.

4. Convergence Analysis

In this section, two assumptions are proposed to analyze the convergence of Algorithms 1 and 2. First, the range of values of the step size $\overline{\alpha}, \overline{\tau}, \overline{\gamma}$ can be obtained according to Gershgorin, and then the convergence of the two algorithms can be proven based on two mild assumptions.

Assumption 5.

1. *The set of Nash equilibria of the variational problem of the original problem is nonempty*
2. *and* $\boldsymbol{R}^T\hat{\mathbb{F}} + \frac{1}{2}c\hat{\boldsymbol{L}}$ *is maximally monotone operator.*

Assumption 6.

1. *The pseudo-gradient operator* $\mathbb{F}$ *is strongly monotone and Lipschitz continuous, i.e., there exist* $\theta_1, \theta_2 > 0$, *satisfying* $\forall x, x' \in \mathcal{R}^n$, $\frac{\langle x-x', \mathbb{F}(x)-\mathbb{F}(x')\rangle}{\|x-x'\|^2} \geq \theta_1$ *and* $\frac{\|\mathbb{F}(x)-\mathbb{F}(x')\|}{\|x-x'\|} \leq \theta_2$.
2. *The operator* $\boldsymbol{R}^T\hat{\mathbb{F}}$ *is Lipschitz continuous, i.e., there exist* $\theta_3 > 0$, *satisfying* $\forall \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}' \in (R)^{nN}$, $\frac{\|\hat{\mathbb{F}}(\hat{\boldsymbol{x}})-\hat{\mathbb{F}}(\hat{\boldsymbol{x}}')\|}{\|\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}'\|} \leq \theta_2$.

Note that if and only if $\mathbf{0} \in Zer(\overline{\Phi}^{-1}\overline{\mathbb{A}} + \overline{\Phi}^{-1}\overline{\mathbb{B}})$, exists $\mathbf{0} \in Zer(\overline{\mathbb{A}} + \overline{\mathbb{B}})$. Define $\boldsymbol{P}_{\overline{\mathbb{A}}} := (I + \overline{\Phi}^{-1}\overline{\mathbb{A}})^{-1}$ and $\boldsymbol{P}_{\overline{\mathbb{B}}} := (I + \overline{\Phi}^{-1}\overline{\mathbb{B}})^{-1}$, iterative process can be expressed as $\boldsymbol{\omega_{k+1}} = \boldsymbol{\omega_k} + 2r_k(\boldsymbol{P}_{\overline{\mathbb{A}}}\boldsymbol{P}_{\overline{\mathbb{B}}} - \boldsymbol{P}_{\overline{\mathbb{B}}} - \boldsymbol{P}_{\overline{\mathbb{A}}})(\boldsymbol{\omega_k})$. Let $\boldsymbol{O}_{\overline{\mathbb{A}}} = 2\boldsymbol{P}_{\overline{\mathbb{A}}} - \boldsymbol{I}$ and $\boldsymbol{O}_{\overline{\mathbb{B}}} = 2\boldsymbol{P}_{\overline{\mathbb{B}}} - \boldsymbol{I}$, when $r_k = 0.5$, the iterative form of the Algorithm 1 can be transformed as (According to reference [20], when $r_k = 1$, it is the alternating direction method of Multipliers.)

$$\boldsymbol{\omega_{k+1}} = \frac{1}{2}\boldsymbol{\omega_k} + \frac{1}{2}\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}(\boldsymbol{\omega_k}). \tag{21}$$

When Assumptions 1–4 hold, the operators $\mathbb{A}$ and $\mathbb{B}$ are both maximally monotone. Referring to the theory of monotone operators [21], it is easy to prove the convergence of Algorithm 2. Next, we focus on the convergence of Algorithm 1.

Consider the real vector space $\mathcal{R}^{n(N+|\varepsilon|)+n(N+|\varepsilon|)}$ and define an inner product operation $\langle \boldsymbol{\omega}, \boldsymbol{\omega}' \rangle = \langle \overline{\Phi}\boldsymbol{\omega}, \boldsymbol{\omega}' \rangle$ by which an inner product space $\mathbb{Q}$ can be obtained.

