Abstract—We consider the framework of aggregative games with affine coupling constraints, where agents have partial information of the aggregate value and can only obtain information from neighboring agents. We propose a single-layer distributed algorithm that reaches a variational generalized Nash equilibrium, under constant step sizes. The algorithm works on a single timescale, i.e., does not require multiple communication rounds between agents before updating their action. The convergence proof leverages an invariance property of the aggregate estimates and relies on a forward-backward splitting for two preconditioned operators and their restricted (strong) monotonicity properties on the consensus subspace.

I. INTRODUCTION

Research in aggregative games has surged in recent years due to their suitability to model decision problems in various application domains: from demand-side management for the smart grid, [1] and for electric vehicles, [2], [3], to wireless systems, [4], network congestion control, [5] and competitive markets, [6]. Aggregative games are non-cooperative games in which each (player’s) agent’s cost depends on some aggregate effect of all other agents’ actions. Often agents have shared coupling constraints, [7], and the relevant equilibrium concept is the generalized Nash equilibrium (GNE).

Many settings involve a large number of agents that have private cost functions and constraints, who are willing and able to exchange information with their neighbours only, [8], [9]. Motivated by the above, in this note we develop a single-layer/single-timescale, distributed GNE seeking algorithm for aggregative games, which is guaranteed to converge exactly to a variational GNE when using fixed-step sizes. Agents perform simultaneous action update and aggregate-estimate update, based on local communication. To the best of our knowledge, this is the first such algorithm in the literature. Our novel contributions consist in relating the algorithm to a preconditioned forward-backward splitting iteration for a specific pair of monotone operators, and developing conditions for distributed convergence.

Literature Review: The problem of finding a (generalized) Nash Equilibrium (G)NE when agents know the actions of all other agents (full-decision information) has been studied thoroughly, e.g., [10], [11], [12], [13]. In recent years, a rapidly growing field is concerned with the development of algorithms that relax this full-information assumption, i.e., (G)NE computation over networks, [14], [15]. Aggregative games can be classified as semi-decentralized (when a central coordinator is required), or distributed (when only local communication is used, under partial information).

Semi-decentralized GNE algorithms have been proposed for aggregative games in [16], [17], [18], [19] (in discrete-time) and [15] (in continuous-time). An elegant operator theory approach is used to show that global convergence can be achieved with fixed-step sizes under (strict) strong monotonicity of the pseudo-gradient of the game, either to a variational generalized Nash equilibrium (GNE), or to an aggregative (Wardrop) equilibrium (GAE), the latter being an ϵ-GNE, with ϵ tending to zero as the number of agent goes to infinity, [18]. The algorithms require a central coordinator to broadcast the aggregate value and to ensure the coupling constraints are met.

In a distributed setting a central coordinator/node does not exist, and distributed algorithms are more difficult to develop. Such distributed algorithms have been mostly developed for NE seeking, i.e., for games with no coupled constraints. The first work to propose distributed NE algorithms for aggregative games over networks was [14], [20]. In order to cope with not knowing the aggregate value, each agent maintains an estimate of the true aggregate, built based on local communication with neighbours, and uses it in the action update instead of the true aggregate. The algorithm proposed in [20] requires diminishing step-sizes for exact convergence, while only guaranteeing convergence to a neighbourhood of the NE for fixed step-sizes, under (strict) strong monotonicity of the pseudo-gradient of the game. The recent algorithm proposed in [21] requires an increasing number of communication rounds before each action update. In both cases, the NE seeking algorithms effectively operate as if on two timescales (fast aggregate estimate, slow action update) and each agent is using an aggregate estimate that is near the true aggregate.

Even fewer results exist for distributed GNE seeking in aggregative games with coupled constraints. In fact, the only such distributed discrete-time algorithm that we are aware of is the one proposed in [22]. The algorithm requires that players exchange information for a fixed number of communication rounds before every action update (thus is also essentially a two time-scale/two-layer algorithm) and is only guaranteed to reach an ϵ-GNE. The ϵ-GNE approaches a variational GNE only if the fixed number of communication rounds between action update goes to infinity, which is impractical. We note that a continuous-time dynamics for aggregative games with equality coupling constraints only is proposed in [23], with gains proportional to the number of agents. However, in general convergence in continuous-time does not guarantee convergence in discrete-time.

Contributions: Motivated by the above, in this paper we propose a distributed, single-layer discrete-time GNE seeking algorithm for aggregative games with affine coupling constraints, that has guaranteed exact convergence a variational GNE when using constant step-sizes. To the best of our knowledge, to date there does not exist such an algorithm. Each agent maintains an estimate of the aggregate and multiplier, and exchanges them with its neighbors over an undirected static connected graph, in order to learn the true aggregate value and ensure that the coupling constraints are satisfied. Each agent updates its action and its aggregate estimates in the same iteration. Compared to the distributed NE algorithm in [20], and the GNE algorithm in [22], our algorithm does not require diminishing step-sizes nor multiple communication rounds between each action update. The algorithm is related to a preconditioned forward-backward operator splitting iteration, inspired by the distributed framework for GNE seeking in general games conceptualized in
[24], [25] for full-decision information, and extended in [26] to partial-decision information. However, we note that the algorithms in [24], [25], [26] are developed for generally coupled games, and are not scalable nor efficient if applied to aggregative games. In [24], [25] while agents estimate dual multipliers distributively, each agent is assumed to know the actions of all players that affect its cost. Hence, when applied to an aggregative game, the algorithm is not distributed, since it requires communication with all other agents. The algorithm in [26] relaxes this assumption to partial-decision information and, to cope with the lack of global information, each agent is required to keep an estimate of all other agents’ actions (as needed due to the generality of the cost). This is not scalable for an aggregative game and moreover, not actually needed since only the estimate of the aggregate is necessary. Thus, while the algorithm in [26] is distributed (unlike [24]), it is computationally and memory-wise inefficient if applied to an aggregative game.

The distributed algorithm we develop here bypasses these disadvantages; it is tailored for aggregative games and exploits the aggregative coupling structure in the cost. Each agent exchanges and maintains only an estimate of the aggregate (dimensionally independent on the number of agents, hence scalable) and an estimate of the dual multiplier. Thus players do not need to share action information, which might be private information. While we use proof techniques similar to those in [26], there are substantial differences. The algorithm is different (the aggregate estimate update has an additional correction term) which technically requires that we use different splitting operators. To show the required (restricted) monotonicity/coercivity properties for these operators we cannot apply results in [26], but rather we do it taking into account their specific structure. Herein we take advantage of the aggregative coupling structure and, unlike [26], we treat separately the actions and the aggregate estimates. Furthermore, we specifically exploit invariance properties on the aggregate consensus subspace. A short version of this work is submitted [27]; compared to the short version, this paper contains detailed proofs omitted from the conference version and additional numerical results.

The paper is organized as follows. Section II gives preliminary background. Section III formulates the aggregative game. Section IV introduces the distributed GNE seeking algorithm for the aggregative game and relates it to a forward-backward operator-splitting iteration. The convergence analysis is presented in Section V Numerical results are provided in Section VI as well as comparison to the algorithm in [22]. Concluding remarks are given in Section VII.

