The END: Estimation Network Design for Games Under Partial-Decision Information

Mattia Bianchi, Member, IEEE, and Sergio Grammatico, Senior Member, IEEE

Abstract—Multiagent decision problems are typically solved via distributed iterative algorithms, where the agents only communicate among themselves on a peer-to-peer network. Each agent usually maintains a copy of each decision variable, while agreement among the local copies is enforced via consensus protocols. Yet, each agent is often directly influenced by a small portion of the decision variables only: neglecting this sparsity results in redundancy, poor scalability with the network size, and communication and memory overhead. To address these challenges, we develop Estimation Network Design (END), a framework for the design and analysis of distributed algorithms. END algorithms can be tuned to exploit problem-specific sparsity structures, by optimally allocating copies of each variable only to a subset of agents, to improve efficiency and minimize redundancy. We illustrate the END’s potential on generalized Nash equilibrium seeking under partial-decision information by designing new algorithms that can leverage the sparsity in cost functions, constraints, and aggregation values, and by relaxing the assumptions on the (directed) communication network postulated in the literature. Finally, we numerically test our methods on a unicast rate allocation problem, revealing greatly reduced communication and memory costs.

Index Terms—Nash equilibrium (NE) seeking, optimization algorithms, variational methods.

I. INTRODUCTION

ARGE-SCALE problems in machine learning [1], signal processing [2], and decentralized control [3] involve huge volumes of data, often spatially scattered. In these settings, distributed multiagent computation (where the computational burden is partitioned among a group of agents, only allowed to share information over a peer-to-peer communication network, without central gathering of the data) is emerging as a fundamental paradigm to enable scalability, privacy preservation, and robustness. On the downside, distributed algorithms require storage and transmission of multiple local copies of some variables, a form of redundancy that is absent in centralized processing. This may result in prohibitive memory and communication requirements and hinders scalability when the dimension of the copies grows with the network size.

For example, consider Nash equilibrium (NE) problems under partial-decision information [4], [5]. Each group of agents \( i \in \mathcal{I} := \{1, 2, \ldots, N\} \) has a private cost function \( f_i(x_i, x_{-i}) \), depending both on its action (decision variable) \( x_i \) and on the actions of the other agents \( x_{-i} = (x_j)_{j \neq i} \). The goal is to compute an NE, namely, a set of actions that simultaneously solves the coupled problems

\[
\text{minimize } f_i(x_i, x_{-i}) \quad \forall i \in \mathcal{I}. \tag{1}
\]

The challenge is that each agent can only receive data from some neighbors over a communication graph, although its cost depends on the actions of nonneighboring agents. As agent \( i \) does not have access to the full vector \( x_{-i} \), standard NE refinement procedures, such as the pseudogradient method

\[
x_{i}^{k+1} = x_{i}^{k} - \nabla_{x_{i}} f_i(x_{i}^{k}, x_{-i}^{k}), \quad i \in \mathcal{I}, k \in \mathbb{N} \tag{2}
\]

cannot be implemented distributedly. Instead, in this “partial-decision information” setup, it is typically assumed that each agent keeps (and exchanges with its neighbors, at each iteration) an estimate of the action of every other agent [4], [6], [7], [8], [9]; yet, this might be impractical, especially if the number of agents \( N \) is large. On the other hand, the cost of agent \( i \) often depends only on the actions of a much smaller subset of agents \( \mathcal{N}^i(i) \subset \mathcal{I} \), possibly of cardinality independent of \( N \) [10], e.g., in congestion control problems for wireless [11] and optical [12] networks. To highlight this dependence, let us write

\[
f_i(x_i, x_{-i}) = f_i((x_j)_{j \in \mathcal{N}^i(i)}). \tag{3}
\]

Unfortunately, the available NE seeking algorithms cannot take advantage of the sparsity structure in (3). The sole exception is the algorithms proposed by Salehisadaghiani and Pavel [10], [13], where each agent is only assigned proxies of the decisions that directly influence its cost—but provided that the communication network can be freely designed, which is, for example, not the case for ad hoc networks.

Similarly, in generalized Nash equilibrium (GNE) seeking (where the agents are also coupled via shared constraints), each agent is typically required to keep an estimate of all dual variables [6], [7], while possibly being directly affected by only a few of them. Methods to improve memory and communication efficiency in this setup are unknown in the literature.
problems are also postulated in the existing literature [22], [23] (specifically, we admit row stochastic graphs and do not rely on Perron eigenvectors). Furthermore, we allow for much more ductile estimate assignment compared to [10] and [13]. In particular, special cases of our method recover both the full- and partial-decision information setups (and a plethora of intermediate scenarios), for which a joint convergence analysis was not available. These theoretical results also demonstrate the potential of the END notation for analysis purposes, even in problems without any sparsity (see Section III-A).

2) For GNE problems, we study a novel class of aggregative games, generalizing that considered in [24]. We demonstrate that the number of copies allocated in the network, for both the aggregation function and the dual variables, can be reduced according to the problem sparsity. In particular, for the first time, we explicitly account for the possible sparsity in the coupling constraints (see Section III-B). Finally, we test our algorithm against its sparsity-unaware counterpart in a unicast rate allocation application. Our simulations show not only that communication and memory overhead are significantly reduced, but also that convergence speed and scalability are improved (see Section IV).

We focus on games under partial-decision information here; we refer the interested reader to the extended draft of this article [25] for numerical and theoretical results of END algorithms for multiagent optimization.

A. Background

1) Basic Notations: \( \mathbb{N} \) is the set of natural numbers, including 0. \( \mathbb{R} (\mathbb{R}_{\geq 0}) \) is the set of (nonnegative) real numbers. \( 0_q \in \mathbb{R}^q \) (\( I_q \in \mathbb{R}^{q \times q} \)) is a vector with all elements equal to 0 (1), \( 0_{q \times p} \in \mathbb{R}^{q \times p} \) is a matrix with all elements equal to 0, and \( I_q \in \mathbb{R}^{q \times q} \) is an identity matrix; the subscripts may be omitted to relax the assumptions on the communication network—thus, \( \lambda(A) \) denotes a vector of appropriate variables of interest that should be expressed and combined among the agents in any distributed algorithm (see Section II). The END notation has the following advantages.

1) Versatile and algorithm free: The variables of interest include any quantity some agents need to reach consensus upon (decision vectors or dual multipliers in optimization problems, the gradient of a cost, and an aggregative value), making END applicable to virtually any networked decision problem, including consensus optimization [19], game equilibrium seeking on networks, and common fixed-point computation [20].

2) High customizability yet unified analysis: The estimate allocation in END algorithms is a degree of freedom and can be tailored to specific problem instances, for example, by embedding efficiency criteria (e.g., minimal memory allocation and bandwidth constraints). As an advancement beyond the state of the art (that considers special sparsity patterns only, resorting to case-by-case analysis), END unifies the convergence proof of sparsity-unaware algorithms (see, e.g., [21]) with that of algorithms specifically devised for problems with a unique sparsity structure (see, e.g., [10]).

Specifically, in this article, we showcase the END framework on the topical problem of GNE seeking under partial-decision information. We analyze two END algorithms and discuss how they improve the literature in terms of both theoretical guarantees and communication/memory efficiency. Concretely, we have the following.

1) For NE problems, we prove the linear convergence of a pseudogradient algorithm over directed graphs. By leveraging the expressivity of the END framework, we are able to relax the assumptions on the communication network and later by Alghunaim et al. [14]. To fill this gap, we introduce Estimation Network Design (END), a rigorous graph-theoretic language to describe how the estimates of any variables of interest are allocated and combined among the agents in any distributed algorithm (see Section II). The END notation has the following advantages.

1) Versatile and algorithm free: The variables of interest include any quantity some agents need to reach consensus upon (decision vectors or dual multipliers in optimization problems, the gradient of a cost, and an aggregative value), making END applicable to virtually any networked decision problem, including consensus optimization [19], game equilibrium seeking on networks, and common fixed-point computation [20].

2) High customizability yet unified analysis: The estimate allocation in END algorithms is a degree of freedom and can be tailored to specific problem instances, for example, by embedding efficiency criteria (e.g., minimal memory allocation and bandwidth constraints). As an advancement beyond the state of the art (that considers special sparsity patterns only, resorting to case-by-case analysis), END unifies the convergence proof of sparsity-unaware algorithms (see, e.g., [21]) with that of algorithms specifically devised for problems with a unique sparsity structure (see, e.g., [10]).

Specifically, in this article, we showcase the END framework on the topical problem of GNE seeking under partial-decision information. We analyze two END algorithms and discuss how they improve the literature in terms of both theoretical guarantees and communication/memory efficiency. Concretely, we have the following.

1) For NE problems, we prove the linear convergence of a pseudogradient algorithm over directed graphs. By leveraging the expressivity of the END framework, we are able to relax the assumptions on the communication network
namely, if \((u, v) \in E\) whenever \((v, u) \in E\), we simply say that \(G\) is connected. The restriction of the graph \(G\) to a set of vertices \(V^A \subseteq V\) is defined as \(G|_{V^A} := (V^A, E \cap (V^A \times V^A))\). We also write \(G = (V^A, V^B, E)\) to highlight that \(G\) is bipartite, namely, \(G = (V, E)\) with \(V = V^A \cup V^B\) and \(E \subseteq V^A \times V^B\). We may associate to \(G\) a weight matrix \(W \in \mathbb{R}^{V \times V}\) compliant with \(G\), namely, \(w_{u,v} := [W]_{u,v} > 0\) if \((u, v) \in E\), \(w_{u,v} = 0\) otherwise; we denote by \(D = \text{diag}((deg(v))_{v \in V})\) and \(L = D - W\) the in-degree and Laplacian matrices, respectively, with \(deg(v) = \sum_{u \in V} w_{u,v}\). \(G\) is unweighted if \(w_{u,v} = 1\) whenever \((u, v) \in E\). Given two graphs \(G^A = (V^A, E^A)\) and \(G^B = (V^B, E^B)\), we write \(G^A \subseteq G^B\) if \(G^A\) is a subgraph of \(G^B\), i.e., if \(V^A \subseteq V^B\) and \(E^A \subseteq E^B\); we let \(G^A \cup G^B := (V^A \cup V^B, E^A \cup E^B)\). Given a graph \(G = (V, E)\) we may define the following problems:

1) **Steiner tree (ST) problem**, \(ST(G, T, W)\): Find an undirected connected subgraph \(G^* = (V^*, E^*) \subseteq G\), \(T \subseteq V^*\), with minimum cost, i.e., minimizing \(\sum_{(u,v) \in E^*} w_{u,v}\).