Lemma 1. *Assuming that the matrix $\overline{\Phi}$ is positive definite, then $\overline{\alpha}, \overline{\tau}, \overline{\gamma}$ must satisfy the following inequalities:*

$$\forall \delta > 0, \quad \begin{cases} \overline{\alpha}_i \leq (\delta + c\hat{L}_i + \max\limits_{j \in \{1,2 \ldots n_i\}} \sum\limits_{k=1}^{m} |[-R_i^T A_i^T]_{jk}|)^{-1}, \\ \overline{\tau}_i \leq (\delta + 2d_i)^{-1}, \\ \overline{\gamma}_i \leq (\delta + \max\limits_{j \in \{1,2 \ldots m\}} \sum\limits_{k=1}^{n_i} |[-A_i R_i]_{jk}| + 2d_i)^{-1}, \end{cases} \tag{22}$$

where $d_i = \frac{1}{2}\sum_{k=1}^{m} |[L]_{ik}|$ denotes the degree of player i. This is a result of a calculation by Gershgorin which states that a sufficient condition for a positive definite matrix is that all eigenvalues are greater than zero [22].

Lemma 2. *Denoted $\widetilde{\mathbb{B}}$ by $\overline{\Phi}^{-1}\overline{\mathbb{B}}$, the operator $\widetilde{\mathbb{B}}$ is maximally monotone on $\mathbb{Q}$ when Assumptions 1–4 hold and Φ satisfies Lemma 1. Moreover, $\boldsymbol{O}_{\overline{\mathbb{B}}}$ is nonexpansive on $\mathbb{Q}$.*

Proof. According to Definition 9.12 in [23] and Theorem 20.25 in [23], the operator $\hat{\mathbb{G}}$ is maximally monotone when Assumptions 1 holds. Operator $\overline{\mathbb{B}}$ consists of a maximally monotone operator and a symmetric matrix. Therefore, operator $\overline{\mathbb{B}}$ is maximally monotone. Since the matrix fair is positive definite, by Proposition 20.24 in [23], we can conclude that $\widetilde{\mathbb{B}}$ is also maximally monotone. Moreover, by Corollary 23.11 in [23], we can know that the inverse preconditioner $\boldsymbol{O}_{\overline{\mathbb{B}}}$ is nonexpansive. □

Lemma 3. *Denoted $\widetilde{\mathbb{A}}$ by $\overline{\Phi}^{-1}\overline{\mathbb{A}}$, the operator $\widetilde{\mathbb{A}}$ is maximally monotone on $\mathbb{Q}$ when Assumptions 2–5 hold and Φ satisfies Lemma 1. Moreover, $\boldsymbol{O}_{\overline{\mathbb{A}}}$ is nonexpansive on $\mathbb{Q}$.*

Proof. The proof is similar to that of Lemma 2. When $\boldsymbol{R}^T\hat{\mathbb{F}} + \frac{1}{2}c\hat{\boldsymbol{L}}$ is a maximally monotone operator, the conclusion of the Lemma 3 can be obtained. □

Theorem 1. *Suppose λ^* satisfies (6) and that x^* is the v-GNE of Problem (3). When Lemma 3 is valid, the sequences $\{\hat{\boldsymbol{x}}_{\boldsymbol{k}}\}$ and $\{\boldsymbol{\lambda}_k\}$ generated by Algorithm 1 satisfy $\lim_{k\to\infty} \hat{\boldsymbol{x}}_{\boldsymbol{k}} = (1_N \otimes x^*)$ and $\lim_{k\to\infty} \boldsymbol{\lambda}_k = (1_N \otimes \lambda^*)$ [21].*

Proof. Since λ^* and x^* are generalized Nash equilibrium solutions, $col(x^*, z^*, \lambda^*) = (\overline{\mathbb{A}} + \overline{\mathbb{B}})col(x^*, z^*, \lambda^*)$ holds. According to Theorem 26.11 in [23], when $\overline{\mathbb{A}}$ and $\overline{\mathbb{B}}$ are maximally monotone operators, the sequence $\{\hat{\boldsymbol{x}}_{\boldsymbol{k}}\}$ and $\{\boldsymbol{\lambda}_k\}$ obtained by (21) converges to x^* and λ^*. □