Notations. For a vector $x \in \mathbb{R}^m$, $x^T$ denotes its transpose and $\|x\| = \sqrt{x^T x}$ the norm induced by inner product $\langle \cdot, \cdot \rangle$. For a symmetric positive-definite matrix $\Phi > 0$, $\lambda_{\min}(\Phi)$ and $\lambda_{\max}(\Phi)$ denote its minimum and maximum eigenvalues. The $\Phi$-induced inner product is $\langle x, y \rangle_\Phi = \langle \Phi x, y \rangle$ and the induced norm, $\|x\|_\Phi = \sqrt{(\Phi x)^T x}$. For a matrix $A \in \mathbb{R}^{m\times n}$, let $\|A\| = \sigma_{\max}(A)$ denote the 2-induced matrix norm, where $\sigma_{\max}(A)$ is its maximum singular value. For $N = \{1, \ldots, N\}$, $col(x_{i})_{i \in N}$ denotes the stacked vector obtained from vectors $x_i$. Similarly, $diag(x_{i})_{i \in N}$ is the diagonal matrix, with element $x_i$ along the diagonal. $Null(A)$ and $Range(A)$ are the null and range space of matrix $A$, respectively, while $|A|_{ij}$ stands for its $(i,j)$ entry. $I$, $1$ and $0$ denotes the identity matrix, ones vector and zeros vector of appropriate dimensions. Denote $\Pi_{i=1}^{N} \Omega_{i}$ as the Cartesian product of sets $\Omega_{i}$, $i = 1, \ldots, N$. For a function $f(x) = f(col(x_{i})_{i \in N})$ let $\nabla_x f(x) = \frac{\partial}{\partial x_i} f(x)$.

II. Background

Monotone Operators

The following are from [28]. Let $A : \mathbb{R}^m \to 2^{\mathbb{R}^m}$ be a set-valued operator. The domain of $A$ is $dom(A) = \{ x \in \mathbb{R}^m \mid A x \neq \emptyset \}$ where $\emptyset$ is the empty set, and the range of $A$ is $ran(A) = \{ y \in \mathbb{R}^m \mid \exists x, y \in \mathbb{R}^m \mid A x = y \}$. The graph of $A$ is $gra(A) = \{ (x,u) \in \mathbb{R}^m \times \mathbb{R}^m \mid u \in A x \}$. The zero set of $A$ is $zer(A) = \{ x \in \mathbb{R}^m \mid 0 \in A x \}$. The operator $A$ is called monotone if $\forall (x,u), (y,v) \in gra(A)$, $\langle x - y, u - v \rangle \geq 0$. It is called strongly monotone if $\exists \mu > 0$ such that $\forall (x,u), (y,v) \in gra(A)$, $\langle x - y, u - v \rangle \geq \mu \| x - y \|^2$. It is maximally monotone if $gra(A)$ is not strictly contained in the graph of any other monotone operator. The resolvent of $A$ is $J_{\lambda A} = (Id + A)^{-1}$, where $Id$ is the identity operator. The fixed points of the operator $A$ are $Fix(A) = \{ x \in \mathbb{R}^m \mid x \in A x \}$.

An operator $T : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$ is nonexpansive if $\|T(x) - T(y)\| \leq \|x - y\|, \forall x, y \in \Omega$. An operator $T \in \mathcal{A}(\alpha)$, where $\mathcal{A}(\alpha)$ denotes the class of $\alpha$-averaged operators, if and only if $\forall x, y \in \Omega$, $\|T x - T y\|^2 \leq \|x - y\|^2 - \frac{\alpha}{2} \|x - y\|^2 - \frac{\alpha}{2} \|T x - T y\|^2$. If $T \in \mathcal{A}(\frac{1}{2})$ it is also called firmly nonexpansive. If $A$ is maximally monotone then $J_{\lambda A} = (Id + A)^{-1}$ is firmly nonexpansive. [28], Prop. 23.7. Let the projection of $x$ onto $\Omega$ be $P_{\Omega}(x) = \arg \min_{y \in \Omega} \|x - y\|$, with $P_{\Omega}(x) = J_{\lambda \Omega}(x) = (Id + \lambda \Omega)^{-1}$, where $\lambda \Omega(x) = \{ v \mid \langle v, y \rangle - \langle x, y \rangle \leq 0, \forall y \in \Omega \}$ is the normal cone operator of $\Omega$ at $x$. If $\Omega$ is closed and convex, $P_{\Omega}$ is firmly nonexpansive since $\Omega$ is maximally monotone [28], Prop. 4.8. The operator $T$ is called $\beta$-cocoercive if $\beta T \in \mathcal{A}(\frac{1}{2})$ for $\beta > 0$, i.e., $\beta \|T x - T y\|^2 \leq \langle x - y, T x - T y \rangle, \forall x, y \in \Omega$.

Graph Theory

Let graph $G = (N, E)$ describe the information exchange among a set $N$ of agents, where $E \subset N \times N$ is the edge set. If agent $i$ can get information from agent $j$, then $(j, i) \in E$ and agent $j$ belongs to agent $i$’s neighbour set $N_i = \{ j \mid (j, i) \in E \}$. $G$ is undirected when $(i, j) \in E$ if and only if $(j, i) \in E$. $G$ is connected if there is a path between any two nodes. Let $W = [w_{ij}] \in \mathbb{R}^{N \times N}$ be the weighted adjacency matrix, with $w_{ij} > 0$ if $j \in N_i$, and $w_{ij} = 0$ otherwise. Let $D^{eg} = diag(d_{i})_{i \in N}$, where $d_{i} = \sum_{j \in N} w_{ij}$. Assume that $W = W^{T}$ so the weighted Laplacian of $G$ is $L = D^{eg} - W$. When $G$ is connected and undirected, $0$ is a simple eigenvalue of $L$, $L 1_N = 0$, $1_N^T L = 0^T$. All other eigenvalues of $L$ are positive, ordered in ascending order as $0 < \lambda_2(L) \leq \cdots \leq \lambda_N(L)$, with $d^* = \max_{i} d_{i}$, $d^*$ is the maximal weighted degree of $G$.

III. Game Formulation

Consider a group of agents (players) $N = \{1, \ldots, N\}$, each player $i \in N$ controls its local decision (action/strategy) $x_{i} \in \mathbb{R}^{n}$. Denote $x = col(x_{i})_{i \in N} \in \mathbb{R}^{Nn}$ as the decision profile. Equivalently we can also write $x = (x_{i}, x_{-i})$ where $x_{-i} = col(\ldots, x_{i-1}, x_{i+1}, \ldots)$ denotes the decision profile of all other agents, except player $i$. Agent $i$ aims to optimize its objective function $J_{i}(x_{i}, x_{-i})$ (coupled to other players’ decisions) with respect to its own decision $x_{i}$ over its feasible decision set $\Omega_{i}$.

Let the globally shared, affine coupled constrained set be

$$K = \prod_{i=1}^{N} \Omega_{i} \cap \{ x \in \mathbb{R}^n \mid \sum_{i=1}^{N} A_{i} x_{i} \leq \sum_{i=1}^{N} b_{i} \}$$

where $\Omega_{i} \subset \mathbb{R}^{n}$ is a private feasible set of player $i$, and $A_{i} \in \mathbb{R}^{m \times n}$, $b_{i} \in \mathbb{R}^{m}$ is local player information. Let the overall feasible action space be $\Omega = \prod_{i=1}^{N} \Omega_{i}$.
We focus on average aggregative games where the cost function $J_i(x_i, x_{-i})$ of each agent $i$ depends on the average of all agents’ actions $\sigma(x) = \frac{1}{N} \sum_{j=1}^{N} x_j \in \mathbb{R}^n$, denoted as $\tilde{J}_i(x_i, \sigma(x))$ to explicitly indicate this dependency. In the remainder of the paper we use either $J_i$ or $\tilde{J}_i$, depending on the context. Given the other’s actions $x_{-i}$, the objective of each player $i$ is to solve the following optimization problem with coupled constraints,

$$\min_{x_i} \tilde{J}_i(x_i, x_{-i}) \text{ s.t. } (x_i, x_{-i}) \in K$$  \hspace{1cm} (1)

A generalized Nash equilibrium (GNE) of the game is a profile $x^* = \{x_i^*\}_{i \in \mathcal{N}}$ such that

$$\forall i \in \mathcal{N} \quad x_i^* = \arg \min_{x_i} \tilde{J}_i(x_i, x_{-i}^*) \text{ s.t. } (x_i^*, x_{-i}^*) \in K$$

### Assumption 1
For each player $i$, given any $x_{-i}$, $\tilde{J}_i(x_i, x_{-i})$ is continuously differentiable and convex in $x_i$ and $\Omega_i$ is a compact convex set. The constraint set $K$ is non-empty and satisfies Slater’s constraint qualification.