2) **Unweighted Steiner tree (UST) problem**, \(UST(G, T)\): Find an undirected connected subgraph \(G^* = (V^*, E^*) \subseteq G\), \(T \subseteq V^*\), with minimum number of nodes (equivalently, of edges).

3) **Euclidean Spaces**: Given a positive-definite matrix \(\mathbb{R}^{q \times q} \ni Q \succ 0\), \(H_q := (\mathbb{R}^q, (\cdot, \cdot)_Q)\) is the Euclidean space obtained by endowing \(\mathbb{R}^q\) with the \(Q\)-weighted inner product \((x, y)_Q = x^T Q y\) and \(\| \cdot \|_Q\) is the associated norm; we omit the subscripts if \(Q = I\). Unless otherwise stated, we assume to work in \(H = H_1\).

4) **Operator Theory**: A set-valued operator \(F : \mathbb{R}^q \rightharpoonup \mathbb{R}^q\) is characterized by its graph \(\text{gra}(F) := \{(x, u) | u \in F(x)\}\), \(\text{dom}(F) := \{x \in \mathbb{R}^q | F(x) \neq \emptyset\}\), \(\text{fix}(F) := \{x \in \mathbb{R}^q | 0 \in F(x)\}\) and \(\text{zer}(F) := \{x \in \mathbb{R}^q | 0 \in F(x)\}\). The inverse operator of \(F\), defined as \(\text{gra}(F^{-1}) = \{(u, x) | (x, u) \in \text{gra}(F)\}\), is \(\mu\)-strongly monotone if \((u - v, x - y) \geq 0 \geq \mu \|x - y\|^2\) for all \((x, u), (y, v) \in \text{gra}(F)\). \(F\) is an identity operator. For a set \(Q \subseteq \mathbb{R}^q\), \(N_Q : Q \rightharpoonup \mathbb{R}^q : x \mapsto \{v \in \mathbb{R}^q | \sup_{z \in Q} (v, z - x) \leq 0\}\) is the normal cone operator of \(Q\). If \(Q\) is closed and convex, then \((I + N_Q)^{-1} = \text{proj} Q\) is the Euclidean projection onto \(Q\).

### II. ESTIMATION NETWORK DESIGN

In this section, we introduce the general END framework and notation. We then leverage these tools to develop novel and improved GNE seeking algorithms in Section III.

#### A. END Setup

We start by defining a generic information structure, useful to describe existing distributed algorithms and to design new ones. It is characterized by:

1) a set of agents \(\mathcal{I} := \{1, 2, \ldots, N\}\);

2) a given (directed) communication network \(G^C = (\mathcal{I}, E^C)\), over which the agents can exchange information: agent \(i\) can receive information from agent \(j\) if and only if \(j \in N^C_i\);

3) a variable of interest \(y \in \mathbb{R}^n\), partitioned as \(y = \text{col}((y_p)_{p \in \mathcal{P}})\), where \(y_p \in \mathbb{R}^n\) and \(\mathcal{P} := \{1, 2, \ldots, P\}\);

4) a given bipartite directed interference graph \(G^I = (\mathcal{I}, E^I)\), \(E^I \subseteq \mathcal{P} \times \mathcal{I}\), that specifies which components of \(y\) are indispensable to each agent: \(p \in N^I_i\) means that agent \(i\) needs (an estimate of) \(y_p\) to perform some essential local computation.\(^1\)

The agents may be unable to access the value of the variable of interest \(y\). Instead, each agent keeps an estimate of some (possibly all) components \(y_p\),\(s\) and exchanges its estimates with some neighbors, as specified by:

5) a bipartite directed estimate graph \(G^E = (\mathcal{I}, E^E)\), \(E^E \subseteq \mathcal{P} \times \mathcal{I}\), that specifies which components of \(y\) are estimated by each agent: agent \(i\) keeps an estimate \(y_{p_i} \in \mathbb{R}^n\) of \(y_p\) if and only if \(p \in N^E_i\), and thus, \(\mathcal{N}^E(p)\) is the set of agents that keeps an estimate of \(y_p\);

6) \(\mathcal{P}\) directed design graphs \(\{G^D_p\}_{p \in \mathcal{P}}\), with \(G^D_p = (\mathcal{N}^E(p), \mathcal{E}^D_p)\), to describe how the agents exchange their estimates: agent \(i\) can receive \(y_{j,p}\) from agent \(j\) if and only if \(j \in \mathcal{N}^D_i(p)\).

Note that the vertices of \(G^D_p\) are \(\mathcal{N}^E(p)\), namely, only the agents that keep an estimate of \(y_p\) could receive an estimate of \(y_p\). Furthermore, note that the graph \(G^E\) is uniquely determined by \(\{G^D_p\}_{p \in \mathcal{P}}\) (but not vice versa).

**Example 1 (Partially separable costs):** Consider a multistage problem, where the agents can only exchange information over a communication network \(G^C\), and each agent \(i \in \mathcal{I}\) has a private cost function \(f_i(x)\), depending on a global decision vector \(x \in \mathbb{R}^n\), for instance, an optimization problem (where the goal is to minimize \(\sum_{i \in \mathcal{I}} f_i\) or a game as in (1)). Set \(\bar{x} = x\) and partition \(y = (y_p)_{p \in \mathcal{P}}\) in \(P\) components. In several engineering applications, like network control and data ranking [14], each cost function \(f_i\) depends only on some of the components of \(y\), as it can be specified by an interference graph \(G^I\): \(f_i\) depends on \(y_p\) if and only if \(p \in N^I_i\) \(\subseteq \mathcal{P}\). With some abuse of notation, we emphasize this fact by writing

\[
f_i(y) = f_i((y_p)_{p \in N^I_i}).
\]

In this setup, the usual approach to solve distributed optimization [19] and NE problems [4] is to assign to each agent \(i \in \mathcal{I}\) an estimate \(\hat{y}_i := \text{col}((y_{i,p})_{p \in \mathcal{P}}) \in \mathbb{R}^n\) of the whole decision variable and to let the agents exchange their estimates with every neighbor over \(G^C\). In END notation, we write this as

\[
E^E = \mathcal{P} \times \mathcal{I}, \quad G^D_p = G^C (\forall p \in \mathcal{P})
\]

(40)

We refer to (40) as the “standard” setup, since it is the usually studied case. This choice of graphs \(G^D_p\) and \(\{G^D_p\}_{p \in \mathcal{P}}\) ignores the possible structure in (4). Note that agent \(i\) only needs \((y_p)_{p \in N^I_i}\) to evaluate (the gradient of) its local cost \(f_i\). Therefore, storing an estimate of the whole vector \(y\) could be unnecessary and inefficient—especially if \(G^I\) is sparse and \(P\) is large. Are smarter choices of \(G^E\) and \(\{G^D_p\}_{p \in \mathcal{P}}\) possible?

On a side note, we mention that different partitions for the variable \(y\) are possible; for example, in the case of the game in

\(^1\)For ease of notation, we assume that \(\mathcal{N}^I(p) \neq \emptyset\) for all \(p \in \mathcal{P}\) (i.e., each component of \(y\) is indispensable to some agent).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
(1), one natural choice is to set $\mathcal{P} = \mathcal{I}$ and $y_p = x_p$ (the local action of agent $p$), for which (4) retrieves exactly (3).

**B. Flexible Design**

From an algorithm deployment perspective, the graphs $\mathcal{G}^C$ and $\mathcal{G}^I$ shall be considered fixed a priori. In contrast, the graphs $\mathcal{G}^E$ and $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ are a design choice, which determines how the estimates of certain variables are actually exchanged during the execution of a distributed algorithm. Informally speaking, one wishes to design the graphs $\mathcal{G}^E$ and $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ so that it is possible to *distributively* and *efficiently* solve a given decision problem. Mathematically, this translates to imposing extra structure on the estimate and design graphs.

**Design Problem 1**: Given the communication graph $\mathcal{G}^C$ and the interference graph $\mathcal{G}^I$, design the estimate graph $\mathcal{G}^E$ and the design graphs $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ such that:

i) $\mathcal{G}^I \subseteq \mathcal{G}^E$;

ii) $\mathcal{G}^D_p \subseteq \mathcal{G}^C$, for all $p \in \mathcal{P}$;

iii) “additional requirements” on $\mathcal{G}^E$ and $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ are met.

In particular, it must hold that $\mathcal{G}^E \subseteq \mathcal{G}^I$, namely, each agent estimates at least the components of $y$, which are indispensable for local computation (in fact, $\mathcal{G}^I$ expresses the minimal information necessary for each agent). Moreover, since data transmission can only happen over the communication graph $\mathcal{G}^C$, it must hold that $\mathcal{G}^D_p \subseteq \mathcal{G}^C$, for all $p \in \mathcal{P}$.

The “additional requirements” in Design Problem 1(iii) can encode not only *feasibility* conditions (seen as hard constraints: for instance, some type of connectedness for the graphs $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ is always needed to ensure that the agents can reach consensus on their estimates) but also *efficiency* specifications (treated as soft constraints, e.g., one might aim at reducing memory allocation by minimizing the number of copies of each variable used, provided that all hard constraints are satisfied).

A very simple design example is presented in Fig. 1. For continuity of presentation, we discuss several instances of Design Problem 1 in detail in Appendix A.