Based on Assumptions 1–5, Algorithm 1 can be shown to be convergent using existing research, i.e., Theorem 1. The operator $\widetilde{\mathbb{A}}$ is restricted monotone on $\mathbb{Q}$ if Assumption 5 does not hold and Assumption 6 holds. Next, referring to the key concept of restricted monotonicity in the literature [13], it is possible to analyze the convergence of the algorithm based on Assumption 6.

Lemma 4. *Define $\sigma = s_2(L)$ and $\widetilde{\mathbb{A}}_p : col(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{\lambda}) \longrightarrow col(\boldsymbol{R}^T\hat{\mathbb{F}}(\hat{\boldsymbol{x}}) + \frac{1}{2}c\hat{\boldsymbol{L}}\hat{\boldsymbol{x}}, 0, \boldsymbol{b})$, assuming that Assumptions 2–4 and 6 hold and that the monotonicity of the operator $\widetilde{\mathbb{A}}_p$ on the inner product space $\mathbb{Q}$ is restricted when c satisfies $c \geq \frac{2}{\sigma}(\frac{(\theta_2+\theta_3)^2+4\theta_1\theta_3}{4\theta_1})$.*

Proof. $\forall\omega$ and ω^*, $\langle\omega-\omega^*,\widetilde{\mathbb{A}}_p(\omega)-\widetilde{\mathbb{A}}_p(\omega^*)\rangle = \langle\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*,\boldsymbol{R}^T(\hat{\mathbb{F}}(\hat{\boldsymbol{x}})-\hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*))\rangle+\langle\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*,\frac{1}{2}c\hat{\boldsymbol{L}}(\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*)\rangle$. Denote $\hat{\boldsymbol{x}}^x=(1_{N\times N}\bigotimes L)\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}^y=(I_{nN}-(1_{N\times N}\bigotimes L))\hat{\boldsymbol{x}}$. Since ω^* is GNE, there exists $\hat{\boldsymbol{x}}^*=\hat{\boldsymbol{x}}^{*x}$. Assuming that Assumptions 4 holds, then $\langle\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*,\frac{1}{2}c\hat{\boldsymbol{L}}(\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*)\rangle\geq\frac{c\sigma}{2}\|\hat{\boldsymbol{x}}^y\|^2$ and

$$\begin{aligned}&\langle\hat{\boldsymbol{x}}-\hat{\boldsymbol{x}}^*,\boldsymbol{R}^T(\hat{\mathbb{F}}(\hat{\boldsymbol{x}})-\hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*))\rangle\\&=\langle\hat{\boldsymbol{x}}^x+\hat{\boldsymbol{x}}^y-\hat{\boldsymbol{x}}^*,\boldsymbol{R}^T(\hat{\mathbb{F}}(\hat{\boldsymbol{x}})-\hat{\mathbb{F}}(\hat{\boldsymbol{x}}^*))\rangle\\&\geq\frac{\theta_1}{N}\|\hat{\boldsymbol{x}}^x-\hat{\boldsymbol{x}}^y\|^2-\frac{\theta_2+\theta_3}{N^{0.5}}\|\hat{\boldsymbol{x}}^y\|\|\hat{\boldsymbol{x}}^x-\hat{\boldsymbol{x}}^*\|-\theta_3\|\hat{\boldsymbol{x}}^y\|^2.\end{aligned}$$

Combining the above two inequalities, when $c\geq\frac{2}{\sigma}(\frac{(\theta_2+\theta_3)^2+4\theta_1\theta_3}{4\theta_1})$ holds, $\widetilde{\mathbb{A}}_p$ is restricted monotone on $\mathbb{Q}$. □