### Assumption 2
A standard assumption which ensures existence of a generalized Nash equilibrium (GNE).

Given the optimization problem (1) over $K$ for each agent $i$, let its Lagrangian be defined as $L_i(x_i, x_{-i}; x_{-i}) = \tilde{J}_i(x_i, x_{-i}) + \lambda_i^T (Ax_i - b)$, where $A = [A_1, \ldots, A_N], b = \sum_{i=1}^{N} b_i$ and the dual multiplier is $\lambda_i \in \mathbb{R}^n_+$. Then, the KKT conditions that an optimal solution $x_i^*$ with $\lambda^*_i$ satisfies can be written as

$$0 \in \nabla_{x_i} \tilde{J}_i(x_i^*, x_{-i}^*) + A_i^T \lambda_i^* + N_{\Omega_i}(x_i^*)$$

$$0 \in -(Ax_i^* - b) + N_{\mathbb{R}^n_+}(\lambda_i^*)$$  \hspace{1cm} (2)

where for an aggregative game, $\nabla_{x_i} \tilde{J}_i(x_i, x_{-i}) = \nabla_{x_i} J_i(x_i, x_{-i}) - \frac{1}{\mathcal{N}} \sum_{j \in \mathcal{N}} \nabla_{x_i} J_i(x_i, x_{-i})$ and $\nabla_{x_i} J_i(x_i, x_{-i}) \in \mathbb{R}^n$.

A variational GNE of the game is defined as $x^* \in K$ solution of the variational inequality $VI(F, K)$:

$$\langle F(x^*) | x - x^* \rangle \geq 0, \quad \forall x \in K$$

where $F$ denotes the pseudo-gradient of the game defined as $F(x) = \text{col}(\nabla_{x_i} J_i(x_i, x_{-i}))_{i \in \mathcal{N}} = \text{col}(\nabla_{x_i} J_i(x_i, \sigma(x)))_{i \in \mathcal{N}}$, i.e., the stacked vector with the partial gradients for all $i \in \mathcal{N}$. $x^*$ solves $VI(F, K)$ if and only if there exists a $\lambda^* \in \mathbb{R}^n_+$ such that the KKT conditions are satisfied, [13, Thm. 10.1],

$$0 \in F(x^*) + A^T \lambda^* + N_{\Omega}(x^*)$$

or, component-wise, for any $i \in \mathcal{N}$,

$$0 \in \nabla_{x_i} \tilde{J}_i(x_i^*, x_{-i}^*) + A_i^T \lambda_i^* + N_{\Omega_i}(x_i^*)$$

$$0 \in -(Ax_i^* - b) + N_{\mathbb{R}^n_+}(\lambda_i^*)$$  \hspace{1cm} (3)

where $N_{\Omega}(x^*) = \prod_{i=1}^{N} N_{\Omega_i}(x_i^*)$. Assumption 1 guarantees the existence of a solution to $VI(F, K)$, by [13, Corollary 2.2.5]. By [12, Thm. 9, §4], every solution $x^*$ of $VI(F, K)$ is a GNE of the game. Furthermore, if $x^*$ together with $\lambda^*$ satisfies the KKT conditions (2) for $VI(F, K)$ then $x^*$ satisfies the KKT conditions (2) with $\lambda_1^* = \ldots = \lambda_N^* = \lambda^*$, hence $x^*$ is a variational GNE of the game.

Given this aggregative game, our aim is to design a distributed iterative algorithm that finds a variational GNE under partial-decision information over a network.

### Assumption 2
$F$ is strongly monotone and Lipschitz continuous, i.e., there exist $\mu > 0$ and $L_F > 0$ such that for any $x$ and $x'$, $\langle x - x' | F(x) - F(x') \rangle \geq \mu \|x - x'\|^2$, and $\|F(x) - F(x')\| \leq L_F \|x - x'\|$.

Assumption 2 is commonly used in algorithms with fixed-step sizes, [20], [30], [31], [21], [16], [17], [18], [24], [22], [26], and guarantees that a unique variational GNE exists, [13]. We consider that each agent $i$ does not have information on the other agents actions $x_{-i}$ or on the aggregate value, $\sigma(x)$, and that agents communicate only locally with neighbouring agents, over a communication graph $G$.

### Assumption 3
The communication graph $G = (\mathcal{N}, \mathcal{E})$ is undirected and connected.

## IV. DISTRIBUTED ALGORITHM

In this section we present our proposed algorithm. To offset the lack of full information, each agent $i$ maintains a local estimate $u_i$ of the aggregate $\sigma(x)$ and a local multiplier $\lambda_i$, and exchanges them with its neighbours over $G$, in order to learn the true aggregate value and the Lagrange multiplier $\lambda^*$. Each agent also maintains an additional auxiliary variable $z_{ik}$, used for the coordination of the coupling constraints and to reach consensus of the local multipliers.

Let $(x_{ik}, u_{ik}, z_{ik}, \lambda_{ik})$ denote the tuple with agent $i$’s decision variable $x_{ik}$, local aggregate estimate $u_{ik}$, local multiplier $\lambda_{ik}$, and auxiliary variable $z_{ik}$ at iteration $k$, respectively. The goal is that over time each agent will have the same aggregate estimate, equal to the average of the agents actions, the same multiplier, and its decision will correspond to a variational GNE with the coupled constraints met. The proposed distributed algorithm is given below.

### Algorithm 1

$$x_{i,k+1} = P_{\Omega_i} \left[ x_{i,k} - \tau \left( \nabla_{x_i} J_i(x_{i,k}, u_{i,k}) + A_i^T \lambda_{i,k} \right) \right]$$

$$u_{i,k+1} = u_{i,k} - \kappa \sum_{j \in \mathcal{N}} (u_{i,k} - u_{j,k}) + (x_{i,k+1} - x_{i,k})$$

$$z_{i,k+1} = z_{i,k} + \nabla_{x_i} J_i(x_{i,k}, u_{i,k}) + A_i^T \lambda_{i,k}$$

$$\lambda_{i,k+1} = P_{\mathbb{R}^n_+} \left( \lambda_{i,k} - \alpha \sum_{j \in \mathcal{N}} (\lambda_{i,k} - \lambda_{j,k}) + b - A_i (2x_{i,k+1} - x_{i,k}) \right)$$

where $x_{i,0} \in \Omega_i$, $u_{i,0} = x_{i,0}$, $z_{i,0}, \lambda_{i,0} \in \mathbb{R}^n$, $\forall i \in \mathcal{N}$ and $\tau, \kappa, \alpha, \kappa \in \mathbb{R}^+$ are positive step sizes. Note that Algorithm 1 is fully distributed and instead of $(\sigma(x))$, each agent is using $u_i$ (own estimate of the aggregate) to evaluate $\nabla_{x_i} J_i(x_i, u_i)$, it’s own partial gradient.