**Remark 1 (On the design cost)**: An optimal design of the graphs $\mathcal{G}^E$ and $\mathcal{G}^D_p$’s can be computationally demanding, especially in large-scale networks. We emphasize that such a design is still part of the tuning of an algorithm, to be subsequently used to solve an underlying decision problem. The question is thus whether the benefits of an efficient estimate allocation—in terms of algorithm execution—are worth the additional initial design effort.

For time-varying [27] and repeated problems, like distributed optimal estimation or model-predictive control [15] (where the same distributed problem is solved multiple times, but for different values of some parameters/measurements, see also Section IV), a careful a priori design can be advantageous, i.e., worth the initial (one-time) computational cost of solving Design Problem 1. Otherwise, especially if $\mathcal{G}^I$ is very dense, it may be convenient to settle for a suboptimal but readily available choice—e.g., (C0), which is the standard solution in literature [6], [28]. Finally, in many relevant applications, the specific problem structure renders the choice of optimal graphs $\mathcal{G}^D_p$ straightforward, as shown in Appendix A-4.

**C. Unified Analysis**

We emphasize that solving Design Problem 1 is not the goal of this article. In fact, the requirements and design procedure for $\mathcal{G}^E$ and $\{\mathcal{G}^D_p\}_{p \in \mathcal{P}}$ are vastly problem dependent. Fortunately, we do not need to consider a specific structure for the estimate and design graphs, but just assume that they satisfy some properties. First, we will assume throughout this article that the design graphs are chosen to satisfy the specifications in Design Problem 1(i) and (ii), without further mention.

**Standing Assumption 1 (Consistency)**: It holds that $\mathcal{G}^I \subseteq \mathcal{G}^E$ and that $\mathcal{G}^D_p \subseteq \mathcal{G}^C$ for all $p \in \mathcal{P}$.
Second, we will simply assume some level of connectedness for the design graphs; for example, we might assume that each \( G_p^D \) is strongly connected. This ensures some properties for the estimate exchange, akin to those exploited in the analysis of standard consensus-based algorithms, as exemplified in Lemmas 1 and 2. In turn, this simple observation allows one to easily generalize the convergence analysis of several distributed algorithms to the END framework, allowing for greater freedom in the estimate exchange design.

However, before going deeper into details, we need to introduce the stacked notation used in the rest of this article. This notation is particularly important in simplifying our analysis, as it allows us to abstract from the complex network interaction and to seamlessly cope with the nonhomogeneity of the agents’ copies (e.g., the local vectors kept by distinct agents may have different dimensions/number of components).

### D. END Notation

For all \( p \in \mathcal{P} \), let \( N_p := |\mathcal{N}_E(p)| \) be the overall number of copies of \( y_p \) kept by all the agents. We define
\[
y_p := \{(y_{i,p})_{i \in \mathcal{N}_E(p)} \in \mathbb{R}^{N_p \times n_p} \quad \forall p \in \mathcal{P} \}
\]
\[
y := \{(y_{p})_{p \in \mathcal{P}} \in \mathbb{R}^{n_p} \quad \forall p \in \mathcal{P} \}
\]
where we recall that \( y_{i,p} \) is the estimate of the quantity \( y_p \) kept by agent \( i \). \( n_p := \sum_{p \in \mathcal{P}} N_p n_{p} \). Note that \( y_p \) collects all the copies of \( y_p \), kept by different agents.

We denote
\[
W_D := \text{diag}((W_p \otimes I_{n_{p,p}})_{p \in \mathcal{P}})
\]
\[
L_D := \text{diag}((L_p \otimes I_{n_{p,p}})_{p \in \mathcal{P}})
\]
where \( W_D \) is the weight matrix of \( G_p^D \), and \( L_D \) is its Laplacian. For all \( p \in \mathcal{P} \), let
\[
C_p := \{y_p \in \mathbb{R}^{N_p \times n_p} : y_p = 1_{N_p} \otimes v, v \in \mathbb{R}^{n_p}\}
\]
\[
C := \prod_{p \in \mathcal{P}} C_p
\]
be the consensus space for \( y_p \), namely, the subspace where all the estimates of \( y_p \) are equal, and the overall consensus space, respectively. Given \( y := \{(y_p)_{p \in \mathcal{P}} \in \mathbb{R}^{n_p} \), we also define
\[
C(y) := \text{col}(\{(1_{N_p} \otimes y_p)_{p \in \mathcal{P}} \).}
\]
We denote by \( \mathcal{C} \), the complementary subspace of \( \mathcal{C} \), and by \( \Pi := \text{diag}(1_{N_p} 1_{N_p} \otimes I_{n_{p,p}}) \) the projection matrices onto \( \mathcal{C} \) and \( \mathcal{C} \), respectively.

For each \( p \in \mathcal{P} \), for each \( i \in \mathcal{N}_E(p) \), we denote by
\[
i_p := \sum_{j \in \mathcal{N}_E(p), j \neq i} 1
\]
the position of \( i \) in the ordered set of nodes \( \mathcal{N}_E(p) \). For all \( i \in \mathcal{N}_E(p) \), we denote by \( R_{i,p} \in \mathbb{R}^{n_p \times n_p} \) the matrix that selects \( y_{i,p} \) from \( y_p \), i.e., \( y_{i,p} = R_{i,p} y_p \).

Sometimes, it is useful to define agent-wise quantities, which we indicate with a tilde. Let
\[
y_{i} := \text{col}(\{(y_{i,p})_{p \in \mathcal{P}}_{E(i)} \} \quad \forall i \in \mathcal{I}
\]
\[
y := \text{col}(\{(y_{i})_{i \in \mathcal{I}}) \in \mathbb{R}^{n_p}
\]
where \( y_i \) collects all the estimates kept by agent \( i \). Let \( P \in \mathbb{R}^{n_p \times n_p} \) be the permutation matrix such that \( P y = \bar{y} \); the graph structures corresponding to \( \bar{y} \) can be defined via permutations, e.g., \( W_D := P W_{D} P^T \), \( L_D := P L_{D} P^T \), and \( \mathcal{C} := \mathcal{P} \).

The following lemmas easily follow by stacking over \( p \in \mathcal{P} \) well-known graph-theoretic properties.

**Lemma 1:** Assume that, for each \( p \in \mathcal{P} \), there exists a root \( r_p \in \mathcal{I} \) such that \( G_p^D \) is rooted at \( r_p \). Then, \( \text{null}(L_D^p) = \mathcal{C} \).

Moreover, \( \text{null}(L_D^B) = \mathcal{C} \).

**Lemma 2:** Assume that \( G_p^D \) is strongly connected and that \( W_D^p \) is balanced, for all \( p \in \mathcal{P} \). Then, for any \( y \in \mathbb{R}^{n_p}, \langle y, L_D^p y \rangle \geq \frac{\lambda}{2} \| (1 - \Pi) y \|^2, \) where \( \lambda := \min_{p \in \mathcal{P}} \{ \lambda_2 (L_D^p + L_D^p) \} > 0 \).

### III. GNE SEEKING

In this section, we consider GNE problems. In particular, each agent \( i \in \mathcal{I} \) is equipped with a private cost function \( f_i(x_i, x_{-i}) \), \( f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_{-i}} \to \mathbb{R} \), which depends both on its local action (decision variable) \( x_i \in \mathbb{R}^{n_i} \) and on the actions of the other agents \( x_{-i} := \text{col}((x_j)_{j \in \mathcal{I}, j \neq i}) \in \mathbb{R}^{n_{-i}} \). Each agent chooses its action in a local feasible set \( \Omega_i \subseteq \mathbb{R}^{n_i} \); let \( x := \text{col}((x_j)_{j \in \mathcal{I}}) \in \Omega \) be the overall action, with \( \Omega := \prod_{i \in \mathcal{I}} \Omega_i \subseteq \mathbb{R}^{n_i} \). The agents’ decisions are also coupled via shared constraints: specifically, the overall feasible set is \( \mathcal{X}_i := \Omega \cap \{ x \in \mathbb{R}^{n_i} | Ax \leq a \} \), where \( A \in \mathbb{R}^{n_{o} \times n_i}, a \in \mathbb{R}^{n_{o}} \). We call generalized game (GNE) a set of interdependent optimization problems:
\[
(\forall i \in \mathcal{I} \) minimize \( f_i(x_i, x_{-i}) \) s.t. \( x_i, x_{-i} \in \mathcal{X} \).
\]

The goal is to distributionally compute a GNE, a set of decisions simultaneously solving all the problems in (15).

**Definition 1:** A GNE is an \( \mathcal{N} \)-tuple \( x^* = \text{col}((x_i^*)_{i \in \mathcal{I}}) \in \mathcal{X} \) such that, for all \( i \in \mathcal{I} \), \( f_i(x_i^*, x_{-i}^*) \leq \inf_{x_i} \{ f_i(x_i, x_{-i}^*) | (x_i, x_{-i}^*) \in \mathcal{X} \} \).

Let us define the pseudogradient operator \( F : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x} \), as follows:
\[
F(x) := \text{col}(\langle \nabla x_i f_i(x_i, x_{-i}) \rangle_{i \in \mathcal{I}}).
\]

We restrict our attention to convex and strongly monotone games; the following are standard conditions for GNE seeking over graphs [6, Asm. 1 and 2], [23, Asm. 1 and 2], [29, Asm. 1].

**Assumption 1:** For all \( i \in \mathcal{I}, \Omega_i \) is closed and convex, \( f_i \) is continuous, and \( f_i(\cdot, x_{-i}) \) is convex and differentiable for every \( x_{-i} \); \( \mathcal{X} \) is nonempty and satisfies Slater’s constraint qualification (i.e., the relative interior of \( \mathcal{X} \) is nonempty).

**Assumption 2:** The pseudogradient \( F \) in (16) is \( \mu \)-strongly monotone and \( \theta \)-Lipschitz continuous, for some \( \mu, \theta > 0 \).