Lemma 5. *Assume that Lemma 4 holds and that the monotonicity of the operator $\widetilde{\mathbb{A}}$ on the inner product space $\mathbb{Q}$ is restricted. In addition, $\boldsymbol{O}_{\overline{\mathbb{A}}}$ is restricted non-expansive on $\mathbb{Q}$.*

Proof. Assume that $\widetilde{\mathbb{A}}_p$ is a restricted monotone operator. By a similar process as in Lemma 2, Lemma 5 can also be proved. □

Remark 3. *The operator $\widetilde{\mathbb{B}}$ is still maximally monotonic since Assumption 1 holds. Furthermore, $\boldsymbol{P}_{\overline{\mathbb{A}}}$ is still single-valued and well-defined, even though $\widetilde{\mathbb{A}}$ is a restricted monotone operator.*

Theorem 2. *Suppose λ^* satisfies (6) and x^* is the v-GNE of Problem (3). When Lemma 5 is valid. The sequences $\{\hat{\boldsymbol{x}}_{\boldsymbol{k}}\}$ and $\{\boldsymbol{\lambda}_k\}$ generated by Algorithm 1 satisfy $\lim_{k\to\infty}\hat{\boldsymbol{x}}_{\boldsymbol{k}}=(1_N\otimes x^*)$ and $\lim_{k\to\infty}\boldsymbol{\lambda}_k=(1_N\otimes\lambda^*)$.*

Proof. According to (21), $\forall\hat{\boldsymbol{\omega}}^*\in Fix(\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}})$, we have

$$\begin{aligned}&\|\hat{\boldsymbol{\omega}}^{\boldsymbol{k+1}}-\hat{\boldsymbol{\omega}}^*\|^2\\&=\frac{1}{2}\|\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^*\|^2+\frac{1}{2}\|\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^*\|^2-\frac{1}{4}\|\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}\|^2\\&\leq\frac{1}{2}\|\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^*\|^2-\frac{1}{4}\|\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}\|^2.\end{aligned}$$

Since $\overline{\mathbb{B}}$ is a maximal monotone operator and $\overline{\mathbb{A}}$ is a restricted monotone operator, the sequences $\{\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}\}$ is Fejer monotone. Note that $\sum_{k=1}^{\infty}\frac{1}{4}\|\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}\|^2\leq\|\hat{\boldsymbol{\omega}}^{\boldsymbol{0}}-\hat{\boldsymbol{\omega}}^*\|^2$, so $\lim_{k\longrightarrow\infty}\|\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}\|^2=0$.

According to the above conclusion, it can be known that there exists a subsequence $\{\hat{\boldsymbol{\omega}}^{\boldsymbol{k_i}}\}$ Based on the previous result, the subsequence, i.e., $\lim_{i\longrightarrow\infty}\hat{\boldsymbol{\omega}}^{\boldsymbol{k_i}}=\hat{\boldsymbol{\omega}}^{\boldsymbol{s}}$. As $\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}}$ is a continuous mapping, $\hat{\boldsymbol{\omega}}^{\boldsymbol{s}}$ is a fixed point Fix$(\boldsymbol{O}_{\overline{\mathbb{B}}}\boldsymbol{O}_{\overline{\mathbb{A}}})$. This implies that $\lim_{k\longrightarrow\infty}(\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}=\hat{\boldsymbol{\omega}}^{\boldsymbol{s}})$. It follows from the above inequality that $\{\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}-\hat{\boldsymbol{\omega}}^{\boldsymbol{s}}\}$ is a monotonically non-increasing sequence. In summary, the sequence $\hat{\boldsymbol{\omega}}^{\boldsymbol{k}}$ can converge to the fixed point $\hat{\boldsymbol{\omega}}^{\boldsymbol{s}}$, and $\hat{\boldsymbol{\omega}}^{\boldsymbol{s}}$ is a zero point of the operator $\mathbb{T}$. □

In summary, in this section, the convergence of Algorithms 1 and 2 based on two parallel assumptions is analyzed, and the interval for the value of step size is given.

