The END
Estimation Network Design for Games under Partial-decision Information

Bianchi, Mattia; Grammatico, Sergio

DOI
10.1109/TCNS.2024.3393668

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Control of Network Systems

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

LARGE-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 $i \in \mathcal{I} := \{1, 2, \dots, N\}$ 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

$$\underset{x_i}{\text{minimize}} \;\; f_i(x_i, x_{-i}) \qquad \forall i \in \mathcal{I}. \tag{1}$$

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

$$x_i^{k+1} = x_i^k - \nabla_{x_i} f_i(x_i^k, x_{-i}^k), \qquad i \in \mathcal{I}, k \in \mathbb{N} \tag{2}$$

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 $\mathcal{N}^{\mathrm{I}}(i) \subset \mathcal{I}$, 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

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

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.

Manuscript received 4 October 2023; revised 7 January 2024; accepted 25 January 2024. Date of publication 25 April 2024; date of current version 31 January 2025. This work was supported in part by the Netherlands Organization for Scientific Research through project OMEGA under Grant 613.001.702, in part by the European Research Council through project COSMOS under Grant 802348, and in part by ETH Zürich funds. Recommended by Associate Editor Y. Wang. *(Corresponding author: Mattia Bianchi.)*

Mattia Bianchi is with the Automatic Control Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mail: mbianch@ethz.ch).

Sergio Grammatico is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.grammatico@tudelft.nl).

Digital Object Identifier 10.1109/TCNS.2024.3393668

Such scenarios are not limited to games. In fact, many big data *optimization* problems are also *partially separable* [14] (i.e., the local cost of each agent only depends on a limited portion of the optimization vector)—but this fact is rarely exploited in *distributed* algorithms. The issue was solely considered, via dual methods, by Mota et al. [15] and later by Alghunaim et al. [16], [17]; in their approach, each component of the optimization variable is only estimated by a suitable subset of the agents. As a drawback, the dual reformulation is effective over undirected communication networks and viable for optimization problems only.

Motivations: Our work is motivated by the observation that, in multiagent applications, the coupling among the agents often exhibits some sparsity. This sparsity *could* and *should* be exploited to enhance the efficiency of distributed algorithms, e.g., by reducing the number of redundant estimates in the network. While ad hoc schemes were developed for special sparsity patterns (e.g., when each agent is only coupled with its communication neighbors [18] or neighbors' neighbors [10]), what is missing is a systematic methodology to account for general, problem-specific, sparsity structures—but without resorting to a case-by-case convergence analysis.

Contributions: 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 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. $\mathbf{0}_q \in \mathbb{R}^q$ ($\mathbf{1}_q \in \mathbb{R}^q$) is a vector with all elements equal to 0 (1), $\mathbf{0}_{q\times p} \in \mathbb{R}^{q\times p}$ is a matrix with all elements equal to 0, and $\mathrm{I}_q \in \mathbb{R}^{q\times q}$ is an identity matrix; the subscripts may be omitted when there is no ambiguity. e_i denotes a vector of appropriate dimension with the ith element equal to 1 and all other elements equal to 0. For a matrix $\mathrm{A} \in \mathbb{R}^{p\times q}$, $[\mathrm{A}]_{i,j}$ is the element on row i and column j, $\mathrm{null}(\mathrm{A}) := \{x \in \mathbb{R}^q \mid \mathrm{A}x = \mathbf{0}_n\}$ and $\mathrm{range}(\mathrm{A}) := \{v \in \mathbb{R}^p \mid v = \mathrm{A}x, x \in \mathbb{R}^q\}$, and $\|\mathrm{A}\|_\infty$ is the maximum of the absolute row sums of A. If $\mathrm{A} = \mathrm{A}^\top \in \mathbb{R}^{q\times q}$, $\lambda_{\min}(\mathrm{A}) =: \lambda_1(\mathrm{A}) \leq \cdots \leq \lambda_q(\mathrm{A}) =: \lambda_{\max}(\mathrm{A})$ denotes its eigenvalues. $\mathrm{diag}(\mathrm{A}_1, \ldots, \mathrm{A}_N)$ is the block diagonal matrix with $\mathrm{A}_1, \ldots, \mathrm{A}_N$ on its diagonal. Given N vectors $x_1, \ldots, x_N$, $\mathrm{col}(x_1, \ldots, x_N) := [x_1^\top \ldots x_N^\top]^\top$. $\otimes$ is the Kronecker product.

