2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Distributed Solution of GNEP over Networks via the Douglas-Rachford Splitting Method

Yuanhanqing Huang and Jianghai Hu

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682862

Abstract— **The aim of this paper is to find the distributed solution of the generalized Nash equilibrium problem (GNEP) for a group of players that can communicate with each other over a connected communication network. Each player tries to minimize a local objective function of its own that may depend on the other players' decisions, and collectively all the players' decisions are subject to some globally shared resource constraints. After reformulating the local optimization problems, we introduce the notion of network Lagrangian and recast the GNEP as the zero finding problem of a properly defined operator. Utilizing the Douglas-Rachford operator splitting method, a distributed algorithm is proposed that requires only local information exchanges between neighboring players in each iteration. The convergence of the proposed algorithm to an exact variational generalized Nash equilibrium is established under two different sets of assumptions. The effectiveness of the proposed algorithm is demonstrated using the example of a Nash-Cournot production game.**

I. Introduction

Due to their numerous practical applications, generalized Nash equilibrium problems (GNEP) for games on networks [1], [2] have received much attention in recent years. Examples include communication networks [3], charge scheduling of electric vehicles [4], formation control [5], and demand management in smart grids [6], etc. In many such problems, a group of players/decision-makers with self-interest aim to optimize their individual objectives through the competition of some shared resources and, due to privacy concerns, may not be willing to disclose their decisions to the general public except for a small number of trusted partners. The goal is then to design distributed protocols for players so that, using information exchanges between neighboring players (trusted partners), the group decisions eventually achieve a generalized Nash equilibrium (GNE).

There has been much existing work on the distributed solution of the GNEP. When player's objectives rely only on the decisions of themselves and their neighbors (i.e. locally dependent objectives) or these players have access to the decisions of each other, i.e., in a full-decision information setting, some elegant methods are proposed in [7] based on the monotone operator theory and the forward-backward (FB) splitting method. To tackle games with non-differentiable objectives, [8] proposes two algorithms based on alternating direction method of multipliers (ADMM), while these players are required to collaboratively solve a group of sub-games at every iteration. The authors of [9] use a proximal-point algorithm (PPA) to compute GNE with non-differentiable yet linearly coupled objectives. Motivated by the practical consideration that players are unwilling to disclose their local information to all other players and the limited amount of communication channels available, there is an increasing interest in developing algorithms that compute GNE in a partial-decision information setting. Under this setting, players on this network only exchange information with their neighbors (trusted players). The author of [10], [11] designs an algorithm by combining the FB splitting method with doubly-augmented information spaces to solve GNEPs with continuously differentiable local objectives. Recently, a preprint [12] comes to our attention, which utilizes a PPA to compute GNEs. The proposed algorithm in [12] allows for non-differentiable local objectives and enjoys a much more desirable convergence speed.

In this paper, we focus on GNEPs with generic objectives, namely, the objectives are globally dependent and have no special structure. In addition, it is assumed that there are some globally shared resource constraints for the group that depends linearly on all players' decisions. We propose a distributed algorithm that requires only communications between neighboring players on an undirected and connected communication graph based on the Douglas-Rachford (DR) splitting method. Further, we provide theoretical guarantees on the exact convergence of the proposed algorithm to a variational GNE under two different sets of assumptions. The effectiveness of the proposed algorithm is demonstrated via a network Nash-Cournot game. Compared to the FB splitting method in [11], our proposed algorithm allows for non-differentiable local objectives and possesses a faster convergence rate. The algorithm proposed in [12] requires each player to solve a constrained optimization problem at each iteration. By contrast, our proposed method can separate this constrained optimization problem into two different steps, i.e., an unconstrained optimization problem and a linear map followed by a projection onto a convex set, which is easier to implement and more computationally efficient in practice. Nevertheless, one drawback is that the amount of information exchanged at each iteration doubles compared to that of [12]. Complete proofs of the main theorems and some intermediate results are omitted in the interest of space; for more details, the interested reader is referred to [13].

Basic Notations: For a set of matrices $\{V_i\}_{i\in\mathcal{S}}$, we let $\mathrm{blkd}(V_1,\ldots,V_{|\mathcal{S}|})$ denote the diagonal concatenation of these matrices, $[V_1,\ldots,V_{|\mathcal{S}|}]$ their horizontal stack, and $[V_1;\cdots;V_{|\mathcal{S}|}]$ their vertical stack. For a set of vectors

This work was supported by the National Science Foundation under Grant No. 2014816.

The authors are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907, USA {huan1282, jianghai}@purdue.edu

$\{v_i\}_{i\in S}$, $[v_i]_{i\in S}$ or $[v_1;\cdots;v_{|S|}]$ denotes their vertical stack. For a vector v and a positive integer i, $[v]_i$ denotes the ith entry of v. Denote $\overline{\mathbb{R}} := \mathbb{R}\cup\{+\infty\}$, $\mathbb{R}_+ := [0,+\infty)$, and $\mathbb{R}_{++} := (0,+\infty)$. $\mathbb{S}^n_+$ (resp. $\mathbb{S}^n_{++}$) represents the set of all $n\times n$ symmetric positive semi-definite (resp. definite) matrices. The notation $\iota_{\mathcal{S}}(x)$ is defined to be the indicator function of a set $\mathcal{S}$, i.e., if $x\in\mathcal{S}$, then $\iota_{\mathcal{S}}(x)=0$; otherwise, $\iota_{\mathcal{S}}(x)=+\infty$. We let $N_S(x)$ denote the normal cone to the set $S\subseteq\mathbb{R}^n$ at the point x: if $x\in S$, then $N_S(x) := \{u\in\mathbb{R}^n \mid \sup_{z\in S}\langle u, z-x\rangle \le 0\}$; otherwise, $N_S(x) := \varnothing$. If $S\in\mathbb{R}^n$ is a closed and convex set, the map $\mathrm{Pj}_S : \mathbb{R}^n \to S$ denotes the projection onto S, i.e., $\mathrm{Pj}_S(x) := \operatorname{argmin}_{v\in S}\|v-x\|$. We use $\rightrightarrows$ to indicate a point-to-set map. For an operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, $\mathrm{Zer}(T) := \{x\in\mathbb{R}^n \mid Tx \ni \mathbf{0}\}$ and $\mathrm{Fix}(T) := \{x\in\mathbb{R}^n \mid Tx\ni x\}$ denote its zero set and fixed point set, respectively. We denote $\mathrm{dom}(T)$ the domain of the operator T and $\mathrm{gra}(T)$ the graph of it.

II. Problem Formulation

A. Game Formulation and GNE

We consider a set of players indexed by $\mathcal{N} := \{1,\dots,N\}$, each of which decides on its decision variables $x_i\in\mathcal{X}_i$ and optimizes its local objective function $J_i(x_i;x_{-i})$. Here, $\mathcal{X}_i\subseteq\mathbb{R}^{n_i}$ denotes player i's local feasible set. Let the index set of players except player i be defined by $\mathcal{N}_{-i}$. The vector $x_{-i}\in\mathbb{R}^{n_{-i}}$ represents the vertical stack of the other players' decision variables, with $n_{-i} := \sum_{j\in\mathcal{N}_{-i}} n_j$. Formally, given the decision of the other players, the player i aims to solve a local optimization problem as follows:

$$\begin{cases}\underset{x_i\in\mathcal{X}_i}{\text{minimize}} & J_i(x_i;x_{-i})\\ \text{subject to} & A_ix_i \le c-\sum_{j\in\mathcal{N}_{-i}}A_jx_j\end{cases}, \tag{1}$$

where $A_i\in\mathbb{R}^{m\times n_i}$, m is the number of the (global) affine coupling constraints, and $c\in\mathbb{R}^m$ is a constant vector. Denote the vertical stack of all decision variables by $x := [x_1;\cdots;x_N]$. The feasible set of the collective decision vector x, is given by:

$$\tilde{\mathcal{X}} := \mathcal{X}\cap\{x\in\mathbb{R}^n | Ax-c\le\mathbf{0}\}, \tag{2}$$

where $\mathcal{X} := \prod_{i=1}^N\mathcal{X}_i$, $A := [A_1,A_2,\dots,A_N]$, and $n = \sum_{i\in\mathcal{N}} n_i$. The feasible decision set of each player $i\in\mathcal{N}$ is characterized by the set-valued mapping $\mathcal{X}_i : \mathbb{R}^{n_{-i}}\rightrightarrows\mathbb{R}^{n_i}$, which is defined as:

$$\tilde{\mathcal{X}}_i(x_{-i}) := \{x_i\in\mathcal{X}_i | A_ix_i\le c-\textstyle\sum_{j\in\mathcal{N}_{-i}}A_jx_j\}. \tag{3}$$