5. Algorithm Simulation

In this section, we verify the functionality of the two algorithms proposed in this paper and compare their convergence with other algorithms based on a non-cooperative game competition network. Referring to the networked course game [24], consider a LAN with 15 clients (i.e., $N=15$) and four routers (i.e., $m=4$), where each client can choose one router to connect, and clients i can pay x_i to the Internet operator to get more network bandwidth. Due to the cost, the client's sentiment and interest are affected by the payment amount x_i, so each client has a private cost function $g_i(x_i):\mathcal{R}^{n_i}\to\mathcal{R}$. Since the bandwidth

that each router can carry is finite, each client has a cost function $f_i(x_i, x_{-i})$ that depends on the decisions of all clients.

Using the matrix A, we can characterize the routers chosen by each client individually, and let $x = \mathrm{col}(x_1, x_2 ... x_n)$, then Ax is the total supply vector to all routers given the action profile x of all clients. A constraint $Ax < b$ is used to denote the bandwidth limit of these routers, where $b = \mathrm{col}(b_1, b_2, ..., b_m) \in \mathcal{R}^m$. In addition, each client is restricted to exchange decisions only with its neighboring clients, and the adjacency relationship between clients is represented by a Laplace matrix. Then, the goal of each player in this model is to solve (3) in a situation where they can only communicate with adjacent clients.

To satisfy Assumptions 1 and 2, we define function $g_i(x_i) = \eta_i(\sum_{j=1}^{n_i}[x_i]_j)^2 + \xi_i^T$, where $x_i, \eta_i, \xi_i \in \mathcal{R}$. $g_i(x_i)$ is a strongly convex, Lipschitz continuous function. Define $F : \mathcal{R}^m \to \mathcal{R}^m$ to map the amount paid by users connected to the router and the bandwidth they can allocate, then $f_i(x_i, x_{-i}) = -F^T(Ax)A_i x_i$. Reference to economics [25], using a linear inverse demand function $F_j(x) = \overline{F_j} - d_j[Ax]_j$ as the mapping F. Denote gradient vector $F(x) = \mathrm{col}(F_1(x), ..., F_m(x))$, $\overline{F} = \mathrm{col}(\overline{F_1}, \overline{F_2}, ..., \overline{F_m})$, $B = \mathrm{diag}\{d_1, ..., d_m\}$. Then $F(x) = \overline{F} - BAx$ is the unit return vector function and the player's cost function can be written as $g_i(x_i) + \sum_{j=1}^{N} x_j^T A_j^T B A_i x_i - \overline{F}^T A_i x_i$.

The experimental setup is described next, with 15 customers, i.e., $N = 15$, 4 routers, i.e., $m = 4$, and the total bandwidth that the 4 routers can carry being 9, 6, 12, and 8, i.e., $b = \mathrm{col}(9, 6, 12, 8)$. Clients 1, 3, 5, and 7 are connected to the first router; clients 2, 4, 6, and 8 are connected to the second router; clients 9, 10, and 13 are connected to the third router; and the remaining clients are connected to the fourth router, as shown in Figure 2.

As shown in Figure 3, Algorithms 1 and 2 and the DA algorithm based on the Doubly-Augmented operator splitting approach [13] all have good convergence results.

Next, the cost functions of all participants are accumulated and gradually decrease with algorithm iterations. As shown in Figure 4, it can be found that all three algorithms mentioned above minimize the cost function, and in comparison, using Algorithm 1 leads to the fastest decrease in the cost function.

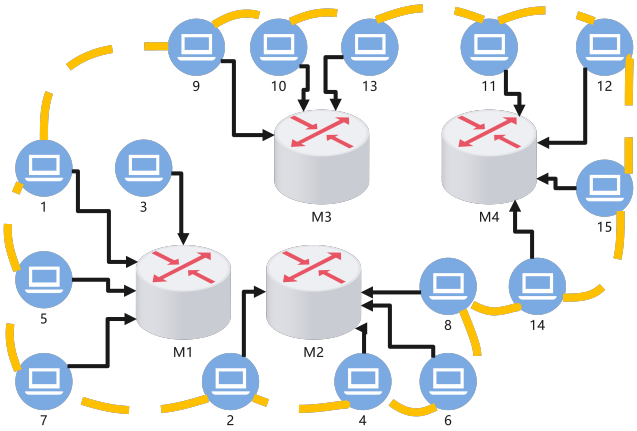

Figure 2. The black arrows start at the client end and end at the router chosen by that client, and the clients at either end of the yellow arc are neighbors.