As per standard practice, we only focus on variational GNEs (v-GNEs), namely GNEs with identical dual variables, which are computationally tractable and economically more justifiable [2]. Under Assumptions 1 and 2, there is a unique v-GNE [30, Th. 2.3.3]; moreover, \( x^* \) is the v-GNE of the game in (15) if and only if there exists a dual variables \( \lambda^* \in \mathbb{R}^{n_o} \) satisfying the following Karush–Kuhn–Tucker (KKT) conditions [2, Th. 4.8] (we recall that \( N_{\mathcal{S}} \) denotes the normal cone of a set \( \mathcal{S} \))
\[
0_{n_x} \in F(x^*) + N_\mathcal{X}(x^*) + A^\top \lambda^* \tag{17a}
\]
\[
0_{n_x} \in - (Ax^* - a) + N_{\mathbb{R}^{n_o}} \tag{17b}
\]
Finally, we consider the so-called partial-decision information scenario, arising in applications without a central coordinator, such as radio networks [31] and sensor positioning [5]. In particular, each agent $i$ only relies on the data received locally from some neighbors over a communication network $G^P = (I, E^P)$. To cope with the limited information, the solution usually explored in the literature is to embed each agent with an estimate of the whole vector $x$ [4], [23] and possibly a copy of a dual variable [6]. Critically, this approach fails to exploit the possible sparsity in the cost and constraint coupling. We remedy this in the rest of this section.

### A. END Pseudogradient Dynamics for NE Seeking

We first consider games without coupling constraints (i.e., $X = \Omega$): then, the notion of v-GNE boils down to that of an NE. In this subsection, we choose the variable of interest for the END framework to be the overall action, i.e., $y = x$, $P = I$, and $y_i = x_i$ for all $i \in I$ (finer partitions are also possible). Hence, we can describe the cost coupling via an interference graph $G^E = (P, I, E^I)$, where $(p, i) \in E^I$ if and only if $f_i$ depends on $x_p$, for all $i \neq p$, and $(i, i) \in E^I$ for all $i \in I$; we also write

$$f_i((x_p)_{p \in N(i)}) := f_i(x_i, x_{-i}).$$

Assume that an estimate graph $G^E$ and design graphs $G^D$s are chosen according to Assumption 1. Then, each agent $i$ keeps and sends the copies $\{y_{i,p}, p \in N^E(i)\}$, estimating the actions of a subset of the other agents. Since the action $x_i$ is actually a local variable, under the control of agent $i$, we formally define $y_{i,i} := x_i$ (i.e., agent $i$’s estimate of its own action coincides with the real value). We study the following iteration (we recall the notation in Section II-D), each agent $i \in I$ performs:

$$\begin{align*}
y_i^{k+1} & := \sum_{j \in N^P(i)} |W^D|_{p,p} y_{j,p}^k \\
y_{i,p}^{k+1} & := y_{i,p}^k \\
y_{i,i}^{k+1} & := \text{proj}_{\Omega} y_i^k - \alpha \nabla_x f_i((y_{i,p}^k)_{p \in N^E(i)})
\end{align*}$$

In (18), the estimates of the agents are updated according to a consensus protocol, with an extra (projected) gradient step for the own estimate $y_{i,i}$. The algorithm boils down to [32, Alg. 1] if $W^D = W^C$ for all $i \in I$, [i.e., the standard setup in (C0)]. Since $G^P \subseteq G^E$, the algorithm is distributed. Note that the local gradient $\nabla_x f_i$ is computed on the local estimates kept by agent $i$, not on the real action $x^k = \text{col}(y_{i,i}^k)_{i \in I}$. We define the extended pseudogradient mapping

$$F(y) := \text{col}(\nabla_x f_i((y_{i,p})_{p \in N^E(i)})_{i \in I}).$$

R := diag((R_{i,i})_{i \in I}), with $R_{i,i}$ as in Section II-D (i.e., $R y = y$), and $\Omega := \{y \mid R y \in \Omega\}$. Then, (18) can be written in stacked form in one line

$$y^{k+1} = \text{proj}_{\Omega} (W^D y^k - \alpha R^T F(W^D y^k)).$$

We now postulate additional conditions on the design graphs.

### Assumption 3:

For each $i \in I$, $G^D$ is rooted at $i$ and $W^D_i 1_{N(i)} = 1_{N(i)}$; we denote by $q_i^D$ the unique nonnegative vector such that $q_i^D W^D_i = q_i^D, 1_{N(i)}^T q_i^D = 1$.

Assumption 3 is very mild: rootedness is necessary for the consensus of the estimates; row stochasticity is immediately satisfiable whenever the agents access to their own in-degree. One may suppose that some restrictively strongly connectedness assumption is that the (Perron) eigenvectors $q_i^D$s have zero entries. In addition, we require one technical condition.

### Assumption 4:

For all $i \in I$, there is a matrix $Q_i > 0$ such that $\sigma_i := ||W^D_i - 1_{N(i)} q_i^D||_{q_i} < 1$, $|1^T Q_i|_{i,i} = 1$, $1_{N(i)}^T W^D_i (1_{N(i)} - 1_{N(i)} q_i^D) = 0_{N(i)}$, and either (i) $Q_i$ is diagonal or (ii) $Q_i = R_{n \times n}$.

### Remark 2:

Assumption 4(i) alone is general enough to comprise all the cases considered in the existing literature [corresponding to the choice $G^D = G^E$ in (CO)].

i) If $G^D$ is strongly connected with self-loops, then Assumption 4(i) holds with $Q_i = \text{diag}(q_i, Q_i)$, [22, Lem. 1]; in particular, if $W^D$ is doubly stochastic, $Q_i = I$. This also means that, if $G^C$ is strongly connected, one can always choose the design graphs to satisfy Assumptions 3 and 4.

ii) If $G^D$ is the directed star graph (namely, there are all and only the edges from node $i$ to every node in $N^E(i)$), then Assumption 4(i) holds with $Q_i = I$ and $\sigma_i = 0, q_i$ with only one nonzero element $[q_i]_{i,i} = 1$; note that having this structure for all $i \in I$ corresponds to the classical full-information scenario, as detailed later.

Other relevant cases, satisfying Assumption 4 but never addressed in the literature, are discussed after the next result.

### Theorem 1:

Let Assumptions 1–4 hold, and let

$$\begin{align*}
\Xi := \text{diag}(Q_i \otimes 1_{n_{i,j}})_{i \in I}, \\
\sigma := \max_{i \in I} \{\sigma_i\},
\end{align*}$$

$$\begin{align*}
\theta := \theta \sqrt{\max_{i \in I} (Q_i)_{i,i} / \lambda_{\min}(\Xi)},
\gamma := \sqrt{1 / \max_{i \in I} 1^T Q_i 1},
\end{align*}$$

$$M_\alpha := \begin{bmatrix}
1 - 2\alpha \mu \gamma^2 + 2\alpha^2 \theta^2 \hat{\gamma}^2 & \hat{\sigma}(\alpha(\hat{\theta}+\hat{\theta})) + \alpha \hat{\sigma} \gamma^2 \\
\hat{\sigma}(\alpha(\hat{\theta}+\hat{\theta})) + \alpha \hat{\sigma} \gamma^2 & \hat{\sigma}^2 (1 + 2\alpha \hat{\theta} + \alpha \hat{\theta}^2)
\end{bmatrix}.$$

Let $\alpha > 0$ be chosen such that

$$\rho_\alpha := \lambda_{\max}(M_\alpha) < 1.$$  

Then, the sequence $(y^k)_{k \in \mathbb{N}}$ generated by (20) converges linearly to $y^* := C(x^*)$, where $x^*$ is the NE of the game (15): for all $k \in \mathbb{N}$, $||y^{k+1} - y^*||_2^2 \leq \rho_\alpha ||y^k - y^*||_2^2$.

### Proof:

See Appendix B-A.

The condition (21) always holds for $\alpha$ small enough (explicit bounds are obtained as in [32]). Let us now highlight some of the novelties of Theorem 1.

1) Consider the standard scenario in (C0), where agents store and exchange an estimate of the whole $x$. If $G^C$ is strongly connected and $W^C$ is doubly stochastic, Theorem 1 retrieves exactly [32, Th. 1]. If $W^C$ is only row stochastic,
Theorem 1 improves on the results in [22] since (20) does not require the knowledge of any Perron eigenvector, but just a small-enough step (as in [33], which however requires a much stronger monotonicity assumption); this is achieved by using the weight matrix Ξ in the analysis.

2) To our knowledge, the only other works that consider partial coupling are [10] and [13]. The authors propose gossip algorithms where, assuming a lower bound for the strongly connected graph G^C related to G^f, each agent must only estimate the actions that directly affect its cost. However, this setup requires that the cost of each agent only depends on the actions of its communication neighbors and neighbors’ neighbors [13, Lem. 3]. By allowing some agents to estimate a larger subset of actions (if needed), Theorem 1 avoids this limitation.

3) Theorem 1 also allows for graphs G^D that are not strongly connected. For instance, if {G^D_i}$_{i \in I}$ are all star graphs [i.e., Remark 2(ii)], the action update in (20) is

\[ x_i^{k+1} = \text{proj}_{\lambda}(x_i^k - \alpha \nabla x_i f_i(x_i^k)) \]  

(22)

which is the standard pseudogradient method for the full-information scenario, where estimates and true values coincide. In particular, when G^D is complete, Theorem 1 retrieves the well-known bound α < 2μ/β^2 (since σ = 0, \( \lambda^2 = \lambda^2 = \frac{1}{N} \) [34, Prop. 26.16].

4) Another (not strongly-connected) case not addressed before is that of a matrix W^D_1 = \begin{bmatrix} 1 & 0_{N^N} \end{bmatrix}, c \in \mathbb{R}^{N^N}$,$N^N$, representing a leader–follower protocol (with agent 1 as the leader for ease of notation). If \( \Omega_1 = \mathbb{R}^{N^N}$, and Assumption 3 holds, it can be checked that Assumption 4 is verified with \( q^D_1 = c = \text{col}(1, 0, \ldots, 0)$ and \( Q_1 = \begin{bmatrix} 1 + 1 & -1 \end{bmatrix}$, where \( X = \begin{bmatrix} x_1 \end{bmatrix}$,$x_2 \end{bmatrix}$,$x_3 \end{bmatrix}$, null, and \( \text{proj}_{\lambda}(\cdot) \) is the projection onto \( \lambda \). Therefore, we can model a scenario in which the exact information on \( x_1 \) received by agent 1 when evaluating its cost. We can also choose a directed tree for \( G^D_2 \) but with some delay, by choosing \( G^D_3 \) as a directed tree.