Assumption 1. *(Objective Functions) For each $i\in\mathcal{N}$, the objective function J_i is proper and continuous. In addition, $J_i(x_i;x_{-i})$ is a convex function w.r.t. x_i, given any fixed x_{-i}.*

Assumption 2. *(Feasible Sets) Each local feasible set $\mathcal{X}_i$ is nonempty, closed, and convex. The collective feasible set $\tilde{\mathcal{X}}$ is nonempty, and the Mangasarian–Fromovitz constraint qualification (MFCQ) holds [14, Sec. 3.2][15, Sec. 12.2.3].*

A generalized Nash equilibrium (GNE) for the game (1) is a joint decision vector x^* such that, for each player i, $x_i^*\in\tilde{\mathcal{X}}_i(x_{-i}^*)$ is a global minimizer of the local optimization problem described in (1) [15, Sec. 12.2]. Under the suitable conditions described in Assumptions 1 and 2 [1][15, Sec. 12.2], a solution to the GNEP is equivalent to that of the Karust-Kuhn-Tucker (KKT) conditions given as follows:

$$\begin{aligned}&\mathbf{0}\in\partial_{x_i}J_i(x_i;x_{-i})+A_i^T\lambda_i+N_{\mathcal{X}_i}(x_i)\\ &\mathbf{0}\in-(Ax-c)+N_{\mathbb{R}^m_+}(\lambda_i),\end{aligned} \tag{4}$$

for each $i\in\mathcal{N}$, where λ_i is the Lagrangian multiplier for the inequality constraints for the local problem (1) of player i. In this paper, we focus on the variational generalized Nash Equilibria (v-GNEs), which are a subset of GNEs [2]. The v-GNEs are solutions to the inclusions in (4) with all $\{\lambda_i\}_{i\in\mathcal{N}}$ on consensus ($\lambda_1=\cdots=\lambda_N$). The KKT conditions for a v-GNE are:

$$\begin{aligned}&\mathbf{0}\in\partial_{x_i}J_i(x_i;x_{-i})+A_i^T\lambda+N_{\mathcal{X}_i}(x_i),\forall\, i\in\mathcal{N}\\ &\mathbf{0}\in-(Ax-c)+N_{\mathbb{R}^m_+}(\lambda).\end{aligned} \tag{5}$$

Another way to characterize v-GNE with non-smooth objectives is via generalized variational inequalities (GVI) [16]. Define the pseudogradient/game Jacobian $F : \mathbb{R}^n\rightrightarrows\mathbb{R}^n$ as:

$$F : x\mapsto[\partial_{x_i}J_i(x_i;x_{-i})]_{i\in\mathcal{N}}. \tag{6}$$

As has been shown in [15, Prop. 12.3], if $\mathrm{GVI}(\tilde{\mathcal{X}},F)$ admits a solution, this solution is a v-GNE of the game defined in (1).

Remark 1. *We focus on v-GNE since we can leverage the rich body of (existence and uniqueness) theory and tools developed for solving VIs [14][15, Ch. 12], and by keeping all $\{\lambda_i\}_{i\in\mathcal{N}}$ on consensus, v-GNE possesses desirable properties such as "economic fairness" and "better social stability/sensitivity" [17].*

B. Networked Game Formulation

To enable the distributed computation of v-GNE, we consider an underlying communication graph $\mathcal{G}=(\mathcal{N}_g,\mathcal{E}_g)$, where players can communicate with their neighbors through arbitrators on the directed edges. The node set $\mathcal{N}_g$ represents the set of all the players, and $\mathcal{E}_g\subseteq\mathcal{N}_g\times\mathcal{N}_g$ is the set of directed edges. The cardinalities $|\mathcal{N}_g|$ and $|\mathcal{E}_g|$ are denoted by N_g and E_g. In this case, $\mathcal{N}_g=\mathcal{N}$ and $N_g=N$. We use (i,j) to denote a directed edge having node/player i as its tail and node/player j as its head. Let $\mathcal{N}_i$ denote the set of immediate neighbors of player i who can directly communicate with it, $\mathcal{N}_i^+ := \{j\in\mathcal{N} \mid (j,i)\in\mathcal{E}_g\}$ the set of in-neighbors of player i, and $\mathcal{N}_i^- := \{j\in\mathcal{N} \mid (i,j)\in\mathcal{E}_g\}$ the set of out-neighbors of player i. Note that although the multipliers we are going to introduce are defined in a directed fashion, we assume each node can send messages to both its in- and out-neighbors and hence the communication graph $\mathcal{G}$ should satisfy the following assumption.

Assumption 3. *(Communicability) The underlying communication graph $\mathcal{G}=(\mathcal{N}_g,\mathcal{E}_g)$ is undirected and connected. Besides, it has no self-loops.*

Our goal is to convert the centralized GNEP in (1) into the zero-finding problem of a certain operator that can be carried out distributedly over the graph $\mathcal{G}$. To construct this operator, in the rest of this section, we will first derive the network Lagrangian. Let each player extend its decision space to maintain the local estimates of the other players' decision vectors. Each player i now has the augmented decision variable $y_i \in \mathbb{R}^n$, which consists of its local decision $y_i^i \in \mathbb{R}^{n_i}$ and the local estimate $y_i^j \in \mathbb{R}^{n_j}$ of player j's decision for each $j \in \mathcal{N}_{-i}$. Let $y_i^{-i} \in \mathbb{R}^{n_{-i}}$ denote the vertical stack of y_i^j for all $j \in \mathcal{N}_{-i}$ in a prespecified order. Denote $n_{<i} = \sum_{j\in\mathcal{N}, j<i} n_j$ and $n_{>i} = \sum_{j\in\mathcal{N}, j>i} n_j$. The extended feasible set $\hat{\mathcal{X}}$ is defined as $\hat{\mathcal{X}} := \hat{\mathcal{X}}_1 \times \hat{\mathcal{X}}_2 \times \cdots \times \hat{\mathcal{X}}_N$, where the feasible set of each y_i is defined as $\hat{\mathcal{X}}_i := \mathbb{R}^{n_{<i}} \times \mathcal{X}_i \times \mathbb{R}^{n_{>i}}$. Notice that $\mathbb{R}^{n_{<i}} \times N_{\mathcal{X}_i}(y_i^i) \times \mathbb{R}^{n_{>i}} = N_{\hat{\mathcal{X}}_i}(y_i)$.