To write the algorithm more compactly, let $x_k = \text{col}(x_{i,k})_{i \in \mathcal{N}}, u_k = \text{col}(u_{i,k})_{i \in \mathcal{N}}, z_k = \text{col}(z_{i,k})_{i \in \mathcal{N}}, \lambda_k = \text{col}(\lambda_{i,k})_{i \in \mathcal{N}}$. Let $F(x, u) = \text{col}(\nabla_{x_i} J_i(x_i, u_i))_{i \in \mathcal{N}}$ where $u_i \in \mathbb{R}^n$ and $u = \text{col}(u_{i,k})_{i \in \mathcal{N}}$ be the extended pseudo-gradient. Note that when all $u_i = \sigma(x)$, $F(x, u) \nabla \sigma(x) = \text{col}(\nabla_{x_i} J_i(x_i, \sigma(x)))_{i \in \mathcal{N}}$, hence $F(x, u) \nabla \sigma(x) = F(x)$. Thus, if in $F(x)$ each agent is evaluating the gradient with the true action aggregate value, in $F(x, u)$ each agent is using its own estimate of the aggregate, $u_i$ instead.

With these we can write Algorithm 1 compactly as,

$$x_{k+1} = P_{\Omega}(x_k - \tau (F(x_k, u_k) + A^T \lambda_k))$$

$$u_{k+1} = u_k - \kappa L_u u_k + (x_{k+1} - x_k)$$

$$z_{k+1} = z_k + \nabla_{x} L\lambda_k$$

$$\lambda_{k+1} = P_{\mathbb{R}^n_+} \left( \lambda_k - \alpha \sum_{j \in \mathcal{N}} (\lambda_j - \lambda_k) + b - \Lambda (2x_{k+1} - x_k) \right)$$

$$+ L \Lambda (2z_{k+1} - z_k)$$

3
where $x_0 \in \Omega \subseteq \mathbb{R}^N$, $u_0 = x_0$, $z_0 \in \mathbb{R}^m$, $\lambda_0 \in \mathbb{R}^m$, $P_0 = \text{col}(P(\lambda_0))_{\lambda_0 \in \mathbb{R}^m}$, $P_{0,x}^\top = \text{col}(P(\lambda_0))_{\lambda_0 \in \mathbb{R}^m}^\top$, $L_0 = L \otimes I_n$, $L_\lambda = L \otimes I_m$, and $L$ is the Laplacian matrix of the graph $G$, $\Lambda = \text{diag}(\Lambda(\mathbb{R}^N \times \mathbb{R}^m))$, $b = \text{col}(b(\lambda_0))_{\lambda_0 \in \mathbb{R}^m}$, $\tau = \text{diag}(\{\gamma\})_{\gamma \in \mathbb{R}^N}$, $v = \text{diag}(\{\gamma\})_{\gamma \in \mathbb{R}^N}$, and $\alpha = \text{diag}(\{\gamma\})_{\gamma \in \mathbb{R}^N}$.

Remark 1: The update for $x_k$ is a projected-gradient descent of the local Lagrangian function using player's aggregate estimate $u_k$ to evaluate the pseudo-gradient. The update for $u_k$ has a consensus term with an additional term to keep track of changes in the actions. The auxiliary variable $z_k$ is updated by a discrete-time integration for the multiplier consensus error. Lastly, the update for $\lambda_k$ is a combination of a projected-gradient ascent of the local Lagrangian and a proportional-integral term for the multiplier consensus error.

We first prove an important inventory property of Algorithm 1, namely that $\sigma(u_k) = \frac{1}{N} \sum x_{i,k}$ (the average of all agents’ aggregate estimates) is always equal to the actions true aggregate, $\sigma(x_k) = \frac{1}{N} \sum x_{i,k}$.

**Lemma 1**: Suppose Assumption 1 and 3 hold. Then the following properties hold for iterates $x_k, u_k, z_k, \lambda_k$ generated by Algorithm 1 or (4).

(i) $(1_N \otimes I_n)u_k = (1_N \otimes I_n)x_k$, hence $\sigma(u_k) = \sigma(x_k)$ for all $k \geq 0$.

(ii) Any limit point $(\bar{x}, \bar{u}, \bar{z}, \bar{\lambda})$ of Algorithm 1 is such that $\bar{x}_i = \bar{x}$ for all $i \in \mathbb{N}$, $\bar{u} = \bar{u}$, and $\bar{z}_i = \bar{z}$ is a variational GNE.

**Proof**: See Appendix.

Exploiting the invariance in Lemma [1], from (4), we next construct an auxiliary iteration with respect to the consensus subspace of the aggregate estimates, which will prove instrumental for the convergence analysis. Let $C$ denote the $n$-dimensional consensus subspace for all agents’ aggregate estimates, i.e., $C = \{u | u = 1_N \otimes c, c \in \mathbb{R}^n \}$ and $C^\perp$ be its orthogonal complement.

Note that $C = \text{Null}(L_0) = \text{Range}(1_N \otimes I_n)$ and $C^\perp = \text{Range}(L_0) = \text{Null}(1_N \otimes I_n)$.

Any $u \in \mathbb{R}^n$ can be decomposed as $u = P_1u + P_2u$, where $P_1u \in C$ and $P_2u \in C^\perp$, by using the projection matrices $P_1 = \frac{1}{N}1_N \otimes 1_N$ and $P_2 = I_n - P_1$.

Any $u_k$ generated by (4) can be decomposed as $u_k = P_1u_k + P_2u_k$, where, using the invariance property in Lemma [1],

$$P_1u_k = 1_N \otimes \sigma(u_k) = P_1x_k \quad \forall k \geq 0.$$  (5)

Using this decomposition together with (4), consider:

$$x_{k+1} = P_1[x_k - \tau(F(x_k, P_1x_k + u_k^\perp) + \lambda^T \lambda_k)]$$

$$u_{k+1} = u_k - \kappa L_0 u_k^\perp + P_2(x_{k+1} - x_k)$$

$$z_{k+1} = z_k + v\Lambda_k \lambda_k$$

$$\lambda_{k+1} = P_{R} \lambda_k - \alpha \Lambda_k \lambda_k = (\lambda_k - \alpha \Lambda_k \lambda_k + \tilde{\lambda} - (2\lambda x_{k+1} - x_k) + \lambda_0)(2x_{k+1} - z_k)).$$

where $x_0 \in \Omega$, $u_0 = P_1x_0$, $z_0 \in \mathbb{R}^m$, $\lambda_0 \in \mathbb{R}^m$.

The next result relates precisely iterates generated by Algorithm 1 or (4) to iterates generated by (6).

**Lemma 2**: Suppose Assumption 1 and 3 hold. Then, any sequence $(x_k, u_k, z_k, \lambda_k)$ generated by Algorithm 1 or (4) with initial conditions $x_0, u_0, z_0, \lambda_0 \in \mathbb{R}^N$ can be derived from some sequence $(x_k^*, u_k^*, z_k^*, \lambda_k^*)$ generated by (6) with initial conditions $x_0^* = x_0, u_0^* = P_1x_0, z_0^* = z_0, \lambda_0^* = \lambda_0$, as in

$$x_k = x_k^*, u_k = P_1(x_k^* + u_k^*), z_k = z_k^*, \lambda_k = \lambda_k^*.$$  (7)

**Proof**: The proof follows an induction argument. Due to the initial conditions, $P_1x_0 + u_0^\perp = P_1x_0 + P_2x_0 = x_0 = u_0$, hence (7) holds for $k = 0$.

Suppose (7) holds at step $k$. Then, from $z_{k+1}^* = \lambda_k^* + \nu\Lambda_k \lambda_k^*$ (cf. (6)) with $z_k = \lambda_k \lambda_k = \lambda_k$, it follows that $z_{k+1}^* = \lambda_k$. Next, using (7) in the right-hand side of the $x'$-update in (6), yields $x_{k+1} = x_{k+1}^*$.

Hence the first and third relations in (7) hold at step $k + 1$, and using them on the right-hand side of the $\lambda'$-update in (6) yields $\lambda_{k+1} = \lambda_{k+1}^*$, where $\lambda_{k+1}$ is generated by (6).

