Generalized Multi-cluster Game under Partial-decision Information with Applications to Management of Energy Internet

Yue Chen and Peng Yi

Abstract—The decision making and management of many engineering networks involves multiple parties with conflicting interests, while each party is constituted with multiple agents. Such problems can be cast as a multi-cluster game. Each cluster is treated as a self-interested player in a non-cooperative game where agents in the same cluster cooperate together to optimize the payoff function of the cluster. In a large-scale network, the information of agents in a cluster can not be available immediately for agents beyond this cluster, which raise challenges to the existing Nash equilibrium seeking algorithms. Hence, we consider a partial-decision information scenario in general Nash equilibrium seeking for multi-cluster games in a distributed manner. We reformulate the problem as finding zeros of the sum of preconditioned monotone operators by the primal-dual analysis and graph Laplacian matrix. Then a distributed generalized Nash equilibrium seeking algorithm is proposed without requiring fully awareness of its opponent clusters’ decisions based on a forward-backward-forward method. With the algorithm, each agent estimates the strategies of all the other agents by communicating with neighbors via an undirected network. We show that the derived operators can be monotone when the communication strength parameter is sufficiently large. We prove the algorithm convergence resorting to the fixed point theory by providing a sufficient condition. We discuss its potential application in Energy Internet with numerical studies.

I. INTRODUCTION

NASH game plays an important role in engineering network systems with multiple agents, where each agent aims to minimize its own payoff function interfered by other player’s decisions. It arises in a broad variety of applications, such as transportation network [1], wireless network [2], construction management [3], mobile sensor network [4]. In this paper, we consider a game composed of multiple clusters (or coalitions [5], [6]) while each cluster is treated as a non-cooperative player in the game, and the agents within the same cluster work collaboratively to minimize the cluster’s payoff function. This formulation is discussed in [5], [7], and Nash equilibrium (NE) seeking algorithms are designed. It is an extension of the zero-sum games of two networks in [8] and [9] by allowing more clusters. The multi-cluster game and its NE seeking have received considerable attention, due to its wide applications in large-scale hierarchical networks, including healthcare networks [10], task allocation networks [11], electricity market [12].

In some engineering problems, both the payoff function and the feasible action set of each player can be coupled with the other’s actions, and the game is called generalized Nash game [13], [14], [15], [16]. Applications of such games are ubiquitous in the engineering field, including smart grids [17], optical network [18], mobile cloud computing [19], charging control of plug-in electric vehicle [20] and management of energy hubs [21]. Generalized Nash equilibrium (GNE) is an solution to generalized Nash game, and its seeking algorithms have been studied in [18], [22], [23], [24]. GNE seeking for multi-cluster game are also studied, such as [6], [12].

The classical NE seeking studies usually assume that all agents can observe the full information of all their opponents’ decisions. In this setting, a central coordinator or an interference network is used to communicate with all the agents [25], [14], while the NE computing method is usually based on best-response [22], or gradient schemes [26]. However, in recent years, there has been an increasing interest to consider partial-decision information regimes that agents are unable to fully observe opponents’ decisions. In partial information regime, many studies consider consensus-based gradient type approach of NE seeking, e.g. [27], [28] and [29] all use the gradient tracking play to calculate NE in multi-cluster games, and [30] designs a distributed NE seeking algorithm by using an inexact-ADMM approach. All the aforementioned literatures consider games (including multi-cluster games) without shared constraints, i.e., non-generalized games. As far as we known, there are little literatures discussing generalized games among multiple clusters except [12], even fewer in partial information regime of multi-cluster games.

In this paper we consider distributed GNE computation in multi-cluster games with shared affine coupling constraints under partial information regime. Solving the generalized Nash equilibrium problems is a challenging task, and one useful way is to find an important class of GNEs called variational GNE (v-GNE) [22], [23], [24], which corresponds to the solution of a variational inequality (VI).

In recent years, operator-theoretic approach gets appealing for solving (generalized) Nash games, since it is regarded being mathematically general and elegant. The GNE solving problems are reformulated into finding zeros of monotone operators, known as monotone inclusion problems [32], [33] proposed a popular forward-backward (FB) operator splitting method by using preconditioned matrix and incorporating the
The contributions of this paper can be summarized as follows: in FB method, it is formulated as a GNE seeking of multi-cluster game. Section [IV] contains two subsections. The subsection [IV-A] introduces the development of the distributed algorithm based on operator splitting method, and subsection [IV-B] shows the convergence analysis. Applications to EI is discussed in Section [V] and concluding remarks are given in Section [VI]. The proofs are given in Appendix [A].

Notations. Denote $\mathbb{R}^m(\mathbb{R}_m^m)$ as the $m$-dimensional (non-negative) Euclidean space. For an integer $m > 0$, denote $N_m^* := \{1,2,\ldots, m\}$. And $N_m^* \setminus j := \{1,2,\ldots, j-1, j+1, \ldots, m-1, m\}$, which contains all elements of set $N_m^*$ except $j$. $\langle x,y \rangle = x^T y$ denotes the inner product of $x$ and $y$, and $\|x\|_2 = \sqrt{x^T x}$ denotes the norm induced by inner product $\langle \cdot, \cdot \rangle$. $x \otimes y$ denotes their Kronecker product. Given a symmetric positive definite matrix $\Psi$, denoted as $\Psi > 0$, denote the $\Psi$-induced inner product $\langle x, y \rangle_{\Psi} = \langle \Psi x, y \rangle$. The $\Psi$-induced norm, $\| \cdot \|_{\Psi}$, is defined as $\| x \|_{\Psi} = \sqrt{\langle \Psi x, x \rangle}$. Denote by $\| \cdot \|$ any matrix induced norm in the Euclidean space. For $A \in \mathbb{R}^{m \times m}$, let $\|A\| = \delta_{\max}(A)$ denote the 2-induced matrix norm, where $\delta_{\max}(A)$ is its maximum singular value. If $A$ is a square matrix, denote $s_{\max}(A)$ and $s_{\min}(A)$ as the maximal eigenvalue and minimum eigenvalue of $A$, respectively. Denote $I_m = (1, \ldots, 1)^T \in \mathbb{R}^m$ and $0_m = (0, \ldots, 0)^T \in \mathbb{R}^m$. Denote $diag(A^j_{j \in N_m^*})$ as the block diagonal matrix with $A^1, \ldots, A^m$ on its main diagonal. Null($A$) and Range($A$) denote the null space and range space of matrix $A$, respectively. Denote $col((x^j_{i})_{j \in N_m^*, i \in N_j^*}) := [x^1, x^{n_1}_{1}, \ldots, x^1, x^{n_1}_{1}, \ldots, x^m, x^{n_m}_{m}]^T$ as the stacked column vector. $I_m$ denotes the identity matrix in $\mathbb{R}^{m \times m}$. Denote $N_j = \{j \in \mathbb{N} [0, m - 1] \}$ as the Cartesian product of the sets $\Omega_j$. Define the indicator function of $\Omega_j$ as $\chi_{\Omega_j}(x) = 1$ if $x \in \Omega_j$ and $\chi_{\Omega_j}(x) = 0$ if $x \notin \Omega_j$.

Let $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes a set-valued operator [32]. The domain of $\mathcal{A}$ is $dom(\mathcal{A}) = \{x \in \mathbb{R}^n | Ax \neq \emptyset \}$ where $\emptyset$ stands for the empty set. The zero set of $\mathcal{A}$ is $zer(\mathcal{A}) = \{x \in \mathbb{R}^n | 0 \in Ax\}$. The resolvent of the operator $\mathcal{A}$ is $J_{\mathcal{A}} = (Id + \Gamma \mathcal{A})^{-1}$, where $Id$ denotes the identity operator. An operator $\mathcal{A}$ is monotone if $(Ax - By, x - y) \geq 0$, and it is maximally monotone if its graph is not strictly contained in the graph of any other monotone operator, where the graph of $\mathcal{A}$ is defined as $gra(\mathcal{A}) = \{(x, u) | u \in Ax\}$. Given a proper, lower semi-continuous, and convex function $h$, the sub-differential is the operator $\partial h : dom(h) \rightarrow 2^{\mathbb{R}^n}$ is $x \mapsto \{u \in \mathbb{R}^n | h(y) \geq h(x) + \langle u, y-x \rangle, \forall y \in dom(h)\}$. The proximal operator is defined as $Prox_{h}(v) := \arg \min_{u \in dom(h)} \{h(u) + 1/2 \|u - v\|^2 \} = J_{\partial h}(v)$, and if $\Psi > 0$, then $Prox_{h}(v) = J_{\Psi \partial h}(v)$. And $N_{\Omega} : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ is a set-valued mapping which denotes the norm cone operator for the set $\Omega$, i.e., if $x \in \Omega$, then $N_{\Omega} = \{v \in \mathbb{R}^n | \sup_{u \in \Omega} \langle v^T (u - x) \rangle \leq 0 \}$, otherwise $N_{\Omega}(x) = \emptyset$.