5) Each graph G^D can be chosen independently. For instance, one variable \( x_1 \) might be publicly available (choose G^D_1 as a star graph), while other actions can only be reconstructed via consensus. The convergence of these cases would otherwise require ad hoc analysis (e.g., to determine bounds on the step sizes).

B. GNE Seeking in Aggregative Games

Next, we also address the presence of the coupling constraints \( Ax \leq a \). These are partitioned in row blocks, i.e., \( A \leq [A_{m,i}]_{m \in \mathcal{M}, i \in \mathcal{I}} \) and \( a = \text{col}(\sum_{i \in \mathcal{I}} a_{m,i})_{m \in \mathcal{M}} \), where, for all \( m \in \mathcal{M} := \{1, 2, \ldots, M\} \), \( A_{m,i} \in \mathbb{R}^{n_{m,i} \times n_i} \) and \( a_{m,i} \in \mathbb{R}^{n_{m,i}} \) are local data kept by agent \( i \). The coupling constraint sparsity pattern is described by the interference graph \( G^{\ell} = (\mathcal{M}, \mathcal{I}, \mathcal{E}^{\ell}) \), where \( (m, i) \in \mathcal{E}^{\ell} \) if agent \( i \) is involved in the constraints block indexed by \( m \); in other terms

\[ (\forall (m, i) \notin \mathcal{E}^{\ell}) \quad A_{m,i} = 0, a_{m,i} = 0 \]  

(23)

and the \( m \)th block constraint can be written as \( \sum_{i \in \mathcal{I}} \alpha_{m,i} x_i - a_{m,i} \leq 0 \). Correspondingly, we partition the dual variable as \( \lambda = \text{col}((\lambda_m)_{m \in \mathcal{M}}) \) and \( \lambda_m \in \mathbb{R}^{n_{m,i}} \).

Furthermore, we focus on the particularly relevant class of aggregative games [29], [35], where the agents only need to reconstruct an aggregation value (usually of dimension independent of \( N \), e.g., the average of all the actions [29]) to evaluate their objective functions. In particular, the cost coupling arises via an aggregation mapping \( \sigma : \mathbb{R}^{n_N} \to \mathbb{R}^{n_N} \), so that for all \( i \in \mathcal{I} \) and for some function \( f_i \), we have

\[ f_i(x_i, x_{-i}) = f_i(x_i, \sigma(x_i)) \]  

(24)

Let \( \sigma \) be partitioned as \( \sigma = \text{col}((\sigma_q)_{q \in \mathcal{Q}}), \mathcal{Q} := \{1, 2, \ldots, Q\} \), and let \( G^{\ell} = (\mathcal{Q}, \mathcal{I}, \mathcal{E}^{\ell}) \) be an interference graph such that, for all \( x \) and (with the usual overloading)

\[ \sigma_q(x) = \sigma_q(x_i)_{i \in \mathcal{N}^q}, \quad \sigma(x) = \text{col}((\sigma_q)_{q \in \mathcal{Q}}) \]  

(25)

\[ f_i(x_i, \sigma(x)) = \bar{f}_i(x_i, (\sigma_q(x))_{q \in \mathcal{N}^q}) \]  

(26)

namely, \( (q, i) \in \mathcal{E}^{\ell} \) whenever either \( \sigma_q(x) \) explicitly depends on \( x_i \) or \( \bar{f}_i(x_i, \sigma(x)) \) explicitly depends on \( \sigma_q \). We restrict the analysis to affine aggregation functions, so that

\[ \sigma_q(x) := \sum_{i \in \mathcal{N}^q} \alpha_{q,i} x_i + b_{q,i} \]  

(27)

\[ \sigma(x) = \text{col}((\sigma_q)_{q \in \mathcal{Q}}) = B x + b \]  

(28)

\[ B_{q,i} \in \mathbb{R}^{n_{q,i} \times n_i}, \quad b_{q,i} \in \mathbb{R}^{n_{q,i}} \]  

being local data of agent \( i \). B := \( [B_{q,i}]_{q \in \mathcal{Q}, i \in \mathcal{I}} \), \( b = \sum_{i \in \mathcal{I}} \text{col}(l_{q,i})_{q \in \mathcal{Q}} \), with

\[ (\forall (q, i) \notin \mathcal{E}^{\ell}) \quad B_{q,i} = 0, b_{q,i} = 0 \]  

(29)

If \( M = Q = 1 \), we recover the standard generalized aggregative games [7], [36], where each agent must estimate the whole dual variable \( \lambda \in \mathbb{R}^n \) and the whole aggregative value \( \sigma(x) \in \mathbb{R}^{n_N} \). Instead, our idea is to leverage the possible problem sparsity by assigning to each agent copies of only some of the components of the dual variable and of the aggregation function, as specified by two aggregative estimate graphs \( \mathcal{G}^{E,\lambda} = (\mathcal{M}, \mathcal{I}, \mathcal{E}^{E,\lambda}) \), \( \mathcal{G}^{E,\sigma} = (\mathcal{Q}, \mathcal{I}, \mathcal{E}^{E,\sigma}) \). We postpone the discussion of a motivating application to Section IV.

3Here, we consider ENDC problem for two variables of interest \( \lambda \) and \( \sigma \). The notation in Section II is recovered by setting \( y = \text{col}(\sigma_q(x), \sigma(x)) \), with \( P = M + Q \). However, for readability, here, we treat the two variables separately, as two independent instances of ENDC, where the corresponding quantities in Section II-D are distinguished via the superscripts \( \lambda \) and \( \sigma \) (e.g., \( G^{\ell}_{\lambda,\sigma} \) and \( b^{\ell}_{\lambda,\sigma} \) describe how the agents exchange the estimates of \( k_m^\lambda \) and \( k_m^\sigma \), respectively). We also indicate the estimates of \( k_m^\lambda \) and \( k_m^\sigma \) (kept by agent \( i \) with \( \lambda_m^\lambda \) and \( \lambda_m^\sigma \) in place of \( y_{m,i}^\lambda \) and \( y_{m,i}^\sigma \) analogously for the stacked vectors—e.g., \( \sigma_q = \text{col}((\sigma_{q,i})_{q \in \mathcal{Q}, i \in \mathcal{I}}) \), \( \sigma = \text{col}((\sigma_q)_{q \in \mathcal{Q}}) \) and \( k_m = \text{col}((k_{m,i})_{i \in \mathcal{I}}) \), \( \lambda = \text{col}((k_m)_{m \in \mathcal{M}}) \))
Let us define the extended pseudogradients mapping
\[ \tilde{F}(x, \sigma) := \text{col}(\nabla_x \tilde{f}_i(x, \sigma_i) + \tilde{B}_i \nabla_{\sigma_i} \tilde{f}_i(x, \sigma_i))_{i \in \mathcal{E}} \]  
where \( \tilde{B}_i := \text{col}(\{B_{q,i} \}_{q \in \mathcal{N}_{E, \sigma(i)}} \), and we recall that \( \tilde{\sigma}_i \) are the estimates of the components of \( \sigma \) kept by agent \( i \) and (with the customary overloading) \( \tilde{f}_i(x_i, \sigma_i) := f_i(x_i, \sigma_i) \). Note that \( \tilde{\sigma}_{i} \) coincides with the pseudo-gradient mapping when the estimates are exact and at consensus, i.e., \( \tilde{F}(x, \hat{\sigma}(\sigma(x)) = F(x) \). We study the following distributed iteration:
\[ x^{k+1} = \text{proj}_{\Omega}(x^k - \beta (\alpha \tilde{F}(x^k, \sigma^k) + B^T L D, \sigma \sigma^k + A^T \lambda^k)) \]  
\[ \sigma^{k+1} = \sigma^k - \beta L D, \sigma \sigma^k + B(x^{k+1} - x^k) \]  
\[ z^{k+1} = z^k + \beta L D, \lambda \]  
\[ \lambda^{k+1} = \text{proj}_{\Omega}(\lambda^k - \beta (\lambda^{D, \lambda}(2x^{k+1} - x^k) - A(2x^{k+1} - x^k) + a)) \]  
(31d)
where \( \alpha, \beta > 0 \) are step sizes; \( z = \text{col}(\{z_{m} \}_{m \in \mathcal{E}}) \in \mathbb{R}^{m} \) and \( z_{m} = \text{col}(\{z_{m,i} \}_{i \in \mathcal{N}_{E, \lambda}(m)}) \), where, for all \( m \in \mathcal{N}_{E, \lambda}(i) \), \( z_{i,m} \in \mathbb{R}^{m} \) is an auxiliary dual variable kept by agent \( i \);
\[ A := P^T \text{diag}(\{\tilde{A}_i \}_{i \in \mathcal{E}}), \quad \tilde{A}_i := \text{col}(\{A_{m,i} \}_{m \in \mathcal{N}_{E, \sigma(i)}}) \]  
\[ a := P^T \text{diag}(\{\tilde{a}_i \}_{i \in \mathcal{E}}), \quad \tilde{a}_i := \text{col}(\{a_{m,i} \}_{m \in \mathcal{N}_{E, \lambda(i)}}) \]  
\[ B := \text{diag}(\{N_q^T I_{n_q} \}_{q \in \mathcal{L}}) \text{P}^\sigma \text{diag}(\{\tilde{B}_i \}_{i \in \mathcal{E}}) \]  
\[ b := \text{diag}(\{N_q^T I_{n_q} \}_{q \in \mathcal{L}}) \text{P}^\sigma \text{col}(\{\tilde{b}_i \}_{i \in \mathcal{E}}) \]  
\[ \tilde{b}_i := \text{col}(\{b_{q,i} \}_{q \in \mathcal{N}_{E, \sigma(i)}}) \]  
and we recall that \( N_q^\sigma = [\nabla_{E, \sigma}(q)] \), that with \( \sigma \), we indicate agent-wise stacked quantities, and that \( P^\lambda \) and \( P^\sigma \) are permutation matrices (i.e., \( P^\lambda \lambda^\sigma = \lambda \) and \( P^\sigma \lambda^\lambda = \lambda \)). We impose
\[ \sigma^0 = Bx^0 + b \]  
(32)
or, agentwise, \( \sigma^0 = \text{col}(\{N_q^\sigma B_{q,i} x^0_i + N_q^\sigma b_{q,i} \}_{q \in \mathcal{N}_{E, \sigma(i)}}) \); the other variables are initialized arbitrarily.