With the introduction of the local estimates of the other players' decisions, the preceding v-GNE problem of each player i can be reformulated as minimizing its local objective $J_i(y_i^i; y_i^{-i})$ while subjecting to the global resource constraints and the consensus constraints among $\{y_i\}_{i\in\mathcal{N}}$. We then sum the individual Lagrangians of the local optimization problems and obtain the following network Lagrangian:

$$\begin{aligned}\mathcal{L}_{\text{net}}(\{y_i\}, \{\mu_{ji}\}, \lambda) := \textstyle\sum_{i\in\mathcal{N}} \big(J_i(y_i^i; x_{-i})_{|x_{-i}=y_i^{-i}} + \iota_{\mathcal{X}_i}(y_i^i)\big) \\ + \lambda^T \big(\textstyle\sum_{i\in\mathcal{N}} A_i y_i^i - c_i\big) + \textstyle\sum_{(j,i)\in\mathcal{E}_g} \mu_{ji}^T (y_i - y_j),\end{aligned} \tag{7}$$

where $\{c_i\}$ is a set of arbitrary vectors satisfying $\sum_{i\in\mathcal{N}} c_i = c$; each μ_{ji} is a Lagrange multiplier ensuring the consensus between y_j and y_i. For convenience, we shall write $\{y_i\}$ in replacement of the more cumbersome notation $\{y_i\}_{i\in\mathcal{N}}$ and similarly for other variables on nodes and edges (e.g. $\{\mu_{ji}\}$ in replacement of $\{\mu_{ji}\}_{(j,i)\in\mathcal{E}_g}$), unless otherwise specified.

Notice that the computation of λ in (7) still requires the direct engagement of all players. We let each player i keep a local estimate λ_i to distribute the computation of λ along with the consensus constraints w.r.t. these local estimates. To facilitate the convergence, we introduce some second-order penalty terms for consensus constraints. The modified augmented network Lagrangian is given as follows:

$$\begin{aligned}\mathcal{L}_{\text{net}}^* := \textstyle\sum_{i\in\mathcal{N}} \big(J_i(y_i^i; x_{-i})_{|x_{-i}=y_i^{-i}} + \iota_{\mathcal{X}_i}(y_i^i) + \lambda_i^T (A_i y_i^i - c_i)\big) \\ + \textstyle\sum_{(j,i)\in\mathcal{E}_g} \big(\mu_{ji}^T (y_i - y_j) - z_{ji}^T (\lambda_i - \lambda_j)\big) \\ + \textstyle\sum_{(j,i)\in\mathcal{E}_g} \big(\frac{\rho_\mu}{2}\|y_i - y_j\|^2 - \frac{\rho_z}{2}\|\lambda_i - \lambda_j\|^2\big),\end{aligned} \tag{8}$$

where $\{z_{ji}\}$ is the set of Lagrange multipliers ensuring the consensus among $\{\lambda_i\}$. We let two positive constant parameters ρ_μ and ρ_z control the weights of the second order penalty terms, whose lower bounds are to be determined later on.

In the augmented network Lagrangian, each player i's objective function $J_i(y_i^i; y_i^{-i})$ is only optimized over y_i^i, while its local estimates y_i^{-i} of the other players are updated to satisfy the global and consensus constraints. Accordingly, the extended pseudogradient $\mathcal{F} : \mathbb{R}^{nN} \rightrightarrows \mathbb{R}^n$ is the set-valued operator defined as

$$\mathcal{F} : \boldsymbol{y} \mapsto [\partial_{y_i^i} J_i(y_i^i; y_i^{-i})]_{i\in\mathcal{N}}, \tag{9}$$

where $\boldsymbol{y} := [y_1; \cdots; y_N] \in \mathbb{R}^{nN}$. For $\boldsymbol{y}$ having $\{y_i\}$ on consensus, $F(y_i) = \mathcal{F}(\boldsymbol{y})$. To incorporate the extended pseudogradient $\mathcal{F}$ into a fixed point iteration, we introduce the individual selection matrices $\{\mathcal{R}_i\}_{i\in\mathcal{N}}$ and their diagonal concatenation $\mathcal{R} \in \mathbb{R}^{n\times nN}$ given as below:

$$\mathcal{R}_i = [\mathbf{0}_{n_i\times n_{<i}}, \mathbf{I}_{n_i}, \mathbf{0}_{n_i\times n_{>i}}], \mathcal{R} = \text{blkd}(\mathcal{R}_1, \ldots, \mathcal{R}_N). \tag{10}$$

Notice that $y_i^i = \mathcal{R}_i y_i$ and $\mathcal{R}_i \mathcal{R}_i^T = I_{n_i}$.

III. Distributed Algorithm with the DR Splitting

In this section, we recast the GNEP as the zero-finding problem for a properly defined operator. We derive the analytical updating steps to solve for the zeros by leveraging the Douglas-Rachford operator splitting method and constructing a design matrix. The analysis of monotonicity and convergence will be discussed in Section IV.

A. Zero-finding Problem

To solve the distributed GNE seeking problem, we need to find the stationary points of the augmented network Lagrangian (8). By taking the partial derivative of $\mathcal{L}_{\text{net}}^*$ w.r.t. each variable and reversing the sign of the rows corresponding to λ_i and μ_i, we can obtain the following set-valued operator $\mathcal{T}$:

$$\mathcal{T} : \begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{\lambda}\\ \boldsymbol{\mu}\\ \boldsymbol{z}\end{bmatrix} \mapsto \begin{bmatrix}\mathcal{R}^T(\mathcal{F}(\boldsymbol{y}) + \Lambda^T\boldsymbol{\lambda}) + B_n\boldsymbol{\mu} + \rho_\mu L_n \boldsymbol{y} + N_{\hat{\mathcal{X}}}(\boldsymbol{y})\\ N_{\mathbb{R}_+^{mN}}(\boldsymbol{\lambda}) - \Lambda\mathcal{R}\boldsymbol{y} + \boldsymbol{c} + B_m\boldsymbol{z} + \rho_z L_m\boldsymbol{\lambda}\\ -B_n^T\cdot\boldsymbol{y}\\ -B_m^T\cdot\boldsymbol{\lambda}\end{bmatrix}, \tag{11}$$

where Λ is the diagonal concatenation of $\{A_i\}$, i.e., $\Lambda := \text{blkd}(A_1, \ldots, A_N)$; $\boldsymbol{c}$ is the vertical stack of $\{c_i\}$; $B_n := (B \otimes I_n)$, $L_n := (L \otimes I_n)$, $B_m := (B \otimes I_m)$, $L_m := (L \otimes I_m)$, B and L are the incidence matrix and Laplacian matrix of the underlying communication graph, respectively, with $L = B \cdot B^T$; $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and $\boldsymbol{z}$ are the stack vectors of $\{\lambda_i\}$, $\{\mu_{ji}\}$, and $\{z_{ji}\}$, respectively; ψ denotes the stack of the former primal and dual variables, i.e., $\psi := [\boldsymbol{y}; \boldsymbol{\lambda}; \boldsymbol{\mu}; \boldsymbol{z}]$.

Theorem 1. *Suppose Assumptions 1-3 hold, and there exists* $\psi^* := [\boldsymbol{y}^*; \boldsymbol{\lambda}^*; \boldsymbol{\mu}^*; \boldsymbol{z}^*] \in Zer(\mathcal{T})$. *Then* $\boldsymbol{y}^* = \mathbf{1}_N \otimes y^*$, $\boldsymbol{\lambda}^* = \mathbf{1}_N \otimes \lambda^*$, *and* (y^*, λ^*) *satisfies the KKT conditions* (5) *for v-GNE with x replaced with* y^*. *Furthermore, for a solution* $(y^\dagger, \lambda^\dagger)$ *of the KKT problem in* (5), *there exist* $\boldsymbol{\mu}^\dagger$ *and* $\boldsymbol{z}^\dagger$ *such that* $\psi^\dagger := [\mathbf{1}_N \otimes y^\dagger; \mathbf{1}_N \otimes \lambda^\dagger; \boldsymbol{\mu}^\dagger; \boldsymbol{z}^\dagger] \in Zer(\mathcal{T})$.

Theorem 1 implies that we can convert the solution of v-GNE of the original GNEP into the zero-finding problem of the operator $\mathcal{T}$ in (11). In the next subsection, we will propose a candidate algorithm to solve the latter problem.