II. GAME FORMULATION

In this section, we introduce the formulation of multi-cluster games. The full-information and partial-information regimes are introduced in subsection [II-A] and [II-B] respectively.
A. Multi-cluster Games with Full-decision Information

Consider a multi-cluster game consists of $m$ clusters where each cluster is treated as a player in a non-cooperative game. Every cluster contains a number of agents, i.e., $n_j$ denotes the number of agents in cluster $j$ where $j \in \mathbb{N}^*_m$. Denote $x^j_i \in \mathbb{R}^{q_j}$ the decision of agent $i$ in cluster $j$, and $x^j = [x^j_1, \ldots, x^j_{n_j}]$ a stacked strategy vector of cluster $j$ piled up by $x^j_i$, the same as $\mathbf{x} \in \mathbb{R}^q$ piled up by $x^j$. The number of agents in the game is $n = \sum_{j=1}^m n_j$, and the dimensions of $\mathbf{x}$ is the sum of all clusters: $q = \sum_{j=1}^m n_j q_j$.

Each cluster’s payoff function is the sum of all agent’s payoff functions in that cluster, i.e., cluster $j$’s payoff function is given by

$$\theta^j(x^j, \mathbf{x}^{-j}) = \sum_{i=1}^{n_j} \theta^j_i(x^j_i, \mathbf{x}^{-j}) : \mathbb{R}^{q_j} \rightarrow \mathbb{R},$$

where $\mathbf{x}^{-j} = \text{col}((x^j_i)_{i \in \mathbb{N}^*_m \setminus \{j\}})$ denotes a stacked strategy vector piled up by all agents’ strategies except agents in cluster $j$, and $\theta^j_i : \mathbb{R}^{q_j} \rightarrow (-\infty, \infty]$ denotes an extended-valued local payoff function of agent $i$ in cluster $j$. We usually consider $\theta^j_i$ can be split into smooth and non-smooth parts, i.e.,

$$\theta^j_i(x^j_i, \mathbf{x}^{-j}) = f^j_i(x^j_i, \mathbf{x}^{-j}) + h^j_i(x^j_i).$$

**Assumption 1:** For each $j \in \mathbb{N}^*_m$, $i \in \mathbb{N}^*_n$, and for fixed $x^{-j} \in \mathbb{R}^{q-n_j q}$, the function $f^j_i(x^j, x^{-j})$ in (1) is convex and continuously differentiable, $h^j_i : \mathbb{R}^{q_j} \rightarrow (-\infty, \infty]$ is lower semi-continuous and convex, and $\text{dom}(h^j_i) = \Omega^j_i$ is a non-empty compact and convex set.

We assume all agents in the same cluster are connected by a graph $G^j$ where $j \in \mathbb{N}^*_m$. Moreover, in order to deliver information between clusters, we assume that all leaders are connected by a graph $G_L$, as shown in Fig.1. Then we make the following assumption.

**Assumption 3:** For all $j \in \mathbb{N}^*_m$, $G^j$ and $G_L$ are undirected and connected.

We denote $G^j = (N^j, E^j)$ is an undirected graph of cluster $j$, $N^j$ is the set of agents, $E^j \subseteq N^j \times N^j$ is the edge set, and $\mathbf{W}^j = [w^j_{i,j}] \in \mathbb{R}^{n_j \times n_j}$ is the adjacency matrix such that $w^j_{i,j} = w^j_{j,i} > 0$ if $i, j \in E$ and $w^j_{i,j} = 0$ otherwise. Let $d^j_i = \sum_{j=1}^{n_j} w^j_{i,j}$, then $\text{Deg}^j = \text{diag}((d^j_i)_{i \in N^*_j}) \in \mathbb{R}^{n_j \times n_j}$. The weighted Laplacian of $G^j$ is defined as $L^j = \text{Deg}^j - \mathbf{W}^j$. If graph $G^j$ is undirected and connected, then $0$ is a simple eigenvalue of $L^j$, $L^j 1_{n_j} = 0_{n_j}$, $1^T L^j 1 = 0^T$, and all other eigenvalues are positive. The eigenvalues of $L^j$ in ascending order is given by $0 = s_1(L^j) < s_2(L^j) \leq \cdots \leq s_{n_j}(L^j)$. And we denote $L^j = \text{diag}((L^j_{i,j})_{i,j \in N^*_j})$.

The undirected and connected graph $G^j$ is defined similarly. Denote its weighted matrix as $\mathbf{W}_L = [w^j_{1,j}] \in \mathbb{R}^{m \times m}$, and Laplacian matrix as $L^L_0$.

![Fig.1. The communication topology of the multi-cluster game.](image)

We recast the condition $x^j_i = x^j_i$ in (2) as $L^j x^j = 0$ [12]. Then we define a local Lagrangian function for cluster $j$ with multiplier $\lambda^j \in \mathbb{R}^{n_j q_j}$, $\mu^j \in \mathbb{R}^w$ as

$$L(x^j, \lambda^j, \mu^j ; x^{-j}) = \theta^j(x^j, x^{-j}) + (\lambda^j)^T L^j x^j + (\mu^j)^T (A x - b),$$

where $L^j = L^j \otimes I_{q_j}$, $b = \sum_{j=1}^{m} b^j$, and

$$A = [A^1, 0_{w \times (n_q_1 - q_1)}, \ldots, A^m, 0_{w \times (n_q_m - q_m)}] \in \mathbb{R}^{w \times q}.$$

Note that $A x = \sum_{j=1}^{m} A^j x^j = \sum_{j=1}^{m} A^j x^j$, where $A^j = [A^j, 0_{w \times (n_q_j - q_j)}] \in \mathbb{R}^{w \times n_j q_j}$. 

We assume $\mathcal{K}$ is non-empty and satisfies Slater’s constraint qualification.

**Assumption 2:** $\mathcal{K}$ is non-empty and satisfies Slater’s constraint qualification.

For all $j \in \mathbb{N}^*_m$, $G^j$ and $G_L$ are undirected and connected.
When $x^*$ is an GNE of game (3), there exist $\lambda^x \in \mathbb{R}^{n_{ij}}$ and $\mu^x \in \mathbb{R}^{w}$ such that the following Karush Kuhn Tucker (KKT) conditions are satisfied:

$$
\begin{align*}
0_{n_{ij}} & \in \nabla x_i f_j(x^+, x^{-j}) + \partial x_i h_j(x^*) + L^j \lambda^x + (A^j)^T \mu^x, \\
0_w & \in b - Ax^* + N_{x^*,R} \mu^x, \\
0_{n_{ij}} & = L^j x^{-j}, \\
& \text{where } \nabla x_i f_j(x^+, x^{-j}) = \sum_{i=1}^{n_{ij}} \nabla x_i f_i^j(x^i, x^{-i}), \text{ and } \partial x_i h_j(x^*) = \sum_{i=1}^{n_{ij}} \partial x_i h_i^j(x^i).
\end{align*}
$$

(6)

In this work, we seek GNE with the same Lagrangian, which is a subclass of GNE, and $x^* \in K$ is a solution of the variational inequality GVI($\Theta, \mathcal{K}$):

$$
\exists \theta_{x^*} \in \Theta(x^*) (\theta_{x^*}, x - x^*) \geq 0, \quad \forall x \in \mathcal{K},
$$

(7)

where $\theta(x^*) = F(x^*) + H(x^*)$, and $F$ is a mapping of the game which is given by

$$
F(x) = \text{col}(\nabla x_i f_i^j(x^i, x^{-i}))_{j \in N_{x^*}},
$$

and $H(x) = \partial x_i h_i^j(x^i) \times \cdots \times \partial x_m h_m(x^m)$.

Before discussing the existence and uniqueness of a v-GNE, we make the following assumption.

**Assumption 4:** $F$ is strongly monotone and Lipschitz continuous: for any $x, x'$, there existed $\eta > 0$ such that $\langle F(x) - F(x'), x - x' \rangle \geq \eta \|x - x'\|^2$, and there exist $\kappa_0 > 0$ such that $\|F(x) - F(x')\| \leq \kappa_0 \|x - x'\|$.

Let Assumption [1][2][4] be satisfied. Note that $H$ is maximally monotone since it is constituted by sub-differential of lower semi-continuous convex function $h_i^j$, for all $j \in N_{x^*}$.

**Assumption 5:** The solution set of GVI($\Theta, \mathcal{K}$) is nonempty.

More sufficient conditions for the existence of solutions to monotone GVI can be found in [46], the compactness of $\Omega, j \in N_{x^*}$ also ensures the Assumption 5.