In order to clearly show how the algorithm works, the decreasing cost function of the 15 participants is shown in Figure 5, and it can be found that the cost function of each participant converges to the minimum value continuously and smoothly.

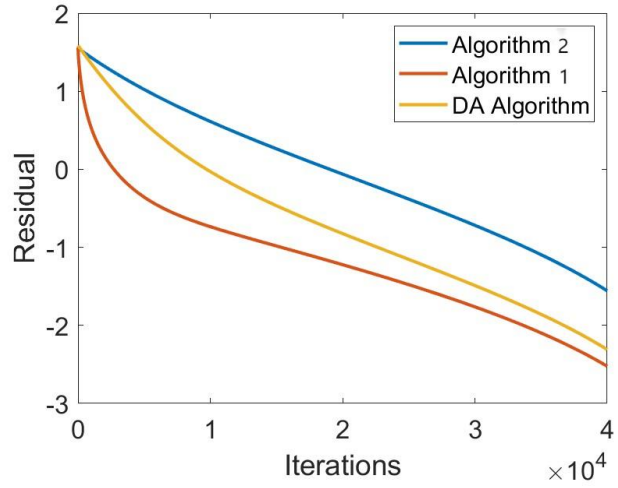

Figure 3. The blue, red, and yellow curves indicate the convergence process of Algorithms 1 and 2, and DA algorithm, respectively.

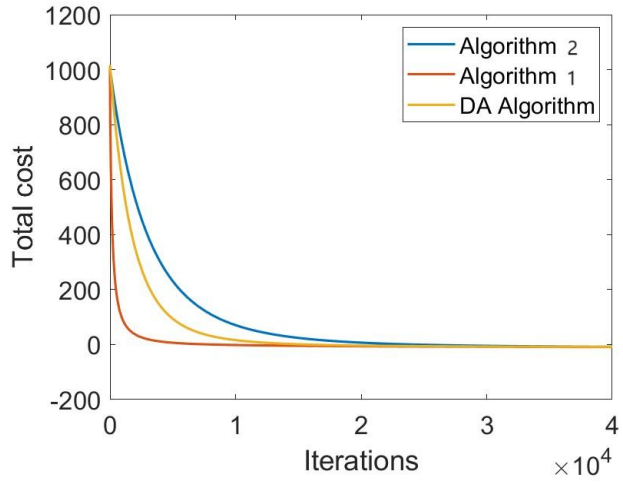

Figure 4. The figure shows how the sum of the costs of all customers changes under the action of the three algorithms. The ordinate is $\sum_{i=1}^{i=N}[f_i(x_i, x_{-i}) + g_i(x_i)]$.

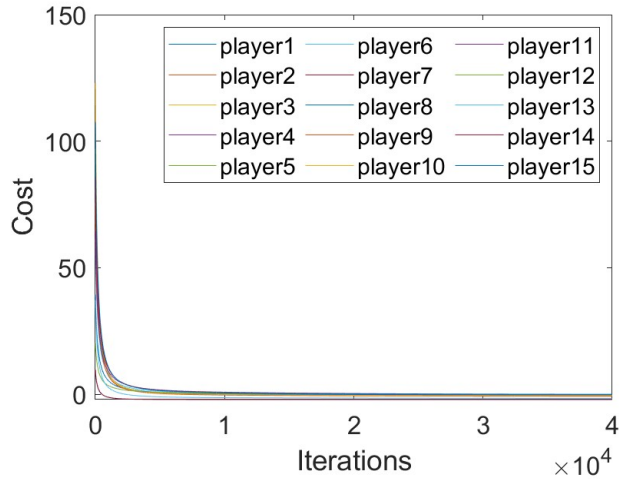

Figure 5. The figure shows how the private cost per customer changes under the action of Algorithm 1. The ordinate is $f_i(x_i, x_{-i}) + g_i(x_i), i \in \{1, ...N\}$.