B. Operator Splitting

Given a generic set-valued operator $T : \mathbb{R}^\ell \rightrightarrows \mathbb{R}^\ell$, T is monotone if, for any $(x, u) \in \text{gra}(T)$ and $(x', u') \in \text{gra}(T)$, $\langle x - x', u - u'\rangle \geq 0$. If T is maximally monotone, a point in $\text{Zer}(T)$ could in principle be determined through the proximal-point algorithm, namely, the fixed point iteration using its resolvent operator $J_T : \mathbb{R}^\ell \to \mathbb{R}^\ell$ defined as $J_T := (I + T)^{-1}$ [18, Thm. 23.41]. However, it is often infeasible or computationally intensive to evaluate the resolvent in a distributed manner for operators arising in network problems

such as the one in (11). For this purpose, we consider the Douglas-Rachford (DR) splitting technique, where $T = A+B$ is split into two maximally monotone operators A and B whose resolvents can be more conveniently and efficiently computed. The resolvent and reflected resolvent of A are defined as $J_A \coloneqq (I+A)^{-1}$ and $R_A \coloneqq 2J_A - I$, respectively. As A is maximally monotone, J_A is firmly nonexpansive and R_A is non-expansive, i.e., for any x and x', $\|J_A x - J_A x'\|^2 + \|(I - J_A)x - (I - J_A)x'\|^2 \leq \|x - x'\|^2$ and $\|R_A x - R_A x'\| \leq \|x - x'\|^2$ (see [18, Cor. 23.11]). Similar properties hold for B.

The DR splitting algorithm can be viewed as a special case of the Krasnosel'skii-Mann algorithm. Given a nonexpansive operator Q which has a nonempty fixed point set $\mathrm{Fix}(Q)$, the Krasnosel'skii-Mann algorithm suggests the iteration:

$$x^{(k+1)} = x^{(k)} + \gamma^{(k)}(Qx^{(k)} - x^{(k)}). \tag{12}$$

If $(\gamma^{(k)})_{k\in\mathbb{N}}$ is a sequence in $[0,1]$ satisfying $\sum_{k\in\mathbb{N}} \gamma^{(k)}(1-\gamma^{(k)}) = \infty$, then $(x^{(k)})_{k\in\mathbb{N}}$ converges to a point in $\mathrm{Fix}(Q)$[18, Sec. 5.2]. In the DR algorithm, Q in (12) is set to be $R_B R_A$. Then, the iteration in (12) converges to some $x^* \in \mathrm{Fix}(R_B R_A)$. It follows from [18, Sec. 26.3] that $J_A(x^*) \in \mathrm{Zer}(A+B) = \mathrm{Zer}(T)$. The detailed steps of the Douglas-Rachford splitting are given below:

$$\begin{aligned}
&\text{Calculate } J_A: x^{(k+1)} \coloneqq J_A(\tilde{x}^{(k)});\\
&\text{R-R updates}: \hat{x}^{(k+1)} \coloneqq 2\cdot x^{(k+1)} - \tilde{x}^{(k)};\\
&\text{Calculate } J_B: \bar{x}^{(k+1)} \coloneqq J_B(\hat{x}^{(k+1)});\\
&\text{K-M updates}: \tilde{x}^{(k+1)} \coloneqq \tilde{x}^{(k)} + 2\gamma^{(k)}(\bar{x}^{(k+1)} - x^{(k+1)}).
\end{aligned} \tag{13}$$

Here, R-R updates stand for evaluating the reflected resolvent of A and K-M updates the Krasnoselskij-Mann updates.

Now we focus on the operator $\mathcal{T}$ defined in (11) and split it into two operators $\mathcal{A}$ and $\mathcal{B}$ defined as follows:

$$\mathcal{A}: \psi \mapsto (D + \mathcal{A}_y)\psi \text{ and } \mathcal{B}: \psi \mapsto (D + \mathcal{B}_y)\psi \tag{14}$$

with D, $\mathcal{A}_y$, and $\mathcal{B}_y$ written as

$$D = \begin{bmatrix} \frac{\rho_\mu}{2}L_n & \frac{1}{2}(\Lambda\mathcal{R})^T & \frac{1}{2}B_n & 0\\ -\frac{1}{2}\Lambda\mathcal{R} & \frac{\rho_z}{2}L_m & 0 & \frac{1}{2}B_m\\ -\frac{1}{2}B_n^T & 0 & 0 & 0\\ 0 & -\frac{1}{2}B_m^T & 0 & 0 \end{bmatrix}, \tag{15}$$

$$\mathcal{A}_y: \psi \mapsto \begin{bmatrix} \mathcal{R}^T\mathcal{F}(\boldsymbol{y})\\ \boldsymbol{c}\\ 0\\ 0 \end{bmatrix}, \text{and } \mathcal{B}_y: \psi \mapsto \begin{bmatrix} N_{\hat{\mathcal{X}}}(\boldsymbol{y})\\ N_{\mathbb{R}_+^{mN}}(\boldsymbol{\lambda})\\ 0\\ 0 \end{bmatrix}. \tag{16}$$

Given the split operators $\mathcal{A}$ and $\mathcal{B}$, to allow for the distributed evaluation of the resolvents $J_{\mathcal{A}}$ and $J_{\mathcal{B}}$, a positive definite matrix Φ is designed. It is constructed in a way such that $D+\Phi$ is lower or upper triangular. One choice is

$$\Phi = \begin{bmatrix} \boldsymbol{\tau}_1^{-1} - \frac{\rho_\mu}{2}L_n & -\frac{1}{2}(\Lambda\mathcal{R})^T & -\frac{1}{2}B_n & 0\\ -\frac{1}{2}\Lambda\mathcal{R} & \boldsymbol{\tau}_2^{-1} - \frac{\rho_z}{2}L_m & 0 & -\frac{1}{2}B_m\\ -\frac{1}{2}B_n^T & 0 & \boldsymbol{\tau}_3^{-1} & 0\\ 0 & -\frac{1}{2}B_m^T & 0 & \boldsymbol{\tau}_4^{-1} \end{bmatrix}, \tag{17}$$

where $\boldsymbol{\tau}_1 \coloneqq \mathrm{blkd}(\tau_{11}I_n, \ldots, \tau_{1N}I_n)$ with $\tau_{1i} > 0$ for $i \in \mathcal{N}$; similarly for $\boldsymbol{\tau}_2$, $\boldsymbol{\tau}_3$ and $\boldsymbol{\tau}_4$.

Let $\tilde{\mathcal{A}}$ denote $\Phi^{-1}\mathcal{A}$ and $\tilde{\mathcal{B}}$ denote $\Phi^{-1}\mathcal{B}$. Note that $\mathbf{0} \in (\mathcal{A}+\mathcal{B})\psi$ if and only if $\mathbf{0} \in (\tilde{\mathcal{A}}+\tilde{\mathcal{B}})\psi$. The Douglas-Rachford splitting method (13) is then leveraged to find a zero of $\tilde{\mathcal{A}}+\tilde{\mathcal{B}}$. A detailed version with explicit operations at each node and edge is given in Algorithm 1, with the first player for-loop and the first edge for-loop implementing the computation of $J_{\tilde{\mathcal{A}}}$ and its reflected resolvent $R_{\tilde{\mathcal{A}}}$, while the second player and edge for-loops the computation of $J_{\tilde{\mathcal{B}}}$.