If $x^*$ is the solution of GVI($\Theta, \mathcal{K}$), then there exist $\lambda^x \in \mathbb{R}^q$ and $\mu^x \in \mathbb{R}^w$ such that the following KKT conditions are satisfied:

$$
\begin{align*}
0_q & \in F(x^*) + H(x^*) + L \lambda^x + A^T \mu^x, \\
0_w & \in b - Ax^* + N_{x^*,R} \mu^x, \\
0_{n_{ij}} & = L^j x^{-j}, \\
& \text{where } L = \text{diag}(L^j)_{j \in N_{x^*}}, \quad \lambda = \text{col}(\lambda^j)_{j \in N_{x^*}}.
\end{align*}
$$

(9)

**Lemma 1:** Let Assumptions [1][2][4] hold. Then $x^* \in \mathbb{R}^q$ is a v-GNE of the game (3) if and only if there exists $\lambda^x \in \mathbb{R}^q$ and $\mu^x \in \mathbb{R}^w$ such that $(x^*, \lambda^x, \mu^x)$ satisfies the KKT conditions in (9).

**Proof:** Assumptions [1][2][4] guarantees the existence of a solution to GVI($\Theta, \mathcal{K}$), by (Prop.12.11 [46], Prop.2.3 [47]). For any $(x^*, \lambda^x, \mu^x)$ satisfies the KKT conditions in (9), the same $x^*$ is the solution of GVI($\Theta, \mathcal{K}$) by (Prop.3.3 [48], Th.3.1.1[49]). And every solution $x^*$ of GVI($\Theta, \mathcal{K}$) is a GNE of game (3) by (Prop.12.4 [46], Th.3.9 [23]). Then the same $x^*$ in (9) is a v-GNE of game (3).

**B. Multi-cluster Games with Partial-decision Information**

In the following, we consider a more realistic scenario that the agents cannot have full decision information on others outside the cluster $x^{-j}$, called partial-decision information scenario. Under this setting, agents have to estimate the strategies of all other agents outside cluster by relying on the information exchanged with some neighbors over communication networks $G^j, \forall j \in N_{x^*}$ and $G_L$.

In cluster $j$, $\hat{x}^j_{i'} \in \mathbb{R}^{q_{i'}}$ denotes the estimation of agent $i$ to agent $i'$ in cluster $j$. Then the estimation vector of agent $i$ in cluster $j$ to all other agents is given by $\hat{x}^j = \text{col}(\hat{x}^j_{i'_{j}}, j' \in N_{x^*}, i' \in N_{x^*}) \in \mathbb{R}^q$.

Moreover, the estimation vector of agent $i$ in cluster $j$ to all agent’s except cluster $j$ denotes as $\hat{x}^j_{i \sim j} = \text{col}(\hat{x}^j_{i'_{j'}}, j' \in N_{x^*}, i' \in N_{x^*}) \in \mathbb{R}^{q_{i} - q_{j}}$. In the same way, the two stacked vectors are denoted as $\hat{x}^j\sim j = \text{col}(\hat{x}^j_{i \sim j}, j' \in N_{x^*})$. Next, we give special matrices $S_i$ and $R_i$ to extract values from $\hat{x}^j_i$, and they are defined as

$$
S_i = \begin{bmatrix}
I_{(n_{j'q'}, j') \in N_{x^*}} & 0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} & \cdots & 0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} \\
0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} & I_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} & \cdots & 0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} \\
\cdots & \cdots & \cdots & \cdots \\
0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} & 0_{(n_{j'q'}, q') \times (n_{j'q'}, n_{j'q'})} & \cdots & I_{(n_{j'q'}, q')}
\end{bmatrix},
$$

(10)

where $n_{j'q'} = \sum_{j' > j}^j n_{j'q'}$ and $n_{j'q'} = \sum_{j' > j}^j n_{j'q'}$. Note that $R_i$ selects the local strategy vector of agent $i$ in cluster $j$ from its estimation of all agents in the game, i.e., $R_i \hat{x}^j = \hat{x}^j_i$, while $S_i$ removes the whole cluster $j$’s strategy vector from $\hat{x}^j_i$, i.e., $S_i \hat{x}^j = \hat{x}^j_{i \sim j}$.

We denote $R^j = \text{diag}(R_i^{(j)}_{i \in N_{x^*}}), \quad S^j = \text{diag}(S_i^{(j)}_{i \in N_{x^*}})$, then it’s easy to derive that $R^j \hat{x}^j = \hat{x}^j, \quad R^j \hat{x}^j = \hat{x}^j_{i \sim j}, \quad S^j \hat{x}^j = \hat{x}^j_{i \sim j}$.

For partial-decision scenario, the minimization problem of cluster $j$ in (3) is rewritten as:

$$
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n_{ij}} f_i^j(x^i, \hat{x}^j_i - x^i) + \sum_{i=1}^{n_{ij}} h_i^j(x^i) \\
& \text{s.t.} \quad L^j R^j \hat{x}^j = 0_{n_{ij}}, \\
& \quad A^j R^j \hat{x}^j \leq b - \sum_{j \in N_{x^*}, \setminus j} A^j L^j \hat{x}^j,
\end{align*}
$$

(11)

The Lagrangian function of partial-decision scenario is defined as

$$
L^j(R^j \hat{x}^j, \hat{x}^j_{i \sim j}) = \theta^j(R^j \hat{x}^j, \hat{x}^j_{i \sim j}) + (\lambda^j)^T L^j R^j \hat{x}^j
$$

(12)
When \( \hat{x}^* \) is a v-GNE of partial-decision scenario in the game (11), then the following KKT conditions (22) must be satisfied:

\[
\begin{cases}
0_{nq} \in R^T F (\hat{x}^*) + R^T H (R \hat{x}^*) + R^T L x^* + R^T A^T \mu^*, \\
0_{w} \in b - R A \hat{x}^* + N_{\lambda^*} (\mu^*), \\
0_{nq} = LR \hat{x}^*,
\end{cases}
\]

where \( F (\hat{x}) = \text{col}((\nabla_{w} f_j (x^j, \tilde{x}^{j - \xi})), j \in \mathcal{N}_n) \) is the extended mapping of \( F(x) \) as [34] defined, and \( \mu = 1_m \otimes \mu \in \mathbb{R}^{w m}, \Lambda = \text{diag}((A^T_j)_{j \in \mathcal{N}_m}) \in \mathbb{R}^{w m \times q} \).

### Assumption 6

The extended pseudo-gradient \( F \), as defined in (13), is Lipschitz continuous, i.e., there exists \( \kappa > 0 \) such that for any \( \hat{x} \) and \( \hat{x}' \), \( \|F(\hat{x}) - F(\hat{x}')\| \leq \kappa \|\hat{x} - \hat{x}'\| \).

Note that a v-GNE of (11) is given by \( \hat{x}^* = 1_n \otimes x^*, \) while the same \( x^* \) is the solution of (7), and the v-GNE of (9). We show the details in convergence analysis section III.

### III. DISTRIBUTED ALGORITHM

In this section, we present a distributed algorithm based on Forward-Backward-Forward approach.

In the scenario of partial-decision, agent \( i \) in cluster \( j \) have to estimate other agents’ strategies. However, it only estimates strategies outside cluster \( j \), i.e., \( \tilde{x}_i^{j - \xi} \in \mathbb{R}^{n_i \times n_j} \), since it already processes its local strategy profile \( x^j \in \mathbb{R}^{n_i \times n_j} \). Moreover, \( \lambda_i \in \mathbb{R}^{n_i} \) and \( \mu_i \in \mathbb{R}^{n_j} \) are dual multipliers for the estimation of components of vector \( \lambda^* \) and \( \mu^* \) respectively. And \( z^j \in \mathbb{R}^w \) denote a local auxiliary variable used for the coordination needed to satisfy the coupling constraint and to reach consensus of the dual variable \( \mu^j \). Moreover, \( \lambda_i \) is a variable used to make all agents’ strategies to come consensus while these agents are in the same cluster.

In Algorithm 1 we split iteration \( k \rightarrow k + 1 \) into two procedures: \( k \rightarrow k' \) and \( k' \rightarrow k + 1 \).