At the end of this section, the process of reaching agreement on the estimated variables $\hat{x}$ of the global decision set x by the participants $i \in \mathcal{N}$ during the iteration of the algorithm is shown in Figure 1. As mentioned in this paper, the algorithm under local information elevates the decision vector latitude of the participants, and each participant has not only their own local decision $x_i = x_i^{[i]} \in \mathcal{R}$, but also an estimate of the other participants' decisions $\boldsymbol{x}_{-i}^{[i]} = \text{col}(..., x_{i-1}^{[i]}, x_{i+1}^{[i]}, ..., x_N^{[i]})$, where $x_j^{[i]}$ denotes the estimate of participant i on participant j. Relatively, for each participant i, there exist N estimates of him, and we define this set as $\mathbf{E}$, i.e., $\mathbf{E} = \{x_i^{[1]}, ..., x_i^{[N]}\}$, let k be the number of iterations, if $\forall i \in \{1, ..., N\}, \lim_{k\to\infty} Var(\mathbf{E}) = 0$, then it is proved that all decision makers reach agreement on the estimate of the global decision set, which is equivalent to knowing the global decision set.

6. Summary and Prospect

In this paper, we study algorithms for finding the GNE of noncooperative game problems for two different situations. In the case of global decision information, each participant can use the global decision set to minimize the cost function. In the case of partial decision information, participants need to estimate the global decision set and perform local optimization by estimating variables, while consensus dynamics with the Laplacian matrix ensures that the local estimations will reach the same GNE. This paper devotes an entire section to the derivation of Algorithms 1 and 2. In addition, we derive the step size condition for the convergence of the algorithm using the Gershgorin Circle Theorem and analyze the convergence of the algorithm under mild assumptions. Finally, we simulated both algorithms and compared them with the FB algorithm. This paper assumes that the objective function of each player is continuously differentiable, but in practical problems, there may be nonsmooth or nonconvex situations. Therefore, how to generalize the algorithm proposed in this paper to more general cases is a worthwhile research question. This paper only considers undirected connected graphs as the communication topology among players, but in real networks, there may be directed graphs or disconnected graphs. Therefore, how to analyze and design distributed algorithms that apply to these complex topological structures is a challenging problem. This paper only discusses the algorithm based on a synchronous update strategy, but in real networks, there may be asynchronous or random updates. Therefore, how to consider the impact of these update strategies on the performance and stability of the algorithm is a meaningful problem.

Author Contributions: Methodology, H.L. and Y.S.; Software, J.C., M.C. and Z.W.; Formal analysis, J.C., H.L. and M.C.; Investigation, J.C. and Y.S.; Resources, J.C. and Y.S.; Data curation, J.C., H.L. and J.T.; Writing—original draft, J.C.; Supervision, Y.S.; Project administration, J.C. 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: No new data were created.

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

References

1. Facchinei, F.; Kanzow, C. Generalized Nash equilibrium problems. *4OR* **2007**, *5*, 173–210. [CrossRef]
2. Fischer, A.; Herrich M.; Schonefeld, K. Generalized Nash equilibrium problems-recent advances and challenges. *Pesqui. Oper.* **2014**, *34*, 521–558. [CrossRef]
3. Garcia, A.; Khatami, R.; Eksin, C.; Sezer, F. An Incentive Compatible Iterative Mechanism for Coupling Electricity Markets. *IEEE Trans. Power Syst.* **2021**, *37*, 1241–1252. [CrossRef]
4. Cadre, H.L.; Jacquot, P.; Wan, C.; Alasseur, C. Peer-to-Peer Electricity Market Analysis: From Variational to Generalized Nash Equilibrium. *Eur. J. Oper. Res.* **2018**, *282*, 753–771. [CrossRef]