Lastly, we show the second relation in (7) at step $k + 1$. Thus,

$$P_1[x_{k+1}^* + u_{k+1}^*] = P_1[x_{k+1}^* + u_{k+1}^* - \kappa L_0 u_{k+1}^* + P_2(x_{k+1}^* - z_{k+1})] = P_1[x_{k+1} + u_k - \kappa L_0 u_k + P_2(x_{k+1} - x_k)],$$

where we used relations for $x$ at step $k + 1$ (cf. (8)) and for $u'$ at step $k$ (cf. (7)). Using $P_1[x_k + P_2x_k = x_k, L_0 P_1x_k = 0$ it follows that $P_1[x_{k+1} + u_{k+1}] = u_k - \kappa L_0 u_k + (x_{k+1} - x_k) = u_{k+1}$, cf. the $u$-update in (4), and the argument is complete.

**Lemma 3**: Suppose that Assumption 1 and 3 hold and let $\sigma_k = \{x_k, u_k^*, z_k, \lambda_k\}$, $A$, $B$, and $\Phi$ be defined as in (10). Suppose that $\Phi > 0$ and $\Phi^{-1}B$ is maximally monotone. Then the following hold:

(i): Iterates (4) are equivalently written as (9), i.e.,

$$A: x \mapsto \text{Null}(L_0) \quad \text{with} \quad \begin{bmatrix} 0 & -\frac{1}{\kappa} & L_0 & 0 \\ 0 & 0 & 0 & 0 \\ L_0 & \lambda & + & b \\ 0 & 0 & 0 & 0 \end{bmatrix} \approx$$

$$B: \Phi \mapsto \begin{bmatrix} 0 & \frac{1}{\kappa} & \frac{1}{2} P_{L_0}^{-1} & 0 \lambda^T \end{bmatrix} \approx$$

$$\Phi = \begin{bmatrix} \tau & 0 & -\frac{1}{2} P_{L_0}^{-1} & 0 \lambda \end{bmatrix} \approx$$

$$\begin{bmatrix} \tau & 0 & -\frac{1}{2} P_{L_0}^{-1} & 0 \lambda \end{bmatrix} \approx$$

where $\sigma = (x, u^*, z, \lambda) \in \mathbb{R}^{n \times C^\perp \times \mathbb{R}^m \times \mathbb{R}^m}$, for two operators $A$ and $B$ and matrix $\Phi$ defined as

$$0 \in A \sigma_k + B \sigma_{k+1} + \Phi(\sigma_k + \sigma_{k+1})$$

(9)

(10)

**Proof**: (i) The equivalence between (9) and (10) can be shown by expanding (9) with $A$, $B$ and $\Phi$ as in (10), cancelling terms and using $\tau = A \sigma_k + B \sigma_{k+1} + \Phi(\sigma_k + \sigma_{k+1})$.

Since $\Phi > 0$, by assumption, (9) is equivalent to $[\text{Id} + \Phi^{-1}B] \sigma_k = \Phi(\sigma_k - \frac{1}{\tau} \sigma_{k+1})$.

(11)

where $T_2 = (\text{Id} + \Phi^{-1}B)^{-1}$ and $T_1 = \text{Id} - \Phi^{-1}$. Furthermore, any such point $\sigma = \{x, u^*, z, \lambda\}$ satisfies $x = u^*$, $z^* = 0$, $\lambda = 1_N \otimes \Lambda$, where $x^*$ and $\Lambda$ satisfy the KKT conditions (9), hence $x^*$ is a variational GNE.
(Id − Φ−1A)(x̃) = 0 ∈ (B + A)(x̃). Thus, using (10), it follows that x̃ = col(x̃, ũ, ̄x, λ̃) satisfies:
0 ∈ N①(x̃) + F(x̃, P||x + ũ) + λT Aλ,
0 = LN ũ,
0 = Lλ, ̄x,
0 ∈ NR = (λ̃) + LA ̄x + αA ̄x + Lx̃.

By Assumption 3 the second and third relations imply that ũ = 0 (since ũ ∈ C⊥) and λ̃ = 1 ∈ λ+, for a λ+ ∈ Rm. Using these in the first and fourth relations together with P|| ̄x = σ(x̃), leads to 0 ∈ N①(x̃) + F(x̃, P||x + ũ) + λT Aλ+, which, using F(x̃, P||x + ũ) = F(x̃), is the first condition in (3). From 0 ∈ NR = (1 ∈ λ+) + b − Aλ̃ + LA ̄x, premultiplying by (T ⊗ I_m) as in the proof of Lemma 2(iii), the second condition in (3) holds hence x̃ = x∗ is a variational GNE.

Remark 2: Lemma 2 and Lemma 2 show that Algorithm 1 is related to a forward-backward iteration (9) or (10) [28], with a preconditioning matrix Φ. Note that A and B (10) each have a skew-symmetric part (monotone) and Φ is symmetric. The matrix Φ is needed, as pointed out in [24] and [26], to be able to distribute the backward step. Our proof techniques are similar to those in [26], however, there are significant differences. One of the reasons is that in the general setup in [26] the action is part of the estimate vector, while in our setup the actions and the aggregate estimate are separate and only consensus on the aggregate is needed. Due to this difference, the splitting in (10) is different than that used in [26]. The operators A and B as well as the metric matrix Φ have a different structure, with a new (skew-symmetric) block involving the projection matrix P|| which needs handled separately.

V. CONVERGENCE ANALYSIS

In this section we show convergence of Algorithm 1. Based on Lemma 2 its convergence can be established once convergence of (6) is shown. In turn, (6) is equivalent to (9) or (10), (cf. Lemma 3). Convergence of iteration (11) is guaranteed when Φ−1A is a cocoercive and Φ−1B is a monotone operator, cf. Theorem 25.8 in [28]. However, Φ is defined in terms of the extended pseudo-gradient F for which monotonicity properties are not guaranteed to hold on the augmented space of actions and aggregate estimates (only strong monotonicity of F(x) is assumed). Instead, we will show that Φ satisfies a restrictive cocoercive property and that B is maximally monotone, and then similar properties for Φ−1A and Φ−1B. This turns out to be sufficient to prove convergence because the restrictive property is with respect to the aggregate consensus subspace where the zeros of A + B lie, cf. Lemma 3(ii). To show the restrictive cocoercive property of Φ we balance the strong monotonicity of the pseudo-gradient F(x) (on the aggregate consensus space C) with that of the Laplacian on its orthogonal component C⊥, under a Lipschitz assumption on the extended pseudo-gradient F(x, u).

Assumption 4: The extended pseudo-gradient F(x, u) is Lipschitz continuous in both arguments, i.e., \( \exists l_F^u > 0, l_F^x > 0 \) s.t.
\[
\| F(x, u) - F(x, \bar{u}) \| \leq l_F^u \| u - \bar{u} \|, \quad \forall x, u, \bar{u} \in \mathbb{R}^n,
\]
\[
\| F(x, u) - F(x, \bar{u}) \| \leq l_F^x \| x - \bar{x} \|, \quad \forall x, u, \bar{x} \in \mathbb{R}^n.
\]

Assumption 4 is also used in other distributed (G)NE algorithms, [20], [31], [22].

The following lemma establishes a (restricted) monotonicity property on part of the A operator, (10), denoted ̄A.

Lemma 4: Let ̄A be defined as,
\[
̄A : (x, u^+) \mapsto \left[ F(x, P||x + u^+) \right] L_u u^+ - \right]

where \((x, u^+) \in \mathbb{R}^n \times C^⊥\). Suppose that Assumption 123 and 4 hold. Then, for any \((x, u^+)\) and any \((x, u^+)\) with \(u^+ = 0\),
\[
\left\langle x - x, A(x, u^+) - ̄A(x, u^+) \right\rangle \geq \mu_A \left\| x - x \right\|^2
\]
where \(\mu_A = \lambda_{\min} \left( \left[ \frac{\mu}{l_F^u} \frac{-l_F^x}{\lambda_2(L)} \right] \right)\).