In procedure \( k \rightarrow k' \), if \( i = 1 \) (leader), then agent obtains information of real strategies and gradient information of other agents in the same cluster \( j \in \mathcal{N}_n \) from virtual center to calculate \( v_i^j \) and \( v_i^j \), where \( v_i^j := \nabla x_i f_j^i (x^j, \tilde{x}_i^{j - \xi}) \) and \( \hat{v}_i^j := \sum_{l=1}^{n_j} \nabla_j (f_j^i (x^j, \tilde{x}_i^{j - \xi})) \). Moreover, agent \( i \) exchanges local estimation information of other agents’ agents \( \hat{x}_i^{j - \xi} \) and multiplier information \( \lambda_i \) via graph \( G^j = (\mathcal{N}_j, \mathcal{E}_j) \). And \( (\xi, i) \in \mathcal{E} \) hold if agent \( i \) in cluster \( j \) receives \( \{\tilde{x}_i^{j - \xi}, \lambda_i\} \) from the neighboring agent \( \xi \) in cluster \( j \), where \( \xi \in \mathcal{N}_n := \{\xi \mid (\xi, i) \in \mathcal{E}_j\} \). The updating procedure of \( x_i^j \) and \( \tilde{x}_i^{j - \xi} \), \( \lambda_i \) is given by:

\[
x_i^j[k'] = \text{Prox}_{\rho_i^j}^\frac{1}{2} (x_i^j[k] - \tilde{x}_i^{j - \xi}[k]) + \sum_{l \in \mathcal{N}_j}^\frac{1}{2} w_{i,l}^j (\lambda_i^j[k] - \lambda_l^j[k]),
\]

\[
\hat{x}_i^{j - \xi}[k'] = \hat{x}_i^{j - \xi}[k] - \rho_i^j \sum_{l \in \mathcal{N}_j}^\frac{1}{2} w_{i,l}^j (\hat{x}_i^{j - \xi}[k] - \hat{x}_l^{j - \xi}[k]),
\]

\[
x_i^j[k + 1] = x_i^j[k] + \rho_i^j \sum_{l \in \mathcal{N}_j}^\frac{1}{2} w_{i,l}^j (x_i^j[k] - \hat{x}_l^{j - \xi}[k]),
\]

\[
\rho_i^j = \rho_i^j (\sum_{l \in \mathcal{N}_j}^\frac{1}{2} w_{i,l}^j (\lambda_i^j - \lambda_l^j)[k] - \hat{v}_i^j[k - k']) + \lambda_{\text{col}}^j (A^T \mu_i[k - k'])
\]
\[ x_i^j[k+1] = cp_i^j(w_i^j, \sum_{\xi \in \Omega_i^j} (x_i^\xi[k] - x_i^j[k]) + \bar{x}_i^j[k], \]
\[ + w_i^j, \sum_{\xi \in \Omega_i^j} (x_i^\xi[k] - x_i^j[k]) + \bar{x}_i^j[k], \]
\]  

(21)

\[ \lambda_i^j[k+1] = \lambda_i^j[k] - \sigma_i^j \left( \sum_{\xi \in \Omega_i^j} (x_i^\xi[k] - x_i^j[k]) \right), \]
\[ z_i^j[k+1] = z_i^j[k] + \sigma_i^j \left( \sum_{\xi \in \Omega_i^j} (x_i^\xi[k] - x_i^j[k]) \right), \]
\[ \mu_i^j[k+1] = \mu_i^j[k] - \sigma_i^j \left( \sum_{\xi \in \Omega_i^j} (z_i^\xi[k] - z_i^j[k]) \right), \]
\[ - A_i^j x_i^j[k - k'], \]

\]  

(24)

where \( x_i^j[k+1], \bar{x}_i^j[k+1], \lambda_i^j[k+1], z_i^j[k+1], \mu_i^j[k+1] \) denote \( x_i^j, \bar{x}_i^j, \lambda_i^j, \) at iteration \( k+1. \)

If \( i \neq 1 \) (not leader), then agent obtains \( x_i^j, v_i^j \) from virtual central, \( \hat{x}_i^j, \lambda_i^j \) from neighbors of inner-cluster graph at iteration \( k'. \) \( x_i^j[k+1] \) is updated by (20) while \( \lambda_i^j[k+1] \) is updated by (22) while \( w_i^j[k+1] = 0; \lambda_i^j[k+1] \) is updated by (22). Then the procedure \( k' \to k+1 \) ends.

**Algorithm I** Distributed Multi-cluster Game with Partial-decision Information Algorithm (DMGPA)

**Initialization:** \( x_i^j[0] \in \Omega_i^j \subset \mathbb{R}^{n_i}, \bar{x}_i^j[0] \in \mathbb{R}^{n_i, n_j}, \lambda_i^j[0] \in \mathbb{R}^{n_i}, z_i^j[0] \in \mathbb{R}^{w}, \mu_i^j[0] \in \mathbb{R}^{w}. \)

**Iteration:** \( k \to k' \):

1. leader agent in cluster \( j \) receives \( \{ x_i^j[k], v_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from virtual central, \( \{ \hat{x}_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in leader-cluster graph, \( \{ \lambda_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in inner-cluster graph, then:
   \[ \text{updates } x_i^j[k], \bar{x}_i^j[k], \lambda_i^j[k], z_i^j[k], \mu_i^j[k] \text{ by (14) and (15) while } \lambda_i^j[k] \text{ by (16)}; \]

2. non-leader in cluster \( j \) receives \( \{ x_i^j[k], v_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from virtual central, \( \{ \lambda_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in inner-cluster graph, then:
   \[ \text{updates } x_i^j[k], \bar{x}_i^j[k], \lambda_i^j[k], z_i^j[k], \mu_i^j[k] \text{ by (14) and (15) while } \lambda_i^j[k] \text{ by (16)}; \]

**Iteration:** \( k' \to k+1 \):

1. leader agent in cluster \( j \) receives \( \{ x_i^j[k], v_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from virtual central, \( \{ \hat{x}_i^j[k], \lambda_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in leader-cluster graph, \( \{ \lambda_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in inner-cluster graph, then:
   \[ \text{updates } x_i^j[k+1], \bar{x}_i^j[k+1], \lambda_i^j[k+1] \text{ by (20) and (21) while } \lambda_i^j[k+1] \text{ by (22); } \mu_i^j[k+1] \text{ by (24) and (25) while } \lambda_i^j[k+1] \text{ by (22)}; \]

2. non-leader in cluster \( j \) receives \( \{ x_i^j[k], v_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from virtual central, \( \{ \hat{x}_i^j[k], \lambda_i^j[k] \}_{i \in \mathcal{N}_{i,j}} \) from neighbors in inner-cluster graph, then:
   \[ \text{updates } x_i^j[k+1], \bar{x}_i^j[k+1], \lambda_i^j[k+1] \text{ by (20) and (21) while } \lambda_i^j[k+1] \text{ by (22).} \]

For cluster \( j \), we assume that the information of all agents (strategies \( x_j \) and gradients \( v_j = [v_j^1, \ldots, v_j^d] \)) are shared by a virtual center before and after iteration procedure. Furthermore, we assume that the information of agents outside this cluster is unknown, and each agent can only replace the true value by prediction. Meanwhile, the estimation value \( \hat{x}_i^j[k] \), \( \mu_i^j[k] \), \( z_i^j[k] \) need to be transmitted through the two undirected graphs and finally achieve the same result as the full decision information.

When Algorithm starts, agent \( i \) in cluster \( j \) needs to update \( x_i^j, \bar{x}_i^j[k], \lambda_i^j[k], z_i^j[k], \mu_i^j[k] \) from \( k \) to \( k' \), and it needs to update from \( i = 1 \) to \( i = n_j \) (all agents in the cluster \( j \) need to be updated), meanwhile, we use \( w_i^j[k] \) to make a distinction between leader agent and normal agent, since \( w_i^j[k] \) will equal to 0, if it is not a leader agent. The same as indicator function \( \lambda_i^j[k] \), it equals to 0 while it is not a leader agent. Then agents in cluster \( j+1 \) start to update from \( k \) to \( k' \) when all agents in the cluster \( j \) are finished updating. Repeat the procedure from \( j = 1 \) to \( j = m \) until all agents in the game finished updating. Then we start procedure \( k' \to k+1 \), and the whole process is the same with first one. Since the objective function of each agent is related to the strategies of all agents \( x \), each agent needs the strategy information of all agents when updating its strategy.

**Remark:** Algorithm I is called distributed since every agent only knows its local information of cluster, the information of other agents outside the cluster can only be estimated. However, if each agent’s objective function is only related to strategies of itself and agents outside the cluster \( \theta_i^j(x_i^j, x^{\mathcal{J}} - x_i^j) \) as in [28], [50], or only related to the strategy of itself and other clusters’ leaders \( \theta_i^j(x_i^j, \Gamma(x^{\mathcal{J}})) \) where \( \Gamma(x^{\mathcal{J}}) \) is the stacked strategies of the leader agents of all the clusters except that of cluster \( j \) as in [28], there is no need for a virtual center in each cluster. And the \( v_i^j \) in Algorithm I becomes \( v_i^j \). Algorithm I changes from cluster distributed to agents distributed. The same virtual center setting can also be seen in [51]. The situation in this paper covers the above-mentioned situations.

**IV. ALGORITHM DEVELOPMENT AND CONVERGENCE ANALYSIS**

**A. Algorithm Development**

In this section, we show how the distributed algorithm for seeking the GNE under partial-decision information is developed and gives the convergence analysis.