5. Wu, C.; Gu, W.; Bo, R.; MehdipourPicha, H.; Jiang, P.; Wu, Z.; Lu, S.; Yao, S. Energy Trading and Generalized Nash Equilibrium in Combined Heat and Power Market. *IEEE Trans. Power Syst.* **2020**, *35*, 3378–3387. [CrossRef]
6. Debreu, G. A Social Equilibrium Existence Theorem. *Proc. Natl. Acad. Sci. USA* **1952**, *38*, 886–893. [CrossRef] [PubMed]
7. Rosen, J.B. Existence and Uniqueness of Equilibrium Points for Concave N-Person Games. *Econometrica* **1965**, *33*, 520–534. [CrossRef]
8. Facchinei, F.; Kanzow, C. Generalized Nash Equilibrium Problems. *Ann. Oper. Res.* **2010**, *175*, 177–211. [CrossRef]
9. Facchinei, F.; Pang, J.S. Nash equilibria: The Variational Approach. In *Convex Optimization in Signal Processing and Communications*; Cambridge University Press: Cambridge, UK, 2009.
10. Yi, P.; Pavel, L. An Operator Splitting Approach for Distributed Generalized Nash Equilibria Computation. *Automatica* **2019**, *102*, 111–121. [CrossRef]
11. Latafat, P.; Freris, N.M.; Patrinos, P. A New Randomized Block-Coordinate Primal-Dual Proximal Algorithm for Distributed Optimization. *IEEE Trans. Autom. Control.* **2019**, *64*, 4050–4065. [CrossRef]
12. Malitsky, Y. Golden Ratio Algorithms for Variational Inequalities. *Math. Program.* **2020**, *184*, 383–410. [CrossRef]
13. Pavel, L. Distributed GNE Seeking Under Partial-Decision Information Over Networks via a Doubly-Augmented Operator Splitting Approach. *IEEE Trans. Autom. Control.* **2020**, *65*, 1584–1597. [CrossRef]
14. Bianchi, M.; Belgioioso, G.; Grammatico, S. Fast generalized Nash equilibrium seeking under partial-decision information. *Automatica* **2022**, *136*, 110080. [CrossRef]
15. Cai, X.; Xiao, F.; Wei, B. Distributed generalized Nash equilibrium seeking for noncooperative games with unknown cost functions. *Int. J. Robust Nonlinear Control.* **2022**, *32*, 8948–8964. [CrossRef]
16. Aragón Artacho, F.J.; Campoy, R. Solving Graph Coloring Problems with the Douglas-Rachford Algorithm. *Set-Valued Var. Anal.* **2018**, *26*, 277–304. [CrossRef]
17. Lakshmi, R.J.; Rao, G.V.S. A Re-Constructive Algorithm to Improve Image Recovery in Compressed Sensing. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 5443–5453.
18. Steidl, G.; Teuber, T. Removing Multiplicative Noise by Douglas-Rachford Splitting Methods. *J. Math. Imaging Vis.* **2010**, *36*, 168–184. [CrossRef]
19. Egerestedt, M. *Graph-Theoretic Methods for Multi-Agent Coordination*; Georgia Institute of Technology: Atlanta, GA, USA, 2007.
20. Giselsson, P.; Boyd, S. Linear Convergence and Metric Selection for Douglas-Rachford Splitting and ADMM. *IEEE Trans. Autom. Control.* **2014**, *62*, 532–544. [CrossRef]
21. Combettes, P.L. Monotone operator theory in convex optimization. *Math. Program.* **2018**, *170*, 177–206. [CrossRef]
22. Bell, H.E. Gershgorin's Theorem and the Zeros of Polynomials. *Am. Math. Mon.* **1965**, *72*, 292–295. [CrossRef]
23. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed.; Springer Publishing Company: Berlin/Heidelberg, Germany, 2017.
24. Peng, Y.; Pavel, L. Asynchronous Distributed Algorithms for Seeking Generalized Nash Equilibria Under Full and Partial-Decision Information. *IEEE Trans. Cybern.* **2020**, *50*, 2514–2526.
25. Moschini, G.; Vissa, A. A Linear Inverse Demand System. *J. Agric. Resour. Econ.* **1992**, *7*, 294–302.