In the pseudocode, for notational convenience, we define $\tilde{y}_{iL}^{-i(k)} \coloneqq \sum_{j\in\mathcal{N}_i}(\tilde{y}_i^{-i(k)} - \tilde{y}_j^{-i(k)})$, and similarly for $\tilde{y}_{iL}^{i(k)}$, $\tilde{\lambda}_{iL}^{(k)}$, $\hat{y}_{iL}^{i(k+1)}$, and $\hat{\lambda}_{iL}^{(k+1)}$; define $\tilde{\mu}_{iB}^{-i(k)} \coloneqq \sum_{j\in\mathcal{N}_i^+}\tilde{\mu}_{ji}^{-i(k)} - \sum_{j\in\mathcal{N}_i^-}\tilde{\mu}_{ij}^{-i(k)}$, and similarly for $\tilde{\mu}_{iB}^{i(k)}$, $\tilde{z}_{iB}^{(k)}$, $\hat{\mu}_{iB}^{(k+1)}$, and $\hat{z}_{iB}^{(k+1)}$; define $\hat{y}_{ji}^{(k+1)} \coloneqq \hat{y}_i^{(k+1)} - \hat{y}_j^{(k+1)}$, and similarly for $\hat{\lambda}_{ji}^{(k+1)}$, $\bar{y}_{ji}^{(k+1)}$, and $\bar{\lambda}_{ji}^{(k+1)}$. The computational workload is concentrated on the update of local decision y_i^i inside the first player for-loop described by an unconstrained minimization problem and the update of $\bar{y}_i^i$ inside the second player for-loop described by a linear transformation followed by a projection onto the local feasible set, while the other updates are computationally trivial. The result $\{y_i^{(k)}\}$ returned by Algorithm 1 will be proved in the next section to converge to a v-GNE of the problem (1) under some suitable conditions.

Algorithm 1: Distributed GNE Seeking

Initialize: $\{\tilde{y}_i^{(0)}\}, \{\tilde{\lambda}_i^{(0)}\}, \{\tilde{\mu}_{ji}^{(0)}\}, \{\tilde{z}_{ji}^{(0)}\}$;
Iterate until convergence:
for *player* $i \in \mathcal{N}$ **do**
  $y_i^{-i(k+1)} = \tilde{y}_i^{-i(k)} - \frac{\tau_{1i}}{2}(\rho_\mu \tilde{y}_{iL}^{-i(k)} + \tilde{\mu}_{iB}^{-i(k)})$;
  $y_i^{i(k+1)} = \mathrm{argmin}_{v_i\in\mathbb{R}^{n_i}}[J_i(v_i; y_i^{-i(k+1)})$
    $+\frac{1}{2}((A_i^T\tilde{\lambda}_i^{(k)} + \rho_\mu\tilde{y}_{iL}^{i(k)} + \tilde{\mu}_{iB}^{i(k)})^T v_i + \frac{1}{\tau_{1i}}\|v_i - \tilde{y}_i^{i(k)}\|^2)]$;
  $\lambda_i^{(k+1)} = \tilde{\lambda}_i^{(k)} + \tau_{2i}(A_i(y_i^{i(k+1)} - \frac{1}{2}\tilde{y}_i^{i(k)}) - \frac{\rho_z}{2}\tilde{\lambda}_{iL}^{(k)} - \frac{1}{2}\tilde{z}_{iB}^{(k)} - c_i)$;
  $\hat{y}_i^{(k+1)} = 2y_i^{(k+1)} - \tilde{y}_i^{(k)}$, $\hat{\lambda}_i^{(k+1)} = 2\lambda_i^{(k+1)} - \tilde{\lambda}_i^{(k)}$;
end
for *edge* $(j,i) \in \mathcal{E}_g$ **do**
  $\mu_{ji}^{(k+1)} = \tilde{\mu}_{ji}^{(k)} + \frac{\tau_{3i}}{2}\hat{y}_{ji}^{(k+1)}$, $\hat{\mu}_{ji}^{(k+1)} = 2\mu_{ji}^{(k+1)} - \tilde{\mu}_{ji}^{(k)}$;
  $z_{ji}^{(k+1)} = \tilde{z}_{ji}^{(k)} + \frac{\tau_{4i}}{2}\hat{\lambda}_{ji}^{(k+1)}$, $\hat{z}_{ji}^{(k+1)} = 2z_{ji}^{(k+1)} - \tilde{z}_{ji}^{(k)}$;
end
for *player* $i \in \mathcal{N}$ **do**
  $\bar{y}_i^{(k+1)} = \mathrm{Pj}_{\hat{\mathcal{X}}_i}[\hat{y}_i^{(k+1)} - \frac{\tau_{1i}}{2}(\mathcal{R}_i^T A_i^T\hat{\lambda}_i^{(k+1)} + \rho_\mu\hat{y}_{iL}^{(k+1)} + \hat{\mu}_{iB}^{(k+1)})]$;
  $\bar{\lambda}_i^{(k+1)} = \mathrm{Pj}_{\mathbb{R}_+^m}[\hat{\lambda}_i^{(k+1)} + \tau_{2i}(A_i(\bar{y}_i^{i(k+1)} - \frac{1}{2}\hat{y}_i^{i(k)})$
    $-\frac{\rho_z}{2}\hat{\lambda}_{iL}^{(k+1)} - \frac{1}{2}\hat{z}_{iB}^{(k+1)})]$;
end
for *edge* $(j,i) \in \mathcal{E}_g$ **do**
  $\bar{\mu}_{ji}^{(k+1)} = \hat{\mu}_{ji}^{(k+1)} + \tau_{3i}(\bar{y}_{ji}^{(k+1)} - \frac{1}{2}\hat{y}_{ji}^{(k+1)})$;
  $\bar{z}_{ji}^{(k+1)} = \hat{z}_{ji}^{(k+1)} + \tau_{4i}(\bar{\lambda}_{ji}^{(k+1)} - \frac{1}{2}\hat{\lambda}_{ji}^{(k+1)})$;
end
K-M updates: $\tilde{\psi}^{(k+1)} = \tilde{\psi}^{(k)} + 2\gamma^{(k)}(\bar{\psi}^{(k+1)} - \psi^{(k+1)})$;
Return: $\{y_i^{(k)}\}$.

Before the first (resp. second) player for-loop of the k-th iteration, each player i receives the multipliers $\tilde{\mu}_{ji}^{(k)}$ and $\tilde{z}_{ji}^{(k)}$ (resp. $\hat{\mu}_{ji}^{(k+1)}$ and $\hat{z}_{ji}^{(k+1)}$) from each in-edge (j,i), $\tilde{\mu}_{ij}^{(k)}$ and $\tilde{z}_{ij}^{(k)}$ (resp. $\hat{\mu}_{ij}^{(k+1)}$ and $\hat{z}_{ij}^{(k+1)}$) from each out-edge (i,j), and $\tilde{y}_j^{(k)}$ along with $\tilde{\lambda}_j^{(k)}$ (resp. $\hat{y}_j^{(k+1)}$ along with $\hat{\lambda}_j^{(k+1)}$) from each neighbor j. At the end of the first player for-loop, each player

i sends $\hat{y}_i^{(k+1)}$ and $\hat{\lambda}_i^{(k+1)}$ to its incident edges and neighboring players. In contrast, at the end of the second player for-loop, each player i sends $\bar{y}_i^{(k+1)}$ and $\bar{\lambda}_i^{(k+1)}$ to its incident edges, while sending $\tilde{y}_i^{(k+1)}$ and $\tilde{\lambda}_i^{(k+1)}$ to its neighboring players for the computation in the next iteration. Before the first (resp. second) edge for-loop, each edge (j,i) receives $\hat{y}_i^{(k+1)}$ and $\hat{\lambda}_i^{(k+1)}$ (resp. $\bar{y}_i^{(k+1)}$ and $\bar{\lambda}_i^{(k+1)}$) from its head i, and $\hat{y}_j^{(k+1)}$ and $\hat{\lambda}_j^{(k+1)}$ (resp. $\bar{y}_j^{(k+1)}$ and $\bar{\lambda}_j^{(k+1)}$) from its tail j; after the computation is completed, the edge (j,i) then sends $\hat{\mu}_{ji}^{(k+1)}$ and $\hat{z}_{ji}^{(k+1)}$ (resp. $\tilde{\mu}_{ji}^{(k+1)}$ and $\tilde{z}_{ji}^{(k+1)}$) to its head i and tail j. At each iteration, player i will receive $(2n+2m)$ variables from each neighbor and send the same amount back to it. At the same time, player i will also receive and send $(2n+2m)$ variables from and to the arbitrator on each incident edge.