In compact notation, denote \( x[k] = col((x_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}), \)
\[ \bar{x}[k] = col((\bar{x}_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}), \]
\[ \lambda[k] = col((\lambda_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}), \]
\[ z[k] = col((z_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}), \]
\[ \mu[k] = col((\mu_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}) \]
\[ b = col((b_i^j[k])_{j \in \mathcal{N}_{i,j}, i \in \mathcal{N}_{i,j}}), \]
\[ F(x) \]

then \( k \to k' \) of Algorithm I can be equivalently written in the compact form as follows:

\[ x[k'] = Prox_{H}^\rho_{\lambda}(x[k] - \rho_{\lambda}(F(\hat{x}[k]) + cR(\hat{x}[k]) \]
\[ + \mathcal{L}(k) + \Lambda^\top \mu[k])), \]
\[ \hat{x}^{-1}[k] = \hat{x}^{-1}[k] - cp_{\pi(S(\hat{x}[k]) \]

(26)
\[ \lambda [k'] = \lambda [k] + \tau L R \dot{x}[k], \]
\[ z [k'] = z [k] - \sigma L_m \mu [k], \]
\[ \mu [k'] = \rho_{B}^\omega (\mu [k] + v (L^0_m z [k] + \Lambda \dot{R}_x [k] - b)), \]
and \( k' \rightarrow k + 1 \) compact form is represent as:
\[ x[k + 1] = x[k'] + \rho_R L \lambda [k' - k] + \rho_R \Lambda [k' - k] + \rho_R F (\dot{x}) [k' - k] + c \rho_R R (L + L^0_m) \dot{x}[k] - k', \]
\[ \dot{x}^{-j}[k + 1] = \dot{x}^{-j}[k'] + c \rho_S S (L + L^0_m) \dot{x}[k] - k', \]
\[ \lambda [k + 1] = \lambda [k'] - \tau L R \dot{x}[k] - k', \]
\[ z[k + 1] = z [k'] - v \dot{L}^0_m z [k'] - \nu \Lambda \dot{R}_x [k] - k'. \]

Where \( \rho = \text{diag}(\rho_i^1 I_q), \rho_s = \text{diag}(\rho_i^1 I_{n_q - \sum_{j=1}^{m} n_j + n_q}), \rho_R = \text{diag}(\rho_i^1 I_Q), \tau = \text{diag}(\tau_i^1), \sigma = \text{diag}(\sigma_i^1), \nu = \text{diag}(\nu_i^1), L = \text{diag}(L_j^q), L^0_m = L^0_m \otimes I_w, \dot{L} = L \otimes I_q, \dot{L}^0_m = \dot{L}^0_m \otimes I_q, \]
and every item in \( \dot{L}^0_m \) is a matrix of particular size which is related to the corresponding item in \( L^0_m \), e.g., for item \( -w_{j,1}^{1,l} \), the corresponding item is \( -w_{j,1}^{1,l} \), according to its position: row \( j \) column \( l \) in \( L^0_m \), it expands to size \( n_j \times n_j \) as:
\[ \begin{bmatrix} -w_{j,1}^{1,l} & \cdots & -w_{j,1}^{m,l} \\ \vdots & \ddots & \vdots \\ -w_{j,1}^{m,1} & \cdots & -w_{j,1}^{m,m} \end{bmatrix} \]
and for items on the diagonal, if the position is row \( j \) column \( j \), it expands to size \( n_j \times n_j \). This avoids the leader vector operating as \( [12] \).

Denote \( \varpi = \text{col}(\dot{x}, z, \lambda, \mu) \in \Omega, \) where \( \Omega := \mathbb{R}^{n_q \times n_w} \times \mathbb{R}^{q} \times \mathbb{R}^{n_w} \). Denote the block-diagonal matrix of the step sizes and the skew symmetric respectively as:
\[ \Psi = \text{diag}(\rho^{-1}, \sigma^{-1}, \tau^{-1}, \nu^{-1}), \]
and
\[ \Phi = \begin{bmatrix} 0 & 0 & R^T L^T & R^T \Lambda ^T \\ 0 & 0 & 0 & 0 \\ -LR & 0 & 0 & 0 \\ -\Lambda R & -L^0_m & 0 & 0 \end{bmatrix}. \]

Define \( A : \Omega \rightarrow \mathbb{R}^{n_q + q + 2n_w}, B : \Omega \rightarrow \mathbb{R}^{n_q + q + 2n_w} \)
as:
\[ A : \varpi \mapsto \text{col}(R^T F (\dot{x}) + c(L + L^0_m) \dot{x}, 0_w, 0_q, b) + \Phi \varpi, \]
\[ B : \varpi \mapsto R^T H(R \dot{x}) \times 0_w \times 0_q \times N_{R^2_w} (\mu), \]
where \( R^T H(R \dot{x}) := \{ R^T v | v \in H(R \dot{x}) \}, N_{R^2_w} (\mu) = \prod_{j=1}^{m} N_{R_{0,j}^2} (\mu_j). \]

Assumption 7: \( \Psi^{-1} \succ 0 \) and \( \|\Psi^{-1}\| < 1/\ell_A \), where \( \ell_A \) is the Lipschitz constant of \( A \) (We prove the Lipschitz property in convergence analysis section.).

Remark: Suppose Assumption 7 holds. Then the biggest step-size of Algorithm 1 \( \max((P^j, \tau_i^j, \sigma^j, \nu^j)_{j \in \mathbb{N}_m, i \in \mathbb{N}_n}) \in (0, 1/\ell_A). \)

Theorem 1: Suppose Assumption 7 hold. Let \( \varpi [k] = \text{col}(\dot{x}[k], z[k], \lambda[k], \mu[k]), \) and \( \Psi, \Phi, A \) and \( B, \) as in (35), (36), (37). Then Algorithm 1 is equivalent to
\[ 0 \in A (\varpi [k]) + B (\varpi [k]), \]
or
\[ \varpi [k] := T \varpi [k], \]
where \( T := \Psi^{-1} A + (I_d - \Psi^{-1} A) \circ J_{\Psi - 1 B} \circ (I_d - \Psi^{-1} A), J_{\Psi - 1 B} := (I_d + \Psi^{-1} B)^{-1} \). And any limit point of Algorithm 1 is a zero of \( A + B \) and a fixed point of \( T \).

Proof: See appendix A for details.
max \((p_i^l, \tau_i^l, \sigma_i^l, \nu_i^l)\)\(\in \mathbb{N}_{m_i}, i \in \mathbb{N}_{n_i}\) \((0, 1/\ell_iA)\), then the operators \(\Psi^{-1}A\) and \(\Psi^{-1}B\) satisfy the following properties under the \(\Psi\)-induced norm \(\|\cdot\|\Psi\):

1. \(\Psi^{-1}A\) and \(\Psi^{-1}B\) are maximally monotone.
2. \(\Psi^{-1}A\) is single-valued and Lipschitz continuous with parameter \(\ell_iA\).

**Proof:** See appendix \(A\) for details.

In the following, we show the convergence of Algorithm \([1]\)

**Theorem 3:** Suppose Assumption 1-7 hold, and \(\Psi, \Phi, A, B\), as defined in (38), (39), \((\tilde{q}_s)\), \(c \geq c_{\min}\) as in (2), \(\max\{(\rho_i^l, \tau_i^l, \sigma_i^l, \nu_i^l)\} \in (0, 1/\ell_iA)\), the sequence \(\{\tilde{w}_i[k]\}_{k \geq 0}\) generated by Algorithm \([1]\) is Fejér monotone with respect to the fixed points set of \(T\), and converges to \(\text{zer}(A+B)\). Moreover, for each agent, like agent \(i\) in cluster \(j\), the prime variable \(\tilde{x}^j_i[k]\) \(\in \{\tilde{w}_i[k]\}_{k \geq 0}\) generated by Algorithm \([1]\) converges to \(\tilde{x}^j_i\), which is a component of \(v\)-GNE (\(\tilde{x}^j_i\)) of (3), and its local component, \(\{\tilde{x}^j_i[k]\}_{k \geq 0}\) where \(j \in \mathbb{N}_{m_i}, i \in \mathbb{N}_{n_{j}}\), converges to the corresponding component \(\tilde{x}^j_i\) in \(\tilde{x}^j\).

**Proof:** See appendix \(A\) for details.

### V. APPLICATION IN ENERGY INTERNET

In this section, a best energy generation strategy problem of EI is formulated as a GNP of multi-cluster game, and the proposed GNE seeking algorithm is implemented to solve this problem in a distributed manner. A numerical simulation example is carried out to validate the effectiveness of the algorithm.

We assume that the EI is composed of several energy subnets while each subnet is constituted with multiple prosumers who own energy sources and can both produce and consume energy as in \([52]\). And every prosumer is an energy circle contains energy facilities (EFs), energy loads (ELs), energy storage devices (ESs). EFs are usually clean renewable facilities like wind turbines or solar photovoltaics and so forth, and also some conventional energy facilities like fossil-fuel power stations. ELs are equipments that consumes energy which can be household appliances or machines used in factories. ESs are essential storage devices which are used to store energy since renewable energy facilities have intermittent and energy fluctuation. Suppose that all energy-subnets can store energy since renewable energy facilities have intermittent factories. ERs are essential storage devices which are used to which can be household appliances or machines used in

In every energy subnet, prosumers are connected by both communication networks which deliver estimations and sometimes multiplexers. Meanwhile, leader prosumer of subnets also communicate estimations and some multiplexers in same structure network. To the purpose of unified management, we suppose prosumers must obey constraint rules of energy control center. Since every facility in the same energy subnet follows the same strategy, we assume all leaders satisfy constraint rules instead of all prosumers.