Furthermore, ̄A is restricted monotone if \(\lambda_2(L) \geq \left(\frac{\|u\|^2}{4\mu}\right)\) and strongly monotone if the inequality is strict.

Proof: With Φ as in (12), for any \((x, u^+), (x, u^+)\) with \(u^+ = 0\), the left-hand side of (13) is written as
\[
\left\langle x - x, ̄A(x, u^+) - ̄A(x, u^+) \right\rangle = \left\langle x - x, F(x, P||x + u^+) - F(x, P||x) \right\rangle
\]
\[
+ \langle u^+, u^- \rangle L_u (u^+ - u^-) + \left\langle x - x, F(x, P||x) \right\rangle - \left\langle x - x, F(x, P||x) \right\rangle.
\]

Note that \(\left\langle x - x, F(x, P||x) \right\rangle - \left\langle x - x, F(x, P||x) \right\rangle \geq 0\), by \(F(x, P||x) = F(x, 1 \times \sigma(x)) = F(x)\) and Assumption 2. Also,
\[
\left\langle x - x, F(x, P||x) - F(x, P||x) \right\rangle \geq \lambda_2(L) \left\| u^+ - u^- \right\|^2,
\]
by Assumption 3. Using these inequalities into (14) yields
\[
\left\langle x - x, ̄A(x, u^+) - ̄A(x, u^+) \right\rangle \geq \lambda_2(L) \left\| u^+ - u^- \right\|^2,
\]
\[
\geq \mu_A \left\| x - x \right\| - \left\| x - x \right\|^2.
\]

Proof: (ii) B is written as the sum of two operators, one being a Cartesian product of normal cone operators, hence maximally monotone operator, and the other one being a skew-symmetric matrix, hence also maximally monotone, with full domain. Thus B itself is maximally monotone, [28].
(ii): With \( \pi = (x, u^t, z, \lambda) \), \( \bar{\pi} = (x, u^t, z, \lambda) \) and \( \bar{A} \) as in \( \text{(10)} \), we can write for \( \bar{A} \):

\[
\langle \pi - \bar{A} \bar{x} - \bar{A} \pi \rangle = \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right)
\]

\[
+ \frac{1}{2} \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right)^T \left[ \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right] + \frac{\lambda - \lambda | \lambda (\lambda - \lambda)|}{2d^2} \| \lambda (\lambda - \lambda) \|^2.
\]

(15)

The second term is 0 since the matrix is skew-symmetric, and for the third term, \( \lambda (\lambda - \lambda) \geq \frac{1}{2d^2} \| \lambda (\lambda - \lambda) \|^2 \), where \( d^2 \) is the maximum degree of graph \( G \). Using this with \( \text{(13)} \) in Lemma \( \text{4} \) from \( \text{(15)} \) it follows that for any \( u^t \in \mathbb{R}^n \), \( y^t = 0 \),

\[
\langle \pi - \bar{A} \bar{x} - \bar{A} \pi \rangle \geq \mu \lambda \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right)^T \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right) + \frac{\lambda - \lambda | \lambda (\lambda - \lambda)|}{2d^2} \| \lambda (\lambda - \lambda) \|^2.
\]

(16)

On the other hand, using \( \bar{A} \) as in \( \text{(12)} \), Lipschitz properties of \( F \), bounds on the eigenvalues of the Laplacian matrix \( L \) and the projection matrix \( P_L \), we obtain the following Lipschitz property

\[
\| \bar{A}(x, u^t, z, \lambda) - \bar{A}(x, u^t, z, \lambda) \|^2 \leq \theta^2 \| \bar{A}(x, u^t, z, \lambda) - \bar{A}(x, u^t, z, \lambda) \|^2
\]

where \( \theta^2 = \max((L^2)^t, (L^2)^t + 4d^2 + \frac{1}{2d^2}) \). Using this into \( \text{(16)} \) yields

\[
\langle \pi - \bar{A} \bar{x} - \bar{A} \pi \rangle \geq \mu \lambda \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right)^T \left( \begin{array}{c}
\frac{x - z}{u^t - u^t} \\
\frac{\pi - \bar{A}(x, u^t, z, \lambda)}{\bar{A}(x, u^t, z, \lambda)}
\end{array} \right) + \frac{\lambda - \lambda | \lambda (\lambda - \lambda)|}{2d^2} \| \lambda (\lambda - \lambda) \|^2.
\]

(16)

which, with \( \bar{A} \) as in \( \text{(10)} \), can be written as

\[
\langle \pi - \bar{A} \bar{x} - \bar{A} \pi \rangle \geq \max(\frac{\mu \lambda}{\theta^2}, \frac{1}{2d^2}) \| \bar{A} \bar{x} - \bar{A} \pi \|^2.
\]

The following lemma shows how agents can select step-sizes independently such that \( \Phi > 0 \). It can be easily proved based on the Schur complement and a diagonal dominance argument (Gershgorin’s theorem).

**Lemma 6:** Given any \( \delta > 0 \) and \( \kappa \leq \frac{1}{2} \), if step-sizes are selected such that

\[
\begin{align*}
\tau_i &\leq \frac{1}{\max_{j=1,\ldots,n} \left( \sum_{k=1}^{n} |A_{ik}|^2 \right) + \frac{4n(1-\kappa^2)}{4\kappa(1-\kappa)}} \quad \text{and} \\
v_i &\leq (2d_i + \delta)^{-1} \\
\alpha_i &\leq \frac{\max_{j=1,\ldots,n} \left( \sum_{k=1}^{n} |A_{jk}|^2 \right) + 2d_i + \delta^{-1}}
\end{align*}
\]

where \( d_i = |N_i| \), then the matrix \( \Phi > 0 \), and \( \Phi - \delta I \geq 0 \).

The next result shows that \( \Phi^{-1}B \) and \( \Phi^{-1}A \) satisfy a monotonicity property in the \( \Phi^{-1} \)-induced norm.

**Lemma 7:** Suppose Assumption \( \text{(12)} \) and \( \text{(3)} \) hold and \( \lambda(\mathcal{L}) > \frac{(\lambda(L)^t(\lambda(L)^t)^t)}{4d^2} \). Take any \( \delta > \frac{1}{\mathcal{L}} \) where \( \beta \) is as in \( \text{(3)} \) and step sizes \( \tau_i, \kappa, v_i \) and \( \alpha_i \) chosen to satisfy Lemma \( \text{6} \). Then under the \( \Phi^{-1} \)-induced norm \( \| \cdot \|_\Phi \), the following hold:

(i) \( \Phi^{-1}B \) is maximally monotone and \( T_\mathcal{L} = (I + \Phi^{-1}B)^{-1} \in \mathcal{Z}(\mathcal{L}) \).

(ii) \( \Phi^{-1}A \) is \( \beta \)-restricted coercive and \( T_\mathcal{L} = I - \Phi^{-1}A \) is restricted nonexpansive and further, the following holds for any \( \bar{x} = (x, u^t, z, \lambda) \) and any \( \pi = (\bar{x}, 0, z, \lambda) \):

\[
\| T_\mathcal{L} \bar{x} - T_\mathcal{L} \pi \|_\Phi \leq \| \pi - \bar{x} \|_\Phi - \| T_\mathcal{L} \bar{x} - T_\mathcal{L} \pi \|_\Phi.
\]

**Proof:** The proof is based on properties of \( A \) and \( B \) in Lemma \( \text{5} \) \( \Phi > 0 \) by Lemma \( \text{6} \) and resolvent properties for maximally monotone operators, \( \text{[28]} \), similar to Lemma \( \text{7} \) in \( \text{[25]} \) and Lemma \( \text{6} \) in \( \text{[26]} \).