IV. Convergence Results

To analyze the convergence properties of the Algorithm 1, we make two parallel assumptions, either of which can guarantee the convergence to a v-GNE. Moreover, depending on the specific game scenario, either one could be less restrictive than the other.

Assumption 4. *(Convergence Condition A) The GNEP admits at least one v-GNE, and the operator $\mathcal{R}^T\mathcal{F} + \frac{\rho_\mu}{2}L_n$ is maximally monotone.*

Assumption 5. *(Convergence Condition B) The pseudo-gradient operator F is strongly monotone and Lipschitz continuous, i.e., there exist $\eta > 0$ and $\theta_1 > 0$, such that $\forall x, x' \in \mathbb{R}^n$, $\langle x - x', F(x) - F(x')\rangle \geq \eta\|x - x'\|^2$ and $\|F(x) - F(x')\| \leq \theta_1\|x - x'\|$. The operator $\mathcal{R}^T\mathcal{F}$ is Lipschitz continuous, i.e., there exists $\theta_2 > 0$, such that $\forall \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{nN}$, $\|\mathcal{F}(\boldsymbol{y}) - \mathcal{F}(\boldsymbol{y}')\| \leq \theta_2\|\boldsymbol{y} - \boldsymbol{y}'\|$.*

Remark 2. *The Lipschitz continuity of $\mathcal{F}$ and that of F implies each other. To see the forward direction, $\forall \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{nN}$ and assuming their local estimates are on consensus ($y_i = y_j$ and $y_i' = y_j'$ for all i, j), we have $\|F(y_i) - F(y_i')\| = \|\mathcal{F}(\boldsymbol{y}) - \mathcal{F}(\boldsymbol{y}')\| \leq \theta_2\|\boldsymbol{y} - \boldsymbol{y}'\| = \sqrt{N}\theta_2\|y_i - y_i'\|$. However, θ_1 could provide a tighter Lipschitz constant than $\sqrt{N}\theta_2$, i.e., $\|F(y_i) - F(y_i')\| \leq \theta_1\|y_i - y_i'\| \leq \sqrt{N}\theta_2\|y_i - y_i'\|$. The other direction has been analyzed in [19, Lemma 3] which also suggests $\eta \leq \theta_2 \leq \theta_1$.*

From the monotone operator theory perspective, Assumption 4 is the least restricted when it comes to guaranteeing the convergence of the proposed algorithm. However, verifying the fulfillment of this assumption is cumbersome and all too often can only be done numerically (see [20, Sec. 4.2.3] for example). From a practical perspective, Assumption 5, which is commonly assumed in the literature [11], [12], is much more accessible and can often be reduced to verifying the properties of F [19, Lemma 3]. Nevertheless, we still keep Assumption 4 since it includes a considerable class of games that fail to satisfy Assumption 5. For example, consider a two-player game where each player i has its objective defined by $J_i(x_i; x_{-i}) := x_i^3$ with $\mathcal{X}_i := \mathbb{R}_+$. Obviously, this game fulfills Assumption 4 but not Assumption 5. Another example would be that $J_i(x_i; x_{-i})$ is continuous yet non-differentiable in x_i for any fixed x_{-i}.

Lemma 1. *Suppose $\{\tau_1, \tau_2, \tau_3, \tau_4\}$ in the design matrix Φ satisfy the following inequalities: $\tau_{1i}^{-1} > \frac{1}{2}\|A_i\|_1 + (\frac{1}{2} + \rho_\mu)d_i$, $\tau_{2i}^{-1} > \frac{1}{2}\|A_i\|_\infty + (\frac{1}{2} + \rho_z)d_i, \forall i \in \mathcal{N}$, and $\tau_{3j}^{-1} > 1$, $\tau_{4j}^{-1} > 1$, $\forall j \in \mathcal{E}_g$. Then Φ is positive definite.*

Here, d_i denotes the degree of node/player i. Lemma 1 is the direct result of the Gershgorin circle theorem: a sufficient condition for the design matrix to be positive definite is that all of its Gershgorin discs lie on the positive orthant [21]. The proof is trivial, and therefore omitted here.

Let $\mathcal{K}$ be the inner product space obtained by endowing the vector space $\mathbb{R}^{n(N+E_g)+m(N+E_g)}$ with the inner product $\langle \psi, \psi' \rangle_{\mathcal{K}} = \langle \Phi\psi, \psi' \rangle$.

Lemma 2. *Suppose the design matrix Φ is positive definite. Then the operator $\tilde{\mathcal{B}}$ is maximally monotone on $\mathcal{K}$, and its reflected resolvent $R_{\tilde{\mathcal{B}}}$ is nonexpansive on $\mathcal{K}$.*

Lemma 3. *Suppose the design matrix Φ is positive definite and Assumptions 2 and 4 hold. Then the operator $\tilde{\mathcal{A}}$ is maximally monotone on $\mathcal{K}$, and its reflected resolvent $R_{\tilde{\mathcal{A}}}$ is nonexpansive on $\mathcal{K}$.*

Theorem 2. *Suppose that Assumptions 1-4 hold and the design matrix Φ satisfies the inequalities in Lemma 1. Then the sequence $(\boldsymbol{y}^{(k)})_{k\in\mathbb{N}}$ and $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ generated by Algorithm 1 satisfy $\lim_{k\to\infty} \boldsymbol{y}^{(k)} = (\mathbf{1}_N \otimes y^*)$ and $\lim_{k\to\infty} \boldsymbol{\lambda}^{(k)} = (\mathbf{1}_N \otimes \lambda^*)$, where y^* is a v-GNE to problem (1) and (y^*, λ^*) together is a solution to the KKT conditions (5).*

Theorem 2 directly follows from the fact that $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are two maximally monotone operators on $\mathcal{K}$ and the standard D-R results [18, Thm. 26.11]. Next, we are going to analyze the convergence properties under the Assumption 5 instead using the key notion of restricted monotonicity [11]. Given an operator T, we say that T is restricted monotone w.r.t. a set S if, for all (x,u) and $(x^*,u^*) \in \mathrm{gra}(T)$ with $x^* \in S$, $\langle x - x^*, u - u^*\rangle \geq 0$. Restricted nonexpansiveness and restricted firm nonexpansiveness are defined similarly. In particular, a single-valued operator T is quasinonexpansive if for all $x \in \mathrm{dom}(T)$ and $x^* \in \mathrm{Fix}(T)$, we have $\|Tx - x^*\| \leq \|x - x^*\|$ [18, Def. 4.1]. The preliminary results are included in Lemma 4 and the main results will be given in Theorem 3 below.

Lemma 4. *Suppose Assumptions 1-3 and 5 hold and $\rho_\mu \geq \frac{2}{\sigma_1}(\frac{(\theta_1+\theta_2)^2}{4\eta} + \theta_2)$. Then the operator $\tilde{\mathcal{A}}$ is restricted monotone w.r.t. $Zer(\mathcal{T})$ on $\mathcal{K}$, i.e., for any ψ and ψ^* with $\psi^* \in Zer(\mathcal{T})$, $\langle \psi - \psi^*, \tilde{\mathcal{A}}(\psi) - \tilde{\mathcal{A}}(\psi^*)\rangle_{\mathcal{K}} \geq 0$. Moreover, its reflected resolvent $R_{\tilde{\mathcal{A}}}$ is restricted nonexpansive w.r.t. $Zer(\mathcal{T})$ on $\mathcal{K}$.*

Remark 3. *We observe that under Assumption 1, even though $\tilde{\mathcal{A}}$ is merely restricted monotone w.r.t. $Zer(T)$ on $\mathcal{K}$, $J_{\tilde{\mathcal{A}}}$ is still well-defined and single-valued on $\mathcal{K}$. The computation of $\psi^{(k+1)} \in J_{\tilde{\mathcal{A}}}(\tilde{\psi}^{(k)})$ is explicitly described in the first player and edge for-loops in Algorithm 1. Obviously, the iterations of $\{y_i^{-i(k+1)}\}$, $\boldsymbol{\lambda}^{(k+1)}$, $\boldsymbol{\mu}^{(k+1)}$, and $\boldsymbol{z}^{(k+1)}$ are linear and hence single-valued. The updates of $\{y_i^i\}$ are described by a set of unconstrained optimization problems.*