2) Graph Theory: A (directed) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a nonempty set of vertices (or nodes) $\mathcal{V} = \{1, 2, \ldots, V\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote by $\mathcal{N}(v) := \{u \mid (u, v) \in \mathcal{E}\}$ and $\overline{\mathcal{N}}(v) := \{u \mid (v, u) \in \mathcal{E}\}$ the set of in-neighbors (or simply neighbors) and out-neighbors of vertex $v \in \mathcal{V}$, respectively. A path from $v_1 \in \mathcal{V}$ to $v_N \in \mathcal{V}$ of length T is a sequence of vertices $(v_1, v_2, \ldots, v_T)$ such that $(v_t, v_{t+1}) \in \mathcal{E}$ for all $t = 1, \ldots, T-1$. $\mathcal{G}$ is rooted at $v \in \mathcal{V}$ if there exists a path from v to each $u \in \mathcal{V}\backslash\{v\}$. $\mathcal{G}$ is strongly connected if there exists a path from u to v, for all $u, v \in \mathcal{V}$; in case $\mathcal{G}$ is undirected, namely, if $(u,v) \in \mathcal{E}$ whenever $(v,u) \in \mathcal{E}$, we simply say that $\mathcal{G}$ is connected. The restriction of the graph $\mathcal{G}$ to a set of vertices $\mathcal{V}^{\mathrm{A}} \subseteq \mathcal{V}$ is defined as $\mathcal{G}|_{\mathcal{V}^{\mathrm{A}}} := (\mathcal{V}^{\mathrm{A}}, \mathcal{E} \cap (\mathcal{V}^{\mathrm{A}} \times \mathcal{V}^{\mathrm{A}}))$. We also write $\mathcal{G} = (\mathcal{V}^{\mathrm{A}}, \mathcal{V}^{\mathrm{B}}, \mathcal{E})$ to highlight that $\mathcal{G}$ is bipartite, namely, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathcal{V}^{\mathrm{A}} \cup \mathcal{V}^{\mathrm{B}}$ and $\mathcal{E} \subseteq \mathcal{V}^{\mathrm{A}} \times \mathcal{V}^{\mathrm{B}}$. We may associate to $\mathcal{G}$ a weight matrix $\mathrm{W} \in \mathbb{R}^{V \times V}$ compliant with $\mathcal{G}$, namely, $w_{u,v} := [\mathrm{W}]_{u,v} > 0$ if $(v,u) \in \mathcal{E}$, $w_{u,v} = 0$ otherwise; we denote by $\mathrm{D} = \operatorname{diag}((\deg(v))_{v \in \mathcal{V}})$ and $\mathrm{L} = \mathrm{D} - \mathrm{W}$ the in-degree and Laplacian matrices, respectively, with $\deg(v) = \sum_{u \in \mathcal{V}} w_{u,v}$. $\mathcal{G}$ is unweighted if $w_{u,v} = 1$ whenever $(v,u) \in \mathcal{E}$. Given two graphs $\mathcal{G}^{\mathrm{A}} = (\mathcal{V}^{\mathrm{A}}, \mathcal{E}^{\mathrm{A}})$ and $\mathcal{G}^{\mathrm{B}} = (\mathcal{V}^{\mathrm{B}}, \mathcal{E}^{\mathrm{B}})$, we write $\mathcal{G}^{\mathrm{A}} \subseteq \mathcal{G}^{\mathrm{B}}$ if $\mathcal{G}^{\mathrm{A}}$ is a subgraph of $\mathcal{G}^{\mathrm{B}}$, i.e., if $\mathcal{V}^{\mathrm{A}} \subseteq \mathcal{V}^{\mathrm{B}}$ and $\mathcal{E}^{\mathrm{A}} \subseteq \mathcal{E}^{\mathrm{B}}$; we let $\mathcal{G}^{\mathrm{A}} \bigcup \mathcal{G}^{\mathrm{B}} := (\mathcal{V}^{\mathrm{A}} \cup \mathcal{V}^{\mathrm{B}}, \mathcal{E}^{\mathrm{A}} \cup \mathcal{E}^{\mathrm{B}})$. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, compliant weights W, and terminals $\mathcal{T} \subseteq \mathcal{V}$, we define the following problems.