Denote \(m\) the number of the energy subnets, and for energy subnet \(j\), there are \(n_j\) prosumers, and each prosumer has several facilities to generate power. All these energy subnets compete for \(w\) energy markets. We assume every prosumer has respective ELs and ESs. \(x^j_i \in \mathbb{R}^{n_j}\) denotes the quantity of stored energy of prosumer \(i\) in energy subnet \(j\). Similarly, \(\bar{x}\) is the power of all energy subnets can provide for competing the utility markets. We consider \(\theta_{ji} < x^j_i < r^j_i\), where \(r^j_i \in \mathbb{R}^{n_j}\) represents the maximum capacity of ESs of prosumer \(i\) in energy subnet \(j\). After satisfying the power consumption by ELs in respective energy subnet, power stored by ESs is used to compete for the utilities. The proper \(x^j_i\) is meaningful for the schedule planning of the maintenance or management of the whole energy-subnet.

Suppose every prosumer \(i\) in energy subnet \(j\) has an energy allocation matrix \(T_i^j \in \mathbb{R}^{w \times n_j}\) to specify which utility market it participates in. Like \([34]\), if the \(e\)-th column of \(T_i^j\) has its \(s\)-th element as \(1\) and only if prosumer \(x^j_i\) power to utility \(T_{es}\), and all other elements are 0, where \(x^j_i\) power to utility 

Assume that every prosumer \(j\) has a strong convex quadratic production payoff function \(c^j(x^j_i) = (x^j_i)^T Q^j x^j_i + q^j_i^T x^j_i - o_i\) where \(Q^j > 0\), \(o > 0\) and \(q^j_i \in \mathbb{R}^{n_j}\). Since all prosumers are involved in competing for energy utilities, then the demand function of utility \(s\) is given as \(p^j_s(x) = (p^j_s)^T x^j_i + d^j_s(T x_s)\), where \(p^j_s, d^j_s > 0\). Then the objective function of prosumer \(i\) in energy subnet \(j\) is defined as \(f^j(x^j_i, \bar{x}^j_i) = c^j(x^j_i) - (p^j_s)^T x^j_i + d^j_s(T x_s)\), and the objective function of energy subnet \(j\) is defined as \(f^j(x^j, \bar{x}^j) = \sum_{i=1}^{n_j} f^j_i(x^j_i, \bar{x}^j_i)\). And all leaders must satisfy a coupled constraint \(\sum_{i=1}^{n_j} A_i x^j_i \leq \sum_{i=1}^{n_j} b_i\), where \(A_i \in \mathbb{R}^{w \times n_j}\) stands for constraint matrix, and \(b_i \in \mathbb{R}_+^\mathbb{R}\).

Then we consider partial-decision scenario, i.e., every prosumer can obtain the real strategies information in energy subnet, however, it can’t easily obtain the information outside the energy subnet. Firstly, we denote \(\tilde{x}^j_i\) the estimation of agent \(i\) in cluster \(j\), and \(R_i^j\) the matrix defined as \([10]\), i.e., \(R_i^j \tilde{x}^j_i = x^j_i\). Moreover, \(R_i^j = \text{diag}((R_i^j)_{i \in \mathbb{N}_{n_j}})\), \(\bar{R} = \text{diag}((R_i^j)_{i \in \mathbb{N}_{n_j}})\). \(R_i^j\) is defined as \(R_i^j = \text{diag}((R_i^j)_{i \in \mathbb{N}_{n_j}})\). Then the objective function is rewritten as \(f^j(x^j, \bar{x}^j_i) = (x^j_i)^T Q^j x^j_i + (q^j_i)^T x^j_i + s^j_i + \sum_{j=1}^{m_j} s^j_i + (x^j_i)^T (T^j T^j)^T (d^j_s)^T x_s^j, \) where \(\tilde{x}^j_i\) stands for the estimation of prosumer \(i\) in energy subnet \(j\) to prosumer \(i\)’s energy subnet \(j\). Since \(F(\tilde{x}) = \text{col}((\nabla x^j_i f^j(x^j_i, \bar{x}^j_i))_{i \in \mathbb{N}_{n_j}})\), for the ease of notation, we denote \(\tilde{T} = \text{diag}(T^1, \ldots, T^m)\), \(\Xi := \sum_{j=1}^{m_j} n_j n_j g_j\), \(\tilde{T} = \text{diag}(T^1, \ldots, T^m) \in \mathbb{R}^{w \times m}\), and \(\tilde{T} = \text{diag}(T^1, \ldots, T^m) \in \mathbb{R}^{w \times m}\), \(Q = \text{diag}((Q^j)_{i \in \mathbb{N}_{n_j}, i = n_j})\). \(\tilde{D} = \text{diag}((d^j_s)_{i \in \mathbb{N}_{n_j}, i = n_j}) \in \mathbb{R}^{w \times n_j}\), \(p = \text{col}((q^j_i)_{i \in \mathbb{N}_{n_j}})\), and \(\eta, \bar{r}, b\) are defined similarly. Then \(\tilde{F}(\tilde{x}) = 2Q^T + q - \tilde{T}^T \tilde{D} \tilde{T} \tilde{R} \tilde{x} + \tilde{T} \tilde{x}\).

We consider the EI consisted of 3 energy subnets, as
shown in Fig.2, where Energy subnets-1 has 4 prosumers, Energy subnets-2 has 2, Energy subnets-3 has 3, and they are competing for 2 utilities. All prosumers are connected by both-way communication networks.

Fig.2. Multi-cluster game of EI: Energy subnets-1(2,3) represent clusters, prosumers represent agents, and each prosumer has EF (like wind turbines), EL and ES.

And leader prosumers are required to satisfy constraints proposed by energy control center. In this setting, we assume $r^j_i$ is randomly drawn from $(5, 10)$, and the same with $b^j_i \in (1, 2)$, $p^j_i \in (10, 20)$, $d^j_i \in (1, 3)$, $q^j_i \in (1, 2)$, $o = 3$, $Q^j_i$ is diagonal with its entries from $(1, 8)$, and set $c = 70$, $\lambda = \tau = \sigma = \nu = 0.002$, then the trajectories, including $x^j_i$, $\frac{1}{n_j-1} \sum_{i=2}^{n_j} \| x^j_1 - x^j_i \|$, $\frac{1}{n_j} \sum_{i=1}^{n_j} \| x^j_i - x^j_i^* \|$ and $\sum_{j=1,j\neq 3}^{3} \sum_{i=1}^{n_j} \| x^j_3 - x^j_3^* \|$ are shown in Fig.3, Fig.4 and Fig.5. The relative error $\frac{\| \hat{x}^k[i] - \hat{x}^k[i^*] \|}{\| \hat{x}^k[i] \|}$ generated by Algorithm 1 is shown in Fig.6.

As shown in Fig.3 and Fig.4, the results satisfy the consistency constrain condition, and show the convergence of Algorithm 1. Fig. 5 demonstrates that every prosumer’s estimations to prosumer 3 in cluster 3 come to a consensus $x^3_3^*$, which is a component of the Nash equilibrium. Fig. 6 shows that the relative error asymptotically decreases to zero, implying that $\{ \hat{x}^k[i] \}_{k \geq 0}$ asymptotically converges to the Nash equilibrium.

VI. CONCLUSION

In this paper, we proposed a distributed GNE seeking algorithm designed by FBF iteration method of multi-cluster game with shared affine coupling constraints of leaders under partial-decision information scenario. Convergence holds
under restricted monotone of the extended pseudo-gradient equation is

\[ (\dot{z}^+) = -L \dot{\hat{z}}^+ + b + N_{R_w^0}(\mu^*) . \]

Proof of Theorem 2: Consider \( \omega^* = \text{col}(\hat{x}^*, z^*, \lambda^*, \mu^*) \) in \( \text{zer}(A + B) \), with \( \Psi, \Phi, A, \) and \( B \), as defined in \( (35), (36), (37) \), \( c \geq c_{\min} \) as in \( (2) \). Then it follows that

\[ 0_{nq} \in R^T F(\hat{x}^*) + c(\dot{L} + \dot{L}_{m}^0 \hat{x}^* + R^T L^T \lambda^*) + R^T \Lambda^T \mu^* + R^T H(\dot{R} \hat{x}^*) , \]

and (45) with \( (1_n^T \otimes I_q) \dot{L}_{m}^0 = 0_{nq} \), \( (1_n^T \otimes I_q) \dot{\hat{L}} = (1_n^T \otimes I_q) (L \otimes I_q) = 1_n^T L \otimes I_q = 0_{nq} \), we preemptively \( 1_n^T \otimes I_q \) to the first line yields

\[ (1_n^T \otimes I_q) 0_{nq} = (1_n^T \otimes I_q) R^T F(\hat{x}^*) + (1_n^T \otimes I_q) R^T \Lambda^T \mu^* + (1_n^T \otimes I_q) R^T L^T \lambda^* + (1_n^T \otimes I_q) R^T H(\dot{R} \hat{x}^*) , \]

and by \( (1_n^T \otimes I_q) R^T = I_q \), it follows that

\[ 0_{q} \in F(\hat{x}^*) + H (x^*) + L^T \lambda^* + \Lambda^T \mu^* . \]