Each objective is the sum of a convex function and a proximal term, and hence admits a unique argmin *solution.*

Theorem 3. *Suppose that Assumptions 1-3 and 5 hold, the design matrix Φ satisfies the inequalities in Lemma 1, and $\rho_\mu \geq \frac{2}{\sigma_1}(\frac{(\theta_1+\theta_2)^2}{4\eta} + \theta_2)$ as suggested in Lemma 4. Then the sequences $(\boldsymbol{y}^{(k)})_{k\in\mathbb{N}}$ and $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ generated by Algorithm 1 satisfy $\lim_{k\to\infty} \boldsymbol{y}^{(k)} = (\mathbf{1}_N \otimes y^*)$ and $\lim_{k\to\infty} \boldsymbol{\lambda}^{(k)} = (\mathbf{1}_N \otimes \lambda^*)$, where y^* is a v-GNE to problem* (1) *and (y^*, λ^*) together is a solution to the KKT conditions* (5)*.*

Altogether, Theorems 2 and 3 establish the convergence of the proposed algorithm to a v-GNE regardless of the initialization. Nevertheless, the convergence speed will depend on the initial point, the choices of $\boldsymbol{\tau}_1$ to $\boldsymbol{\tau}_4$, ρ_μ and ρ_z, and the topology of $\mathcal{G}$. Characterizing the convergence speed is one of our future directions.

V. Case Study and Numerical Simulations

In this study, we evaluate the performance of the proposed algorithm with a Nash-Cournot game over a network. In the Nash-Cournot game, N manufacturers/players indexed by $\mathcal{N}$ are involved in producing a homogeneous commodity and competing for m different markets. The maximal capacities of these m markets are denoted by the vector $c \in \mathbb{R}^m_{++}$.

For each manufacturer i in this network, it supplies n_i markets with $x_i \in \mathbb{R}^{n_i}$ units of commodities respectively. It is subject to the global market capacity constraints $A_i x_i \leq c - \sum_{j\in\mathcal{N}_{-i}} A_j x_j$. The binary full-column-rank matrix $A_i \in \mathbb{R}^{m\times n_i}$ maps each entry of the decision vector x_i to one among the m markets. Specifically $[A_i]_{j,k} = 1$ implies that the firm i supplies market j with $[x_i]_k$ units of commodities. Let $n := \sum_{i\in\mathcal{N}} n_i$, $x := [x_1; \cdots; x_N] \in \mathbb{R}^n$, and $A := [A_1, A_2, \ldots, A_N] \in \mathbb{R}^{m\times n}$. Then, $Ax \in \mathbb{R}^m$ denotes the total quantities of commodities delivered to these m markets.

The local objective function is assumed to be of the form $J_i(x_i; x_{-i}) = f_i(x_i) - (P(Ax))^T A_i x_i$. We set $f_i(x_i) = x_i^T Q_i x_i + q_i^T x_i$ as the local production cost function, with $Q_i \in \mathbb{S}^{n_i}_{++}$. Let $P(Ax) = w - \Sigma Ax$ map the total quantities of supply to their unit prices, where $w \in \mathbb{R}^m_{++}$ and $\Sigma \in \mathrm{diag}(\mathbb{R}^m_{++})$.

To sum up, manufacturer $i \in \mathcal{N}$, given the supply strategies of others (x_{-i}), aims to solve the optimization problem:

$$\begin{aligned} &\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\ x_i^T Q_i x_i + q_i^T x_i - (w - \Sigma\cdot Ax)^T A_i x_i \\ &\text{subject to } A_i x_i \leq c - \textstyle\sum_{j\in\mathcal{N}_{-i}} A_j x_j. \end{aligned} \tag{18}$$

A. Analysis of Nash-Cournot Game

For any $i \in \mathcal{N}$, the objective function $J_i(x_i; x_{-i})$ is a smooth function, and it satisfies the Assumption 1. To find the pseudogradient F, notice that $\nabla_{x_i} J_i(x_i; x_{-i}) = (2Q_i + A_i^T \Sigma A_i)x_i + A_i^T \Sigma Ax + q_i - A_i^T w$. By concatenating these partial derivatives together, we obtain $F : x \mapsto (M_F + A^T\Sigma A)x + \boldsymbol{q} - A^T w$, where $M_F := \mathrm{blkd}([2Q_i + A_i^T \Sigma A_i]_{i\in\mathcal{N}})$ and $\boldsymbol{q} := [q_1; \cdots; q_N]$.

Since $Q_i \succ \mathbf{0}, \forall i \in \mathcal{N}$, and $\Sigma \succ \mathbf{0}$, we can conclude that $M_F \succ 0$ and $A^T\Sigma A \succcurlyeq \mathbf{0}$. Altogether, $M_F + A^T\Sigma A \succ \mathbf{0}$. Its minimal eigenvalue $\sigma_{F\min} > 0$ and maximal eigenvalue $\sigma_{F\max} > 0$ are the strongly monotone constant (η) and the Lipschitz constant (θ_1) of the pseudogradient F, respectively.

The extended pseudogradient $\mathcal{F}$ can be expressed as: $\mathcal{F} : \boldsymbol{x} \mapsto M_F \cdot \mathcal{R}\boldsymbol{x} + \mathrm{blkd}([A_i^T \Sigma A]_{i\in\mathcal{N}})\boldsymbol{x} + \boldsymbol{q} - A^T w$, where $\mathcal{R}$ is defined in (10). It can be further simplified to the form: $\mathcal{F} : \boldsymbol{x} \mapsto \mathcal{R}(I_N \otimes (M_F + A^T\Sigma A))\boldsymbol{x} + \boldsymbol{q} - A^T w$. The Lipschitz constant of $\mathcal{F}$ is given by the greatest singular value of $\mathcal{R}(I_N \otimes (M_F + A^T\Sigma A))$, and denoted by $\sigma_{\mathcal{F}\max}$.

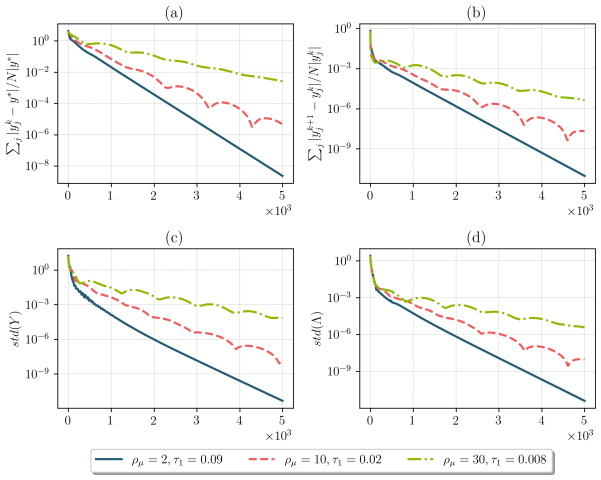

Fig. 1: Convergence Result of Algorithm 1 on Nash Cournot Game (Under Assumption 4)

B. Simulation Results

In the numerical study, let $N = 20$ and $m = 10$. The communication graph consists of a directed circle and 10 randomly selected edges which satisfies Assumption 3. The related parameters are drawn uniformly randomly from suitable intervals. Each entry of vector c satisfies $c_i \sim U[0.5, 1]$; for $P(Ax)$, each entry of vector w satisfies $w_i \sim U[2, 4]$ and each diagonal entry of Σ satisfies $\Sigma_{ii} \sim U[0.5, 0.7]$; for $f_i(x_i)$, assuming Q_i is diagonal, the diagonal entry $[Q_i]_{jj} \sim U[1, 1.5]$, and each entry of q_i has $[q_i]_j \sim U[0.1, 0.6]$; the feasible set $\mathcal{X}_i$ is the direct product of n_i connected compact interval $[0, \mathcal{X}_{ij\max}]$, and $\mathcal{X}_{ij\max} \sim [0.2, 0.5]$; manufacturer i supplies to n_i markets with $n_i \sim \{2, \ldots, 6\}$.