The algorithm is based on primal–dual pseudogradients dynamics, where the update of each \( \sigma_q \) represents a dynamic tracking of the aggregation function \( \sigma_q \) over the graph \( G^D, \sigma \). It is inspired by the methods in [6] and [36], and as in these works it is derived as a forward–backward (FB) method [34, Sec. 26.5].

**Assumption 5:** For each \( m \in \mathcal{M} \), \( G^m, D, \lambda \) is undirected and connected and \( W_{D, m}^{\lambda} \) is symmetric. For each \( q \in Q \), \( G^D, \sigma \) is strongly connected and \( W_{D, m}^{\sigma, \lambda} \) is balanced. □

**Theorem 2:** Let Assumptions 1, 2, and 5 hold, and assume that \( \tilde{F}(x, \sigma) \) is \( \beta \)-Lipschitz continuous. Then, for any small enough \( \alpha > 0 \), there is a small enough \( \beta > 0 \) such that the sequence \( (x^k, \sigma^k, z^k, \lambda^k)_{k \in \mathbb{N}} \) generated by (31) converges to a point \( (x^*, C^\sigma(x^*)), z^*, \lambda^* \) where \( (x^*, \lambda^*) \) satisfies the KKT conditions in (17); hence, \( x^* \) is the v-GNE of the game in (15) (and the estimates of aggregation and dual variables are at consensus). □

![Fig. 2. Unicast rate allocation game. The main complication is that some users send data over the same link but cannot communicate: for instance, link (4,5)—labeled \( p \)—is used by users 3 and 5 (i.e., \( N_1(p) = \{3, 5\} \), which are not communication neighbors.)](image-url)
Let us relabel the “active” edges $\bigcup_{i \in I} \{l \in L_i\}$ (i.e., the edges of $G^C$ hosting at least one path) as $\{1, 2, \ldots, P\} =: \mathcal{P}$; further, let us define for all $p \in \mathcal{P}$ and $i \in I$

$$A_{p,i} = B_{p,i} = \begin{cases} 1, & \text{if } p \in L_i \text{ (user } i \text{ uses link } p) \\ 0, & \text{otherwise} \end{cases}$$

the so-called routing matrix [11]. With these definitions, the problem retrieves an aggregative game of the type described in Section III-B, with $M = Q = \mathcal{P}$. In particular, only the aggregative variables and constraints relative to the links in $L_i$ directly affect agent $i$, i.e., $N^{1,\sigma}(i) = N^{1,\lambda}(i) = L_i$.

A similar setup was considered in [10] (as a nonaggregative game, which is inefficient) and [24]; however, both works cannot deal with the capacity coupling constraints and further assume that the graphs $\{G^{C_i}\}_{i \in (p)} \subseteq \mathcal{P}$ are connected, which is, for example, not the case in Fig. 2 (see also Example 3 in Appendix A).

Hence, the methods in [10] and [24] cannot be applied.

To seek a v-GNE, we implement algorithm in (31). In particular, we fix $W^D, \sigma = W^D, \lambda := W^D_p$ for all $p \in \mathcal{P}$. We are interested in comparing the performance of the algorithm for two different choices of the design graphs.

1) Standard: $W_p^D$s are chosen as in (C0), i.e., each agent estimates the whole dual variable and aggregative value: the problem sparsity is ignored.

2) Customized: $W_p^D$s are chosen to minimize the per-iteration communication cost, by solving Design Problem 1 with:

“(iii) For all $p \in \mathcal{P}$, $G_p^D$ is connected and has minimal number of edges.”

We note that, for the customized algorithm, the design of each graph $G_p^D$ corresponds to solving an ST problem as in Fig. 2. This is achieved in a distributed manner, and it is further only done once even if the link capacities change after some time (as long as the routing is fixed).

Choose $U_i(x_i) = 10 \times \log(x_i + 1)$, $c_i(x_i, \sigma_i) = \psi_i \lVert x_i - x^* \rVert$ with $\psi_i$ sampled randomly uniformly in the interval $[0, 1]$ [Assumption 2 is satisfied in the feasible set, invariant for (31)], Metropolis–Hastings weights for every graph, $\alpha = 0.1$, and $\beta = 10^{-3}$. The results are illustrated in Fig. 3. In our first experiment, we consider the scenario in Fig. 2, with $I = P = 12$. The customized algorithm converges to the unique v-GNE $x^*$ over ten times faster than the standard version; the communication burden at each step is also reduced, as the mean size of the (aggregative/dual) estimates kept and transmitted by each agent is 2.6, instead of $P = 12$. Since the latter quantity grows

The setup also fits more general cases where from a node $i$, multiple distinct flows $x_{i,1}, \ldots, x_{i,n_x}$ are sent over distinct paths $L_{i,1}, \ldots, L_{i,n_x}$, either by a single user (adjusting the cost function) or by multiple competitive users (considering an augmented communication graph $G^C' \supseteq G^C$).

1) For $\{S_i\}_{i \in I}$ we use the weights $w_i$ from $\lambda_i^{\prime}$.

2) In the experiments, the choice of $\sigma$, $\lambda$, and $\Gamma$ was set as $\sigma = \lambda = 1$, and $\Gamma = 0.1$.

3) As $\lambda$ is the largest eigenvalue of the dual matrix, $\lambda^2 I$ appears in the optimal choice of $\sigma$.

4) The setup also fits more general cases where from a node $i$, multiple distinct flows $x_{i,1}, \ldots, x_{i,n_x}$ are sent over distinct paths $L_{i,1}, \ldots, L_{i,n_x}$, either by a single user (adjusting the cost function) or by multiple competitive users (considering an augmented communication graph $G^C' \supseteq G^C$).

V. CONCLUSION

In this article, we presented END, a graph-theoretic language of consensus in distributed iterations. Our framework allows for unprecedented flexibility in the assignment and exchange of estimates among the agents. In fact, END algorithms can be tailored to exploit the sparsity of specific problem instances, improving scalability and reducing communication and memory bottlenecks, without requiring case-by-case convergence analysis.

We have exploited END for GNE seeking under partial-decision information, improving on known algorithms in terms of both theoretical guarantees and numerical efficiency. Yet, END can be applied to virtually any distributed decision problem, e.g., common fixed point computation, consensus optimization, multicluster games, and aggregative optimization [37]. Furthermore, we focused on fixed communication graphs, but END can also be used in case of time-varying networks [25].
Future work should focus on computationally efficient and distributed methods to perform the allocation of the estimates; in particular, it would be highly valuable to dynamically assign the estimates online, thus avoiding the need for any a priori design.

APPENDIX A
EXAMPLES OF THE END DESIGN PHASE

In this section, we present some examples of Design Problem 1 and discuss choices for the design and estimate graphs.

A. Minimal Memory Allocation: We consider the problem of minimizing, for each \( P \), the number of copies of \( y_p \), provided that the conditions in Design Problem 1(i) and (ii) are satisfied and the graphs \( \{ G^D_p \}_{p \in P} \) enjoy some connectivity properties. In particular, consider Design Problem 1 with (iii) for each \( p \in P \), \( G^D_p \) is rooted at \( r_p \in I \) (respectively, \( G^D_p \) is strongly connected); the number \( |N^E(p)| \) of nodes in \( G^D_p \) is minimal (provided that all the other specifications are satisfied).

If the problem is feasible, then by definition a solution is given by choosing each \( G^D_p \) as a solution of an unweighted directed ST problem (of a strongly connected subgraph problem) \(^{38}\) (similarly to Fig. 1). A sufficient condition for the existence of a solution is that \( G^C \) is strongly connected.

Note that the solution is generally not unique, especially because the specification is only given in terms of nodes. In fact, given any optimal choice \( G^* \) for \( G^D_p \), any graph \( G'' \) such that \( G'' \subseteq G'' \subseteq G^C \) is also a solution for Design Problem 1. In simple terms, we can add edges to \( G'' \), an extra degree of freedom that can be employed to improve connectivity or robustness to link failure (possibly at the cost of extra communication). One can also impose a different connectedness/efficiency specification on each graph \( G^D_p \) (see Fig. 1).

B. Fair Allocation and Bandwidth Constraints: Instead of minimizing the overall dimension of \( y \), it can be convenient to promote allocations where the memory occupation (or the communication requirements) are partitioned equally among the agents: for example, assuming that undirected connected graphs are required, this could be achieved by sequentially designing the graphs \( G^D_p \)'s as solutions of an ST problem (see Section 1-A), but sequentially penalizing unbalanced allocations by opportune choosing the edge weights for each \( p \). Bandwidth constraints can be addressed similarly, to avoid overloading some channels of the communication network \( G^C \).

C. Designing the Communication Graph: In this article, we consider the graph \( G^C \) as given, which is natural for ad hoc networks or when relying on existing infrastructures. Yet, other works \(^{10},^{13},^{24}\) assume that the communication network can be freely designed. In the END framework, this case is addressed by formally assuming that \( G^C \) is complete; then, the graphs \( \{ G^D_p \}_{p \in P} \) can be chosen to fulfill some specifications (e.g., minimize the number of active edges in \( G^C \)—which determines the physical channels/edges actually needed).