(i): By \( \text{[28], Proposition 20.24.3} \), \( \Phi^{-1}B \) is maximally monotone when the Hilbert space is endowed with scalar product

\[
\langle (x, y) \rangle = \langle (x, y) \rangle.
\]

By \( \text{[28, Proposition 23.7]} \), \( (I + \Phi^{-1}B)^{-1} \) is firmly nonexpansive under the \( \Phi^{-1} \)-induced norm, hence \( (I + \Phi^{-1}B)^{-1} \in \mathcal{Z}(\mathcal{L}) \).

(ii): For \( \Phi^{-1}A \), note that by Lemma \( \text{5} \), for any \( \pi = (x, u^t, z, \lambda) \) and any \( \bar{x} = (x, u^t, z, \lambda) \) with \( u^t = 0 \),

\[
\langle \pi - \bar{x} \| \Phi^{-1}A \bar{x} - \Phi^{-1}A \pi \|_\Phi \geq \| \pi - \bar{x} \| \Phi^{-1}A \bar{x} - \Phi^{-1}A \pi \|_\Phi \geq \beta \| \pi - \bar{x} \|_\Phi.
\]

By Lemma \( \text{6} \) \( \Phi - \delta I \) is positive semi-definite, hence \( \lambda_{\max}(\Phi) \geq \lambda_{\min}(\Phi) \geq \lambda_{\max}(\Phi) \geq \lambda_{\min}(\Phi) \geq \lambda(\mathcal{L}) \geq 1/\mathcal{L} \).

Thus, \( \| \pi - \bar{x} \| \Phi^{-1}A \bar{x} - \Phi^{-1}A \pi \|_\Phi \geq (1/\mathcal{L}) \| \pi - \bar{x} \| \Phi^{-1}A \bar{x} - \Phi^{-1}A \pi \|_\Phi \geq \| \pi - \bar{x} \|_\Phi \).
converges to $\varpi = (\bar{x}, \bar{u}^0, \bar{\varepsilon}, \bar{\lambda})$ a fixed-point of $T_2 \circ T_1$, which by Lemma 3(ii), satisfies $x^* = x^*$, $u^* = 0$, $\lambda = 1N \otimes \lambda^*$, where $x^*$ is the variational GNE and $\lambda^*$ the corresponding multiplier. By Lemma 2 any $\{x_k, u_k, z_k, \lambda_k\}$ generated by Algorithm 1 or 4 converges, and, by 7, its limit is $(x^*, 1N \otimes \sigma(x^*), \bar{\varepsilon}, 1N \otimes \lambda^*)$, hence $\{x_k\}$ converges to $x^*$, the variational GNE.

Remark 4: The result in Theorem 4 provides conditions for fully distributed convergence of the GNE seeking algorithm on a single time-scale. We note that this is more challenging to establish than in a semi-decentralized setting where the true aggregate is provided by a coordinator. In a distributed setting, when each agent updates action and estimates the aggregate simultaneously, effectively each has to track the changing aggregate (which depends on the other agents’ changing actions) while also updating its own action. This means the algorithm operates on an augmented space of decisions and aggregate estimates. This feedback coupling introduces major difficulties, one of them being the lack of guaranteed monotonicity when extending the operators to an augmented space of decisions and aggregate estimates. By using a two time-scale operation (either via diminishing step-sizes, [20]), or by multiple communication rounds before action update, [21], [22]) this issue is circumvented and the analysis is simplified to only the aggregate consensus subspace. However, typically two time-scale operation means slow convergence. Herein, we achieve distributed convergence on a single time-scale by using operator-splitting techniques and balancing the lack of monotonicity of the pseudo-gradient off the consensus subspace by monotonicity properties of the Laplacian.

VI. NUMERICAL SIMULATIONS

In this section we consider a Nash-Cournot game over a network, as in [32] [20] [22], for a single market with production constraints and globally coupling market capacity constraints, where $N = 20$, $\Omega_i = [0, 200]$ $\forall i \in N$, $A = 1T$, $b = 400$. The cost function for each agent is $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)$, where $c_i(x_i) = 20 + 40(i - 1)|x_i|$ is the production cost and $f(x) = 1200 - \sum x_i$ is the demand price. The variational GNE is on the boundary at $x^* = [160, 120, 80, 40, 0, \ldots, 0]T$. Assume that agents communicate over a cycle communication graph. For simplicity for each agent parameters are taken the same $\tau = \frac{1}{20}, \kappa = \frac{1}{10}, \nu = \frac{1}{10}, \alpha = \frac{1}{10}$. The agents’ production plots are shown in Figure 1 for Algorithm 1. We compare the results with those of the algorithm in [22], in which each agent performs $\nu$ communication rounds to update the local multiplier, then updates its action, followed by another $\nu$ communication rounds to update its aggregate estimate. The algorithm in [22] converges to an $\epsilon$-GNE, which approaches the variational GNE if $\nu$ goes to infinity. For simulation we chose $\nu = 140$, which means that there are 280 communication rounds before an action update, unlike Algorithm 1 where only 2 communication rounds are needed per each action update. The agents’ production plots are shown in Figure 2, indicating that the algorithm converges slowly to an $\epsilon$-GNE that is quite far from the true Nash. Figure 3 shows plots of the normalized error, $\|x - x^*\| / \|x^*\|$ 100% of agents’ actions from the variational GNE for the two algorithms, indicating better convergence properties of Algorithm 1.

VII. CONCLUSION

In this paper we proposed a distributed algorithm that converges to a variational GNE for aggregative games with affine coupling constraints. The algorithm employs simultaneous action and aggregate estimate update, based on local communication, and uses fixed step-sizes. We proved its convergence by a forward-backward operator-splitting method for two preconditioned operators. We specifically exploited the invariance of the estimate average to show that the operators are restricted monotone. Among future work directions we can mention extension to time-varying and/or directed communication graphs.

APPENDIX

Proof of Lemma 7: (i) Note that from 4, using $\langle 1N \otimes I_n \big| L_{k-1} \rangle = 0^T$, it follows that $\langle 1N \otimes I_n \big| u_{k-1} \rangle =$
(1_\lambda \otimes I_n - u_k) - (1_\nu \otimes I_n - u_{k-1}) = \lambda^* - \nu^* + \lambda_\alpha + \nu_\sigma$, for all $k \geq 0$. Since $v_0 = x_0$, the first claim follows by induction, and the second one follows by using $u_k = \frac{1}{\lambda^* (I_\lambda \otimes I_n) u_k + \lambda^* (1_\lambda \otimes I_n) x_k}$.