If we preemptively \( R^T \), it follows that the first line of KKT condition \( (33) \). And we can derive that \( (L + \dot{L}_{m}) \omega^* = 0_{nq} \). By \( \dot{L} = \text{diag}(L^i)_{i \in N^q} , \dot{L}_{m}^0 = L_{m}^0 \otimes I_q \), it follows that \( \hat{x}^* \) satisfies \( \hat{x}^*_i = \hat{x}^*_{i'} \), \( \hat{x}^*_i = \hat{x}^*_{i} \), when \( i, i' \in N^q \) and \( j, l \in N^m \). Thus \( \hat{x}^* \in E \) the consensus subspace, then \( \hat{x}^*_i = 1 \omega^* \), then \( F(\hat{x}^*) = F(1_n \otimes x^*) = F(x^*) \) holds, and by \( \Lambda^T \mu^* = A^T \mu^* \), it follows that

\[ 0_{q} \in F(x^*) + H (x^*) + L^T \lambda^* + \Lambda^T \mu^* , \]

which is the first line of KKT condition \( (9) \).

\[ L_{m}^0 \mu^* = (1_n^T \otimes I_q) \cdot (1_n^T \otimes \mu) = L_{m}^0 1_n \otimes I_w^\mu = 0_{m} \otimes I_w^\mu = 0_{w} \text{w.m.} . \]

In \( (37) \), \( \dot{R} \hat{x}^* = x^* \), we can get \( Lx^* = 0_{q} \). It is the third line of KKT condition \( (13) \). In \( (36) \), \( L_{m}^0 = L_{m}^0 \otimes I_w \in R^{u_m \times w_m} , \mu^* = 1 \omega^* \) \( \in R^{u_m} \), it follows that

\[ L_{m}^0 \mu^* = (1_n^T \otimes I_q) \cdot (1_n^T \otimes \mu) = L_{m}^0 1_n \otimes I_w^\mu = 0_{m} \otimes I_w^\mu = 0_{w} \text{w.m.} . \]

In \( (47) \), \( \dot{x}^* = x^* \), we can get \( Lx^* = 0_{q} \). It is the third line of KKT condition \( (13) \) and \( (9) \).

In \( (48) \), assuming that \( v_1, v_2, \ldots, v_m \in N_{R_{w}^0}(\mu^*) \), using \( L_{m}^0 = L_{m}^0 \otimes I_w , A = \text{diag}(A^i)_{i \in N^q} , (1_n^T \otimes I_q) L_{m}^0 = 0_{w} \text{w.m.} , (1_n^T \otimes I_w) A = A , (1_n^T \otimes I_w) b = \sum_{i=1}^m b^i = b \), we preemptively \( 1_n^T \otimes I_w \) to \( (37) \) it yields

\[ (1_n^T \otimes I_w) 0_{w} = - (1_n^T \otimes I_w) L_{m}^0 z^* - (1_n^T \otimes I_w) \Lambda \dot{R} \hat{x}^* + (1_n^T \otimes I_w) b + (1_n^T \otimes I_w) N_{R_{w}^0}(\mu^*) , \]

and it can be equivalently written as \( 0_{w} = -A x^* + b + \sum_{i=1}^m v_i \), which is \( 0_{w} = b - A x^* + N_{R_{w}^0}(\mu^*) \), the second line of KKT condition \( (13) \) and \( (9) \).

Proof of Lemma 2: Define \( E = \{ \dot{x} \in R^{nq} | \dot{x} = \dot{x}^*_i , \forall j, j' \in N^q \} = \{ \dot{x} = 1_n \otimes x , x \in R^q \} \) as the estimation consensus subspace, \( E^\perp \) its orthogonal complement with \( R^{nq} = E + E^\perp \). Since \( R^{nq} = E + E^\perp \), where \( E = \text{Null}(L + \dot{L}_{m}) = \{ \dot{x} \in R^{nq} | \dot{x} = \dot{x}^*_i , \forall j, j' \in N^q \} \), \( \dot{x} = 1_n \otimes x , x \in R^q \) , \( E^\perp = \text{Null}(L + \dot{L}_{m}) \) are the estimate consensus subspace and its orthogonal complement respectively. Any \( \dot{x} \) can be decomposed as \( \dot{x} = \dot{x}^+ + \dot{x}^* \), where \( \dot{x}^+ \in E \) and \( \dot{x}^* \in E^\perp \). Thus \( \dot{x} = 1_n \otimes x \) for \( x \in R^q \).
And \( \min_{x \in \mathbb{E}} \langle x \rangle^T (x) = s_2(L) \| x \|_2^2 \), since \( s_2(L) = s_2(L \otimes I_n) = s_2(L) \). We assume any \( x' \in \mathbb{E} \) can be rewritten as \( x' = 1_n \otimes x' \), for any \( x' \in \mathbb{R}^n \). Then using \( x = x' + \hat{x}' \), \( x = \hat{x}' \), \( R \hat{x} = x \), \( R \hat{x}' = x \), \( (\hat{x} - \hat{x}')^T (R^T F(\hat{x}) - R^T F(\hat{x}')) + c(L + \hat{L}_m)(\hat{x} - \hat{x}') \) can be rewritten as the following [34],

\[
(\hat{x} - \hat{x}')^T (R^T F(\hat{x}) - F(\hat{x}')) + (\hat{x}')^T R^T (F(\hat{x}) - F(\hat{x}')) + (\hat{x}')^T R^T (F(\hat{x}) - F(\hat{x}')) + c(\hat{x}')^T R \hat{x} - c(\hat{x}')^T R \hat{x} = 0,
\]

by Assumption 1 and \( \|R\| = 1 \), \( \|x' - x\| = \|1_n \otimes (x - x')\| = \|1_n \otimes (x - x')\| = \|1_n \otimes (x - x')\| = \|1_n \otimes (x - x')\| = \|1_n \otimes x - x\| \), so it follows that

\[
(\hat{x} - \hat{x}')^T (R^T F(\hat{x}) - R^T F(\hat{x}')) + c(L + \hat{L}_m)(\hat{x} - \hat{x}') \] 

\[ \geq -\kappa \|x - x'\|_2^2 + \kappa \|\hat{x} - \hat{x}'\|_2^2 + c \|L + \hat{L}_m\| \|\hat{x} - \hat{x}'\|_2^2 \]

\[ \geq \frac{\kappa}{\sqrt{n}} \|\hat{x} - \hat{x}'\|_2^2 + \frac{\kappa}{\sqrt{n}} \|\hat{x} - \hat{x}'\|_2^2 \]

\[ \geq \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) + \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) \]

\[ \geq \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) + \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) \]

\[ \geq \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) + \left( \frac{\kappa}{n} \right)(\|\hat{x} - \hat{x}'\|_2^2) \]

Thus, \( \|\hat{x} - \hat{x}'\| \leq \frac{\kappa}{\sqrt{n}} \|\hat{x} - \hat{x}'\|_2 \)

**Proof of Lemma 3:**

1) Assume that \( A = A_1 + A_2 \), where \( A_1 = \text{col}(R^T F(\hat{x}) + c(L + \hat{L}_m) \hat{x}, 0, 0, 0, b) \), \( A_2 = \Phi \varphi \). Since \( \Phi \) is a skew-symmetric matrix, i.e., \( \Psi T = - \Psi \). As a single-valued operator, \( \Phi \) is maximally monotone (Ex.20.30, [32]).

2) By [35] and the definition of \( \Phi \)-induced norm, the \( \Phi \)-induced Lipschitz continuous property of \( A \) is shown as follows:

\[
\|\Psi^{-1} A(\varphi) - \Psi^{-1} A(\varphi')\|_\Psi \leq \frac{1}{s_{\max}(\Psi)} \|\varphi - \varphi'\|_\Psi \]

Thus, \( A \) is maximally monotone since the direct sum of maximally monotone operators is maximally monotone (Proposition 20.23, [32]).

**Proof of Theorem 3:** Suppose Assumption 6 holds, there exists a fixed points set of \( T \). Denote \( \varphi^* \in fix(T) \), \( \varphi \in \varphi^* \).
\(J_{\Phi^{-1}}(\varpi[k] - \Psi^{-1}A\varpi[k]), \varpi[k+1] = \mathcal{T}\varpi[k], \) for \(\varpi[k] \in H\) arbitrary. We can get
\[
\|\varpi[k+1] - \varpi^*\|_\Psi^2 = \|\varpi[k+1] - \varpi^*\|_\Psi^2 + \|\varpi[k] - \varpi^*\|_\Psi^2 + 2\langle \varpi[k+1] - \varpi^*, \varpi[k] - \varpi^* \rangle_{\Psi},
\]
with some \(\varpi^*\) feasible in the Rayleigh Quotient, \(\forall i, i' \in \mathbb{N}, i \neq i', \) the local sequence \(\{\hat{x}_i^j[k]\}_{k \geq 0}\) converges to the corresponding component \(x_{i,j}^*\) in \(x^*\).