D. Straightforward Design: In some cases, the problem structure immediately suggests an optimal estimate allocation.

Example 2 (Partitioned optimization): Motivated by distributed estimation and resource allocation applications, the works \(^{18},^{39},^{40}\) solve optimization problems of the form

\[
\min_{y \in \mathbb{R}^{n_y}, i \in I} \sum_{i \in I} f_i(y_i, (y_j)_{j \in N^C(i)})
\]

where the cost \( f_i \) of agent \( i \) depends on its local action \( y_i \) and on the actions of its neighbors over the undirected communication network \( G^C \): this is a special case of Example 1, with \( P = I \), \( N^C(i) = N^C(i) \cup \{i\} \). Consider Design Problem 1, with (iii) \( \forall i \in I \), \( G^C_i \) is connected; \( G^C \) has minimum number of nodes (provided that all other specifications are met)." A solution is to fix each \( G^D_p \) as the undirected star graph centered in \( i \) with vertices \( V^D_p = N^C(i) \); then, agent \( i \) keeps all and only proxies of the actions that affect its cost. In fact, this is the solution employed in \(^{18},^{39},^{40}\).

Example 3 (\( G^E = G^C \)): With the goal of minimizing the overall memory allocation, consider the choice

\[
G^D_p = G^C|_{N^1(p)} \quad \forall p \in P.
\]

In this case, \( G^E = G^C \), i.e., each agent only estimates the minimum number of variables needed for local computation. Yet, this is only a viable option if the resulting graphs \( \{ G^D_p \}_{p \in I} \) ensure the desired connectedness properties in Design Problem 1(iii), which usually not verified (see \( G^D_1 \) in Fig. 1), but holds in some particular cases (e.g., Example 2, \( G^D_p \) in Fig. 1; see also \(^{10}, \text{Asm. } 5,^{13}, \text{Asm. } 6\) for sufficient conditions in the context of NE seeking).

Example 4: Another example where \( (34) \) is a viable option was studied in \(^{13},^{10}\) for NE problems arising in social networks, where the cost of agent \( i \) is

\[
\hat{f}_i(y_i, (y_j)_{j \in N^C(i)}) = \langle y_i, \alpha \rangle_{N^C(i)} + \langle y_i, \alpha \rangle_{N^C(i)}
\]

for some \( \alpha \). In this case, \( G^E = G^C \), i.e., each agent only estimates the minimum number of variables needed for local computation. Yet, this is only a viable option if the resulting graphs \( \{ G^D_p \}_{p \in I} \) ensure the desired connectedness properties in Design Problem 1(iii), which usually not verified (see \( G^D_1 \) in Fig. 1), but holds in some particular cases (e.g., Example 2, \( G^D_p \) in Fig. 1; see also \(^{10}, \text{Asm. } 5,^{13}, \text{Asm. } 6\) for sufficient conditions in the context of NE seeking).

APPENDIX B
PROOFS

A. Proof of Theorem 1

We study convergence of \((20)\) in the space weighted by \( \Xi > 0 \). Let \( y^* = \mathcal{C}(x^*) \), where \( x^* \) is the NE of \((15)\). Our proof is based on the following lemma.

Lemma 3: Let \( \mathcal{F}(y) := W^D y - \alpha R^T \mathcal{F}(W^D y) \). Then, for any \( y \in \mathbb{R}^{n_y} \), it holds that

\[
||\mathcal{F}(y) - \mathcal{F}(y^*)||_{\Xi} \leq \sqrt{\rho_{\alpha}}||y - y^*||_{\Xi}.\]

Proof: Let \( y = y_i + y_j \), where \( y_i := \text{diag}((1, q_i^{D/T}) \otimes I) y_j \in \mathcal{C} \), and thus, \( y_i \in \mathcal{C}(y_i) \) for some \( y_i \in \mathbb{R}^{n_y} \). Let \( \hat{y} := W^D y = y_i + y_j \), where \( \hat{y}_i := W^D y_i = W^D \text{diag} ((1 - 1, q_i^{D/T}) \otimes I) y_j \) and we used that \( W^D y_i = \mathcal{C}(y_i) \) (by row stochasticity). By Assumption 4, we have \( q_i^{D/T} 1 \in \mathbb{N}, Q_i W^D (1 - 1, q_i^{D/T}) = 0 \) and hence \( \langle y_i, \hat{y} \rangle = 0 \). Therefore

\[
||\hat{y} - \alpha R^T \mathcal{F}(\hat{y}) - (y^* - \alpha R^T \mathcal{F}(y^*))||_{\Xi}^2
\]

\[
= ||y_i - y^*||_{\Xi}^2 + ||y_j||_{\Xi}^2
\]

\[
+ \alpha^2 ||R^T (F(\hat{y}) - F(y_i) + F(y_j) - F(y^*))||_{\Xi}^2
\]
where the last inequality follows by the Cauchy–Schwarz inequality and using that $\| R^T v \|_2 \leq \max_{i\in I} \{ |Q_i|, i, i \} \| v \|_2$ for all $v \in \mathbb{R}^{n_x}$, because $R \in \mathbb{R}^{n_x}$ is a diagonal matrix with positive entries. Thus, $F$ is $\theta$-Lipschitz continuous if $F$ is (32, Lem. 1), and, finally, that $\| F(y_i) − F(y^*) \|_2^2 = \| F(y_i) − F(x^*) \|_2^2 \leq \theta^2 \| γ_1 \|_2^2 = \theta^2 \| γ \|_2^2 = \| \gamma \|_2^2$. Similarly, $\| \gamma \|_2^2\). Now, using the normal cone $\mathcal{C} (y_1) = \{ y \} \setminus \mathcal{C} (y_1)$, we have $\| y_1 \|_2 \leq \| y_1 \|_2$. Finally, we obtain $\| y \|_2 \leq \| y \|_2$. Together with (35), this yields

$$\| F(y) - F(y^*) \|_2 \leq \left( \begin{array}{c} \| y - y^* \|_2^2 \| y - y^* \|_2 \\ \| y - y^* \|_2 \| y - y^* \|_2 \end{array} \right)^\top \Lambda \left( \begin{array}{c} \| y - y^* \|_2 \\ \| y - y^* \|_2 \end{array} \right) \leq \lambda_{\max}(M_{\omega}) \left( \begin{array}{c} \| y - y^* \|_2 \\ \| y - y^* \|_2 \end{array} \right) \| y^* - y^t \|_2.$$

To conclude the proof of Theorem 1, we note that (20) is equivalently written as $y^{k+1} = \text{proj}_{\Omega}^{\mathcal{D}}(W^D y^k - \alpha R^T F(W^D y^k))$, where $\text{proj}_{\Omega}^{\mathcal{D}}$ is the projection in $\mathcal{H}$, i.e., $\text{proj}_{\Omega}^{\mathcal{D}}(x) = \arg \min_{y \in \Omega} \{ \| x - y \| \}$. In fact, $\text{proj}_{\Omega} = \text{proj}_{\Omega}$ blockwise under either Assumption 4(i) due to block diagonality of $Q_i$ and the rectangular structure of $Q_i$ in Section II-D. Moreover, $y^k$ is a fixed point for (20). By nonexpansiveness of the projection operator [34, Prop. 12.28], we can finally write

$$\| y^{k+1} - y^* \|_2 = \| \text{proj}_{\mathcal{D}}(F(y)) - \text{proj}_{\mathcal{D}}(F(y^*)) \|_2 \leq \| F(y) - F(y^*) \|_2,$$

and the conclusion follows by Lemma 3.

B. Proof of Theorem 2

We can rewrite (31) with the change of variable $s := \sigma - B x - b$ by replacing $\sigma^k$ with $s^k + B x^k + b$ in (31a) and (31b) with

$$s^{k+1} = s^k - \beta L^D \sigma (s^k + B x^k + b), \quad s^0 = 0. \tag{36}$$

Let us define $\omega := \text{col}(x, s, z, \lambda)$.

$$\mathcal{A}(\omega) := \left[ \begin{array}{c} \alpha F(x, \sigma) + B^T L^D \sigma \\ L^D \sigma \\ 0 \\ b \end{array} \right] + \left[ \begin{array}{c} A^\top \lambda \\ 0 \\ -L^D \lambda \\ \beta^{-1} \end{array} \right] \mathcal{A}_{\omega} + \left[ \begin{array}{c} 0 \\ 0 \\ -A^\top \mathcal{A}_{\omega} \end{array} \right], \quad \Phi := \left[ \begin{array}{c} \beta^{-1} I \\ 0 \\ 0 \\ 0 \\ \beta^{-1} I \\ 0 \\ 0 \end{array} \right]. \tag{37}$$

where $\sigma$ here is just a shorthand notation for $\sigma = x + B x + b$. We assume that $0 < \beta < 1/\| A \|_2, \| L^D \|_2$, so that $\Phi > 0$; denote $\delta := \lambda_{\min}(\Phi)$. The proof is based on the following auxiliary results. We recall the definition of $\Pi_i$ (the projection matrix into the consensus subspace) and $\mathcal{C}(\cdot)$ [see (11)] in Section II-D, and that with the superscript $\sigma$ or $\lambda$, we distinguish the spaces of different dimension related to the variables $\sigma$ and $\lambda$.

Lemma 4 (Invariance): For all $\mathbb{K} \in \mathbb{N}$, $\Pi^\omega s^k = 0$.

Proof: Via induction and by (36), since $\Pi^\omega L^D \sigma = 0$.

Lemma 5 (Algorithm derivation): The iteration in (31), with (31b) replaced by (36), can be written as

$$\mathcal{A}_2 (w^{k+1}) + \mathcal{A}_3 (w^{k+1}) + \Phi (\omega^{k+1} - \omega^k) \cong -\mathcal{A}_1 (w^k) \tag{39}$$

with $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ as in (37).