(ii) Let $(\pi, \nu, \lambda)$ be a limit point of Algorithm 1 or 2. Then, $\pi = \pi + \sigma L_\pi$, i.e., $L_\pi \pi = 0$ which, by Assumption 3, implies that $\pi_i = \nu_i$ for all $\nu_i \in \mathbb{R}^n$. From the update for $u_k$ in (4) it follows that $\pi_i = \nu_i = \kappa L_\pi \pi_i + (\pi_i - \pi_i)$, i.e., $L_\pi \pi_i = 0$ which by Assumption 3 implies that $\pi_i = \nu_i$ for all $\nu_i \in \mathbb{R}^n$, hence $\sigma(\pi) = \nu$. Using part (i) it follows that $\nu = \sigma(\pi)$, i.e., in steady-state all agents have the same estimate equal to the action aggregate value. From the update of $x_k$ in (4) it follows that $\pi = P_1(\pi - \tau (F(\pi, \pi) + \lambda^*) \lambda^*)$. With $P_1 = (I_\pi + N_\pi)^{-1}$, this yields $0 \in \tau^{-1} N_1(\pi) + F(\pi, \pi) + \lambda^* \lambda^* \lambda^*$, where $N_1(x) = \Pi_{i=1}^n N_{1_i}(x)$. Since $\tau_i > 0$, $\tau^{-1} N_{1_i}(x) = N_{1_i}(x)$, and with $\pi_i = \sigma(x)$, $\lambda_i = \lambda^*$, $\forall i \in \mathcal{N}$, this yields component-wise, $0 \in \nabla x_i J_i(\pi, \sigma(x)) + \lambda^* \lambda^* \lambda^* \lambda^* \lambda^*$, $\forall i \in \mathcal{N}$, which is the first KKT condition. From the update for $\lambda_k$ in (4) it follows that, $\lambda_i = \lambda_{k-1} \lambda_i = \lambda^* - (\lambda^* - \lambda_i)$, or $\Omega_{\lambda n} = v - (\lambda^* - \lambda_i)$, for some $v = \sigma(\pi)$ with $v \in \Omega_{\lambda n}(\lambda^*)$, $\forall i \in \mathcal{N}$. Premultiplying by $(1_\lambda \otimes I_n)$ and using $(1_\lambda \otimes I_n) \lambda = 0$ (by Assumption 3) and $(1_\lambda \otimes I_n) \lambda = \lambda$, $\lambda(1_\lambda \otimes I_n) b = b$, yields $\Omega_{\lambda n} = \sum_{i=1}^N v_i = (\lambda^* - b) - (\lambda^* - \lambda_i)$, i.e., $0 \in \Omega_{\lambda n}(\lambda^*) \lambda^* \lambda^* \lambda^* \lambda^* \lambda^* \lambda^* \lambda^*$ (by Corollary 16.39 in [28]), which gives the second KKT condition. Therefore, $\pi = \pi^*$ is a variational GNE and $\lambda^*$ its multiplier.

□

REFERENCES

[1] W. Saad, Z. Han, P. H.V., and B. T., “Game theoretic methods for the smart grid,” IEEE Signal Processing Magazine, pp. 86–105, 2012.
[2] F. Parise, M. Colombino, S. Grammatico, and J. Lygeros, “Mean field constrained charging policy for large scale plug-in electric vehicles,” in IEEE Conf. Decision and Control, 2014, pp. 5101-5106.
[3] Z. Ma, S. Zou, L. Ran, X. Shi, and I. Hiskens, “Efficient decentralized coordination of large-scale plug-in electric vehicle charging,” Automatica, vol. 69, pp. 35–47, 2016.
[4] T. Alpcan, T. Başar, R. Srikant, and E. Altman, “CDMA uplink power control as a noncooperative game,” Wireless Networks, vol. 8, no. 6, pp. 659–670, 2002.
[5] J. Barrera and A. Garcia, “Dynamic incentives for congestion control,” IEEE Trans. on Automatic Control, vol. 60, no. 2, pp. 299-310, 2015.
[6] N. Li, L. Chen, and M. Dahleh, “Demand response using linear supply function bidding,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 1827–1838, 2015.
[7] H. Yin, U. Shanbhag, and P. G. Mehta, “Nash Equilibrium Problems With Scaled Congestion Costs and Shared Constraints,” IEEE Trans. on Automatic Control, vol. 56, no. 7, pp. 1702–1708, 2011.
[8] C. Tekin, M. Liu, R. Southwell, J. Huang, and S. Ahmad, “Atomic congestion games on graphs and their applications in networking,” IEEE Trans. on Networking, vol. 20, no. 5, pp. 1541–1552, 2012.
[9] K. Bimpikis, S. Ehsani, and R. Ilišić, “Cournot competition in networked markets,” Proc. of 15th ACM Conf. on Economics and Computation, p. 733, 2014.
[10] K. Arrow and G. Debreu, “Existence of an equilibrium for a competitive economy,” Econometrica, vol. 22, no. 3, pp. 265–290, 1954.
[11] T. Başar and G. Olsder, Dynamic Noncooperative Game Theory: Second Edition, ser. Classics in Applied Mathematics. SIAM, 1999.
[12] F. Facchinei and C. Kanzow, “Generalized Nash equilibrium problems,” Annals of Operat. Research, vol. 175, no. 1, pp. 177–211, 2010.
[13] F. Facchinei and J. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Springer New York, 2007.
[14] J. Koshal, A. Nedić, and U. V. Shanbhag, “A gossip algorithm for aggregative games on graphs,” in Proc. IEEE Conf. on Decision and Control, 2012, pp. 4840–4845.
[15] D. de Persis and S. Grammatico, “Continuous-time integral dynamics for monotone aggregative games with coupling constraints,” arXiv:1805.03270, 2018.
[16] 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.
[17] G. Belgioioso and S. Grammatico, “Semi-decentralized Nash equilibrium seeking in aggregative games with separable coupling constraints and non-differentiable cost functions,” IEEE Control Systems Letters, vol. 1, no. 2, pp. 400–405, 2017.
[18] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros, “Nash and Wardrop equilibria in aggregative games with coupling constraints,” IEEE Transactions on Automatic Control, pp. 1–1, 2018.
[19] G. Belgioioso and S. Grammatico, “A Douglas-Rachford splitting for semi-decentralized equilibrium seeking in generalized aggregative games,” Proc. IEEE Conf. Decision and Control, pp. 3541–3546, 2018.
[20] J. Koshal, A. Nedić, and U. V. Shanbhag, “Distributed Algorithms for Aggregative Games on Graphs,” Operations Research, vol. 64, pp. 680–704, 2016.
[21] J. Lei and U. V. Shanbhag, “Linearly convergent variable sample-size schemes for stochastic Nash games: Best-response schemes and distributed gradient-response schemes,” in Proc. IEEE Conf. on Decision and Control, 2018, pp. 3547–3552.
[22] F. Parise, B. Gentile, and J. Lygeros, “A distributed algorithm for average aggregative games with coupling constraints,” arXiv e-prints arXiv:1706.04634, 2017.
[23] S. Liang, P. Yi, and Y. Hong, “Distributed Nash equilibrium seeking for aggregative games with coupled constraints,” Automatica, vol. 85, pp. 179–185, 09 2017.
[24] P. Yi and L. Pavel, “A distributed primal-dual algorithm for computation of generalized Nash equilibria via operator splitting methods,” in Proc. IEEE Conf. on Decision and Control, 2017, pp. 3841–3846.
[25] ———, “An operator splitting approach for distributed generalized Nash equilibria computation,” Automatica, vol. 103, pp. 27–35, 2019.
[26] L. Pavel, “Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach,” arXiv e-prints arXiv:1808.04465, 2018.
[27] D. Gadkoy and L. Pavel, “Distributed GNE seeking over networks in aggregative games with coupled constraints via forward-backward operator splitting,” submitted to IEEE Conference on Decision and Control, 2019.
[28] H. Bauschke and P. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer New York, 2011.
[29] A. Kulkarni and U. Shanbhag, “On the variational equilibrium as a refinement of the generalized Nash equilibrium,” Automatica, vol. 48, no. 1, pp. 45–55, 2012.
[30] F. Salehisadaghian and L. Pavel, “Distributed Nash equilibrium seeking: A gossip-based algorithm,” Automatica, vol. 72, pp. 209–216, 2016.
[31] F. Salehisadaghian, W. Shi, and L. Pavel, “Distributed Nash equilibrium seeking under partial-decision information via the Alternating Direction Method of Multipliers,” Automatica, vol. 103, pp. 27–35, 2019.
[32] D. Gadko and L. Pavel, “A passivity-based approach to Nash equilibrium seeking over networks,” IEEE Transactions on Automatic Control, vol. 64, no. 3, pp. 1077–1092, 2019.