\textbf{REFERENCES}

[1] Y. Hollander and J. N. Prashker, “The applicability of non-cooperative game theory in transport analysis,” Transportation, vol. 33, no. 5, pp. 481–496, 2006.
[2] D. E. Charalis and A. D. Panagopoulos, “A survey on game theory applications in wireless networks,” Computer Networks, vol. 54, no. 18, pp. 3421–3430, 2010.
[3] E. Kaplinsky and J. Terso, “Game theory applications in construction engineering and management,” Technological and economic development of economy, vol. 16, no. 2, pp. 348–363, 2010.
[4] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, “Distributed seeking of nash equilibria with applications to mobile sensors networks,” IEEE Transactions on Automatic Control, vol. 57, no. 4, pp. 904–919, 2011.
[5] Y. Ye, G. Hu, and F. L. Lewis, “Nash equilibrium seeking for n-coalition noncooperative games,” Automatica, vol. 95, pp. 266–272, 2018.
[6] X. Zeng, S. Liang, and Y. Hong, “Distributed variational equilibrium seeking of multi-coalition game via variational inequality approach,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 940–945, 2017.
[7] M. Ye, G. Hu, F. L. Lewis, and X. Xie, “A unified strategy for solution seeking in graphical n-coalition noncooperative games,” IEEE Transactions on Automatic Control, vol. 64, no. 11, pp. 4645–4652, 2019.
[8] B. Gharesifard and J. Cortes, “Distributed convergence to nash equilibria in two-network zero-sum games,” Automatica, vol. 49, no. 6, pp. 1683–1692, 2013.
[9] Y. Lou, X. Hong, L. Xie, G. Shi, and K. H. Johansson, “Nash equilibrium computation in subnetwork zero-sum games with switching communications,” IEEE Transactions on Automatic Control, vol. 61, no. 10, pp. 2920–2935, 2015.
[10] T.-J. A. Peng and M. Bourne, “The coexistence of competition and cooperation between networks: implications from two taiwanese healthcare networks,” British Journal of Management, vol. 20, no. 3, pp. 377–400, 2009.
[11] O. Shehory and S. Kraus, “Methods for task allocation via agent coalition formation,” Artificial intelligence, vol. 101, no. 1–2, pp. 165–200, 1998.
[12] X. Zeng, J. Chen, S. Liang, and Y. Hong, “Generalized nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game,” Automatica, vol. 103, pp. 20–26, 2019.
[13] I. S. Pang, G. Scutari, F. Facchinei, and C. Weng, “Distributed power allocation with rate constraints in gaussian parallel interference channels,” IEEE Transactions on Information Theory, vol. 54, no. 8, pp. 3471–3489, 2008.
[14] M. Zhu and E. Frazzoli, “Distributed robust adaptive equilibrium computation for generalized convex games,” Automatica, vol. 63, pp. 82–91, 2016.
[15] J. Ghaderi and R. Srikant, “Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate,” Automatica, vol. 50, no. 12, pp. 3209–3215, 2014.
[16] Z. Deng and X. Nian, “Distributed generalized nash equilibrium seeking algorithm design for aggregative games over weight-balanced digraphs,” IEEE Transactions on Network and Learning Systems, vol. 30, no. 3, pp. 695–706, 2018.
[17] Z. Deng, “Distributed algorithm design for aggregative games of euler-lagrangian systems and its application to smart grids,” IEEE Transactions on Cybernetics, 2021.
[18] L. Pavel, “An extension of duality to a game-theoretic framework,” Automatica, vol. 43, no. 2, pp. 226–237, 2007.
[19] V. Cardellini, V. D. N. Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. F. Presti, and P. Picci, “A game-theoretic approach to computation offloading in mobile cloud computing,” Mathematical Programming, vol. 157, no. 2, pp. 421–449, 2016.
[20] S. Grammatico, “Dynamic control of agents playing aggregative games with coupling constraints,” IEEE Transactions on Automatic Control, vol. 62, no. 9, pp. 4537–4548, 2017.
[21] Y. Liang, W. Wei, and C. Wang, “A generalized nash equilibrium approach for autonomous energy management of residential energy hubs,” IEEE Transactions on Industrial Informatics, vol. 15, no. 11, pp. 5892–5905, 2019.
[22] F. Facchinei and J.-S. Pang, Finite-dimensional variational inequalities and complementarity problems. Springer Science & Business Media, 2007.

[23] F. Facchinei and C. Kanzow, “Generalized nash equilibrium problems,” Annals of Operations Research, vol. 175, no. 1, pp. 177–211, 2010.

[24] G. Arslan, M. F. Demirkol, and S. Yüksel, “On games with coupled constraints,” IEEE Transactions on Automatic Control, vol. 63, no. 2, pp. 388–3901, 2015.

[25] B. Swenson, S. Kar, and J. Xavier, “Empirical centroid fictitious play: An approach for distributed learning in multi-agent games,” IEEE Transactions on Signal Processing, vol. 63, no. 15, pp. 3856–3867, 2015.

[26] J. Lei, U. V. Shanbhag, J.-S. Pang, and S. Sen, “On synchronous, asynchronous, and randomized best-response schemes for stochastic nash games,” Mathematics of Operations Research, vol. 45, no. 1, pp. 157–190, 2020.

[27] J. Zimmermann, T. Tatarenko, V. Willert, and J. Adamy, “Solving leaderless multi-cluster games over directed graphs,” European Journal of Control, vol. 62, pp. 14–21, 2021.

[28] M. Meng and X. Li, “On the linear convergence of distributed nash equilibrium seeking for multi-cluster games under partial-decision information,” arXiv preprint arXiv:2005.06923, 2020.

[29] X. Nian, F. Niu, and Z. Yang, “Distributed nash equilibrium seeking for multicluster game under switching communication topologies,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[30] Y. Pang and G. Hu, “Gradient-free nash equilibrium seeking in multicluster games with uncoordinated constant step-sizes,” arXiv preprint arXiv:2008.13088, 2020.

[31] A. A. Kulkarni and U. V. Shanbhag, “On the variational equilibrium as a refinement of the generalized nash equilibrium,” Automatica, vol. 48, no. 1, pp. 45–55, 2012.

[32] H. H. Bauschke, P. L. Combettes et al., Convex analysis and monotone operator theory in Hilbert spaces. Springer, 2011, vol. 408.

[33] Yi, Peng and Pavel, Lacra, “An operator splitting approach for distributed nash equilibrium computation,” Automatica, vol. 102, pp. 111–121, 2019.

[34] L. Pavel, “Distributed nash equilibrium seeking under partial-decision information over networks via a doubly-augmented operator splitting approach,” IEEE Transactions on Automatic Control, vol. 65, no. 4, pp. 1584–1597, 2019.

[35] Y. Pang and G. Hu, “Gradient-free nash equilibrium seeking in multicluster games with uncoordinated constant step-sizes,” arXiv preprint arXiv:2008.13088, 2020.

[36] M. Bianchi, G. Belgioioso, and S. Grammatico, “Fast generalized nash equilibrium seeking under partial-decision information,” Automatica, vol. 136, p. 110080, 2022.

[37] S. Cui, B. Franci, S. Grammatico, U. V. Shanbhag, and M. Staudigl, “A relaxed-inertial forward-backward-forward algorithm for stochastic generalized nash equilibrium seeking,” arXiv preprint arXiv:2103.13115, 2021.

[38] B. Franci, M. Staudigl, and S. Grammatico, “Distributed forward-backward (half) forward algorithms for generalized nash equilibrium seeking,” in 2020 European Control Conference (ECC). IEEE, 2020, pp. 1274–1279.

[39] J. Koshal, A. Nedec, and U. V. Shanbhag, “Distributed algorithms for aggregative games on graphs,” Operations Research, vol. 64, no. 3, pp. 680–704, 2016.

[40] D. Chao, P. Chakraborty, T. Nishikawa, and A. E. Motter, “Hierarchical power flow control in smart grids: Enhancing rotor angle and frequency stability with demand-side flexibility,” IEEE Transactions on Control of Network Systems, 2021.

[41] W. Saad, Z. Han, H. V. Poor, and T. Basar, “Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications,” IEEE Signal Processing Magazine, vol. 29, no. 5, pp. 86–105, 2012.

[42] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale, “The future renewable electric energy delivery and management (freedm) system: the energy internet,” Proceedings of the IEEE, vol. 99, no. 1, pp. 133–148, 2010.

[43] Q. Sun, Energy Internet and We-Energy. Springer, 2019.

[44] F. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, and J. Wu, “A survey on energy internet: Architecture, approach, and emerging technologies,” IEEE Systems Journal, vol. 12, no. 3, pp. 2403–2416, 2017.