For the case of Assumption 4, since $\mathcal{F}$ is affine in this case, it suffices to select a $\rho_\mu \geq 2$ such that $\frac{1}{2}(M_\mathcal{F} + M_\mathcal{F}^T) + \frac{\rho_\mu}{2}(L \otimes I_n) \succcurlyeq 0$, where $M_\mathcal{F} = \mathcal{R}^T \cdot \mathcal{R}(I_N \otimes (M_F + A^T\Sigma A))$; hence $\mathcal{R}^T\mathcal{F} + \frac{\rho_\mu}{2}L_n$ is maximally monotone. Let $\boldsymbol{\tau}_1 = \tau_1 I_{Nn}$, $\boldsymbol{\tau}_2 = \tau_2 I_{Nm}$, $\boldsymbol{\tau}_3 = \tau_3 I_{En}$, and $\boldsymbol{\tau}_4 = \tau_4 I_{Em}$, which are chosen according to Lemma 1.

For the case of Assumption 5, for F, $\eta = \sigma_{F\min} \approx 2.6513$, and $\theta_1 = \sigma_{F\max} \approx 10.6646$; for $\mathcal{F}$, $\theta_2 = \sigma_{\mathcal{F}\max} \approx 4.7084$. The nodes on $\mathcal{G}$ have maximal degree equal to 4. Moreover, for the Laplacian matrix L, $\sigma_1 \approx 0.4701$. Then, based on Lemmas 1 and 4, select $\rho_\mu = 115 \geq \frac{2}{\sigma_1}(\frac{(\theta_1+\theta_2)^2}{4\eta} + \theta_2) \approx 114.8432$, $\rho_z = 1$, $\tau_1 = 0.00195$, $\tau_2 = 0.14$, and $\tau_3 = \tau_4 = 0.9$. We compare the performance of the proposed algorithm with the proximal-point algorithm in [12] and the FB splitting method in [11] under the same communication graph and properly chosen parameters such as step sizes.

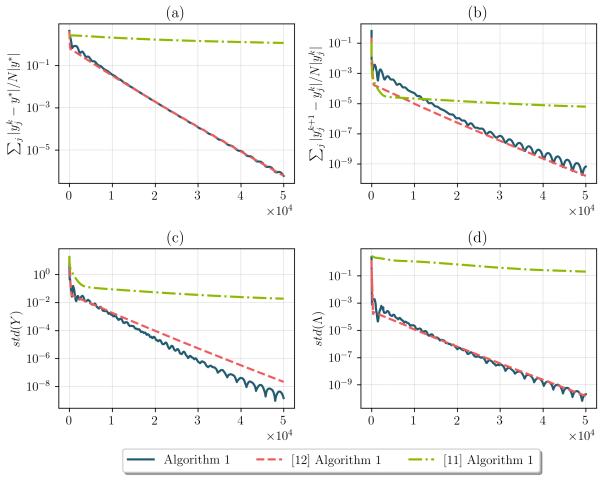

Fig. 2: Convergence Result of Algorithm 1 on Nash Cournot Game (Under Assumption 5)

The performances of our proposed algorithm are illustrated in Fig. 1 and 2, under the Assumptions 4 and 5, respectively. Fig. 1/2a show the average of the normalized distances to the v-GNE calculated using the centralized method from [22]. Note that y_j^k denotes player j's local estimate of the decision vector at the kth iteration, and y^* the generalized Nash equilibrium of the game. Fig. 1/2b show the relative length of the updating step at each iteration. Let $\bar{y}^k := \frac{1}{N}\sum_{j\in\mathcal{N}} y_j^k$. Fig. 1/2c exhibit how the sum of the standard deviations of the local estimates $\{y_j\}_{j\in\mathcal{N}}$, i.e., $\sum_{\ell=1}^{n}(\frac{1}{N}\sum_{j\in\mathcal{N}}([y_j^k]_\ell - [\bar{y}^k]_\ell)^2)^{\frac{1}{2}}$, evolves over the iterations. It measures the level of consensus among different local estimates y_j. Fig. 1/2d are almost the same as Fig. 1/2c except that we are now investigating the consensus of local dual variables $\{\lambda_j\}_{j\in\mathcal{N}}$. The numerical results verify Theorems 2 and 3 and show a linear convergence rate for each metric considered. As illustrated in Fig. 2, our proposed algorithm achieves a similar convergence rate as the proximal point algorithm in [12] and shows a much faster convergence compared with the FB splitting algorithm in [11], whose step sizes are severely restricted to satisfy the cocoercive conditions of the preconditioned forward operator.

VI. Conclusion and Future Directions

This paper focuses on the GNEP with generic interdependence inside the local objectives and affine coupling constraints. A distributed algorithm is proposed, which ensures exact convergence to a v-GNE and only requires local communications. For future directions, it would be interesting to develop a set of equivalent transformations that can simplify the problem solution while preserving the v-GNE. Moreover, in this paper, we let each player keep a local copy of all players' decisions. It would be beneficial to explore the possibility of reducing the number of local estimates for those games with structured interdependency, e.g., average/network aggregate games.

References

[1] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized Nash games and variational inequalities," *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, 2007.

[2] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[3] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for network resource allocation problems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 621–636, 2019.

[4] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on control systems technology*, vol. 21, no. 1, pp. 67–78, 2011.

[5] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, "Distributed seeking of Nash equilibria with applications to mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 904–919, 2011.

[6] X. Wei and A. Anastasopoulos, "Mechanism design for demand management in energy communities," *Games*, vol. 12, no. 3, p. 61, 2021.

[7] P. Yi and L. Pavel, "An operator splitting approach for distributed generalized Nash equilibria computation," *Automatica*, vol. 102, pp. 111–121, 2019.

[8] ——, "Distributed generalized Nash equilibria computation of monotone games via double-layer preconditioned proximal-point algorithms," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 299–311, 2018.

[9] G. Belgioioso and S. Grammatico, "A distributed proximal-point algorithm for Nash equilibrium seeking in generalized potential games with linearly coupled cost functions," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1–6.

[10] L. Pavel, "A doubly-augmented operator splitting approach for distributed GNE seeking over networks," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3529–3534.

[11] ——, "Distributed GNE 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.

[12] M. Bianchi, G. Belgioioso, and S. Grammatico, "Fast generalized Nash equilibrium seeking under partial-decision information," *arXiv preprint arXiv:2003.09335*, 2020.

[13] Y. Huang and J. Hu, "A distributed GNE seeking algorithm using the Douglas-Rachford splitting method," *arXiv preprint arXiv:2103.09393*, 2021.

[14] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[15] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

[16] D. Chan and J. Pang, "The generalized quasi-variational inequality problem," *Mathematics of Operations Research*, vol. 7, no. 2, pp. 211–222, 1982.

[17] 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.

[18] H. H. Bauschke, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed., ser. CMS Books in Mathematics, Ouvrages de mathématiques de la SMC, 2017.

[19] M. Bianchi and S. Grammatico, "A continuous-time distributed generalized Nash equilibrium seeking algorithm over networks for double-integrator agents," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 1474–1479.

[20] R. Johansson and A. Rantzer, *Distributed decision making and control*. Springer, 2012.

[21] H. E. Bell, "Gershgorin's theorem and the zeros of polynomials," *The American Mathematical Monthly*, vol. 72, no. 3, pp. 292–295, 1965.

[22] G. Belgioioso and S. Grammatico, "Projected-gradient algorithms for generalized equilibrium seeking in aggregative games arepreconditioned forward-backward methods," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2188–2193.