Proof: The iteration in (31a), (31c), (31d), and (36) is retrieving expanding the terms in (39) and canceling identical terms, by noting that $\mathcal{A}_3$ is the normal cone of the set $\mathcal{S} = \Omega \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$, and hence, $w^{k+1} + \mathcal{A}_3 (w^{k+1}) \cong \omega^k$ is equivalent to $w^{k+1} = \text{proj}_{\mathcal{D}} (\omega^k)$ for any $\omega^k$.

Lemma 6 (Fixed points): The fixed points of the iteration in (39) coincide with $\mathcal{Z}(\mathcal{A})$. The set $\mathcal{Z}(\mathcal{A}) \cap \Sigma = \{ \omega \mid \Pi^\omega s = 0 \}$ is nonempty. Moreover, for any $\omega^* = (x^*, s^*, z^*, \lambda^*) \in \mathcal{Z}(\mathcal{A}) \cap \Sigma$, we have that $\sigma^* := s^* + B x^* + b = C^\sigma (x^*)$, $\lambda^* = C^\lambda (\lambda^*)$, where $C^\sigma = C^\sigma (x^*)$ solves (12) hence, $x^*$ is the ν-GNE of the game in (15).

Proof: By the definition of inverse operator, (39) is equivalent to $\omega^k + \mathcal{A}_3 (w^{k+1}) = \Pi^\omega s^k - \mathcal{A}_1 (w^{k+1}) \cong \Pi^\omega s^k - \mathcal{A}_1 (w^k)$, i.e., the FB algorithm [34, Sec. 26.5] applied to the operator $\Phi^{-1} \mathcal{A}_1$; its fixed points are the zeros of $\Phi^{-1} \mathcal{A}_1$ [34, Prop. 26.1(iv)], and clearly, $\mathcal{Z}(\Phi^{-1} \mathcal{A}_1) = \mathcal{Z}(\mathcal{A})$. Note that $\sigma^* := \Pi^\omega s^* + B x^* + b = C^\sigma (x^*)$, where the first equality follows by the second row in (37) and Assumption 5 (and Lemma 1), and the second by definition of $B$, $b$, and $\Pi^\omega$. By the third row in (37), $\lambda^* \in C^\lambda$; in turn, the first row retrieves the first KKT condition in (17a). The second condition in (17b) is obtained by left-multiplying the last row in (37) by $\Pi^\omega$. Conversely, it can be shown as in [7, Lem. 10] that, for any $(x^*, \lambda^*)$ solves (17) (at least one such pair exists by Assumption 2), there exists $z^* \in \mathbb{R}^{n_x}$ such that $(x^*, s^* := C^\sigma (x^*) + B x^* - b, z^*, C^\lambda (\lambda^*)) \in \mathcal{Z}(\mathcal{A})$; therefore, $\mathcal{Z}(\mathcal{A}) \cap \Sigma \neq \emptyset$, since $\Pi^\omega s^* = 0$.

Authorized licensed use limited to: UNIVERSITY OF WESTERN ONTARIO. Downloaded on June 09,2023 at 22:10 (UTC) from IEEE Xplore. Restrictions apply.
Lemma 7 (Monotonicity properties): The operator \( A_2 + A_3 \) is maximally monotone by [34, Def. 20.20]. Let \( \lambda := \lambda_2(D_{\sigma} + D_{\sigma'})/2 \); then, for any \( 0 < \alpha < 4\lambda_2^2 \), the operator \( A_1 \) is \( \eta \)-restricted cocoercive, for some \( \eta > 0 \) depending on \( \alpha \): for all \( \omega \in \Sigma \) and all \( \omega^* \in \text{zer}(A) \cap \Sigma \), \( \langle A_1(\omega) - A_1(\omega^*), \omega - \omega^* \rangle \geq \eta \| A_1(\omega) - A_1(\omega^*) \|^2 \).

Proof: \( A_2 \) is maximally monotone because it is a skew-symmetric linear operator by [34, Ex. 20.35], the normal cone \( A_3 \) is maximally monotone by [34, Ex. 20.26]; then, since \( A_2 \) has full domain, \( A_2 + A_3 \) is maximally monotone by [34, Th. 25.2]. For the second statement, for all \( \omega \in \Sigma \), we have \( \langle \omega - \omega^*, A_1(\omega) - A_1(\omega^*) \rangle = \alpha \langle x - x^*, F(x, \sigma) - F(x^*, \sigma^*) \rangle + \langle \sigma - \sigma^*, L_{D_{\sigma}}(\sigma - \sigma^*) \rangle \), and the result follows identical to [36, Lemma 5] by using Lemma 2, that is \( F \) is \( \theta \)-Lipschitz, that \( F(x, \sigma) = F(x) \), and \( F(x) \) is \( \mu \)-strongly monotone (we refer to [36] for the expression of \( \eta \)).

Lemma 5 recasts (31) as a preconditioned FB algorithm [6], applied to the operators \( A_1 \) and \( A_2 + A_3 \), with preconditioning matrix \( \Phi > 0 \). In turn, Lemma 7 ensures the conditions on the operators \( A_1 \) and \( A_2 + A_3 \) that guarantee the convergence of the preconditioned FB method to a fixed point in \( \text{zer}(A) \cap \Sigma \), provided that the step size \( \beta \) is chosen such that \( 0 < \beta < 2\eta_0 \beta \) (this can be proven by showing the decrease of the Lyapunov function \( |\omega - \omega^*|_\Phi \), with analysis restricted to the invariant subspace \( \Sigma \), as in [6, Th. 2] and [36, Th. 1]); the argument is standard and omitted here due to space limitations. Finally, Lemmas 4 and 6 characterize such a fixed point as per statement.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2010.
[2] F. Facchinei and K. Kanzow, “Generalized Nash equilibrium problems,” 4OR, vol. 5, pp. 46–62, 2007.
[3] H.-B. Dür, M. S. T Stanekov, and K. H. Johanson, “Distributed positioning of autonomous mobile sensors with application to coverage control,” in Proc. Amer. Control Conf., 2011, pp. 4822–4827.
[4] M. Ye and G. Hu, “Distributed Nash equilibrium seeking by a consensus-based approach,” IEEE Trans. Autom. Control, vol. 62, no. 9, pp. 4811–4818, Sep. 2017.
[5] A. R. Romano and L. Pavel, “An inexact-penalty method for GNE seeking in games with dynamic agents,” IEEE Trans. Control Netw. Syst., vol. 10, no. 4, pp. 2072–2084, Dec. 2023.
[6] L. Pavel, “Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach,” IEEE Trans. Autom. Control, vol. 65, no. 4, pp. 1584–1597, Apr. 2020.
[7] M. Bianchi, G. Belgioioso, and S. Grammatico, “Fast generalized Nash equilibrium seeking under partial-decision information,” Automatica, vol. 136, 2022, Art. 110080.
[8] D. Gadjo and L. Pavel, “An algorithm for resilient Nash equilibrium seeking in the partial information setting,” IEEE Trans. Control Netw. Syst., vol. 10, no. 4, pp. 1645–1655, Dec. 2023.
[9] Y. Zhu, W. Yu, W. Ren, G. Wen, and J. Gu, “Generalized Nash equilibrium seeking via continuous-time coordination dynamics over digraphs,” IEEE Trans. Control Netw. Syst., vol. 8, no. 2, pp. 1023–1033, Jun. 2021.
[10] F. Salehisadaghi and L. Pavel, “Distributed Nash equilibrium seeking in networked graphical games,” Automatica, vol. 87, pp. 77–24, 2018.
[11] T. Alpcan and T. Basar, “A game-theoretic framework for congestion control in general topology networks,” in Proc. IEEE 41st Conf. Decis. Control, 2002, vol. 2, pp. 1218–1224.
[12] L. Pavel, Game Theory for Control of Optical Networks, Cambridge, MA, USA: Birkhäuser, 2012.
M. Todescato, N. Bof, G. Cavraro, R. Carli, and L. Schenato, “Partition-based multi-agent optimization in the presence of lossy and asynchronous communication,” *Automatica*, vol. 111, 2020, Art. no. 108648.

Mattia Bianchi (Member, IEEE) received the bachelor’s degree in information engineering and the master’s degree in control systems engineering from the University of L’Aquila, L’Aquila, Italy, in 2016 and 2018, respectively, and the Ph.D. degree in automatic control from the Delft University of Technology, Delft, The Netherlands, in 2023.

He is currently a Postdoctoral Researcher with the Automatic Control Laboratory, ETH Zürich, Zürich, Switzerland. In 2018, he visited the Control Systems Technology Group, Eindhoven University of Technology, Eindhoven, The Netherlands. In 2021 and 2022, he visited the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA, USA. His research interests include game theory and operator theory to solve decision and control problems for complex systems of systems, in the presence of network coupling and uncertainty.

Dr. Bianchi was the recipient of the Roberto Tempo Best Paper Award at the IEEE Conference on Decision and Control in 2021.

Sergio Grammatico (Senior Member, IEEE) was born in 1987. He received the bachelor’s degree in computer engineering and the master’s degree in automatic control engineering from the University of Pisa, Pisa, Italy, in 2008 and 2009, respectively, the master’s degree in engineering science from the Sant’Anna School of Advanced Studies, Italian Superior Graduate School (Grande École) for Applied Sciences, Pisa, in 2011, and the Ph.D. degree in automatic control from the University of Pisa in 2013.

He is currently an Associate Professor with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. From 2013 to 2015, he was a Postdoctoral Researcher with the Automatic Control Laboratory, ETH Zürich, Zürich, Switzerland. From 2015 to 2018, he was an Assistant Professor with the Department of Electrical Engineering and Control Systems, Eindhoven University of Technology, Eindhoven, The Netherlands.

Prof. Grammatico was the recipient of a 2005 F. Severi B.Sc. Scholarship from the Italian High-Mathematics National Institute, the 2008 M.Sc. Fellowship from the Sant’Anna School of Advanced Studies, the 2021 Roberto Tempo Best Paper Award at the IEEE Conference on Decision and Control. He has been an Associate Editor for *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* since 2018 and for *Automatica* since 2020.