1) *Steiner tree (ST) problem, ST$(\mathcal{G}, \mathcal{T}, \mathrm{W})$:* Find an undirected connected subgraph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*) \subseteq \mathcal{G}$, $\mathcal{T} \subseteq \mathcal{V}^*$, with minimum cost (i.e., minimizing $\sum_{(v,u) \in \mathcal{E}^*} w_{u,v}$).
2) *Unweighted Steiner tree (UST) problem, UST$(\mathcal{G}, \mathcal{T})$:* Find an undirected connected subgraph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*) \subseteq \mathcal{G}$, $\mathcal{T} \subseteq \mathcal{V}^*$, with minimum number of nodes (equivalently, of edges).

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

4) Operator Theory: A set-valued operator $\mathsf{F} : \mathbb{R}^q \rightrightarrows \mathbb{R}^q$ is characterized by its graph $\operatorname{gra}(\mathsf{F}) := \{(x,u) \mid u \in \mathsf{F}(x)\}$. $\operatorname{dom}(\mathsf{F}) := \{x \in \mathbb{R}^q \mid \mathsf{F}(x) \neq \varnothing\}$, $\operatorname{fix}(\mathsf{F}) := \{x \in \mathbb{R}^q \mid x \in \mathsf{F}(x)\}$, and $\operatorname{zer}(\mathsf{F}) := \{x \in \mathbb{R}^q \mid \mathbf{0} \in \mathsf{F}(x)\}$ are the domain, set of fixed points, and set of zeros, respectively. F^{-1} denotes the inverse operator of F, defined as $\operatorname{gra}(\mathsf{F}^{-1}) = \{(u,x) \mid (x,u) \in \operatorname{gra}(\mathsf{F})\}$. F is (μ-strongly) monotone if $\langle u - v, x - y \rangle \geq 0$ ($\geq \mu \|x - y\|^2$) for all $(x,u), (y,v) \in \operatorname{gra}(\mathsf{F})$. Id is the identity operator. For a set $\Omega \subseteq \mathbb{R}^q$, $\mathsf{N}_\Omega : \Omega \rightrightarrows \mathbb{R}^q : x \mapsto \{v \in \mathbb{R}^q \mid \sup_{z \in \Omega} \langle v, z - x \rangle \leq 0\}$ is the normal cone operator of Ω. If Ω is closed and convex, then $(\mathsf{Id} + \mathsf{N}_\Omega)^{-1} = \operatorname{proj}_\Omega$ is the Euclidean projection onto Ω.

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 $\mathcal{G}^{\mathrm{C}} = (\mathcal{I}, \mathcal{E}^{\mathrm{C}})$, over which the agents can exchange information: agent i can receive information from agent j if and only if $j \in \mathcal{N}^{\mathrm{C}}(i)$;
3) a variable of interest $y \in \mathbb{R}^{n_y}$, partitioned as $y = \operatorname{col}((y_p)_{p \in \mathcal{P}})$, where $y_p \in \mathbb{R}^{n_{y_p}}$ and $\mathcal{P} := \{1, 2, \ldots, P\}$;
4) a given bipartite directed *interference* graph $\mathcal{G}^{\mathrm{I}} = (\mathcal{P}, \mathcal{I}, \mathcal{E}^{\mathrm{I}})$, $\mathcal{E}^{\mathrm{I}} \subseteq \mathcal{P} \times \mathcal{I}$, that specifies which components of y are indispensable to each agent: $p \in \mathcal{N}^{\mathrm{I}}(i)$ 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_ps and exchanges its estimates with some neighbors, as specified by:

5) a bipartite directed *estimate* graph $\mathcal{G}^{\mathrm{E}} = (\mathcal{P}, \mathcal{I}, \mathcal{E}^{\mathrm{E}})$, $\mathcal{E}^{\mathrm{E}} \subseteq \mathcal{P} \times \mathcal{I}$, that specifies which components of y are estimated by each agent: agent i keeps an estimate $\boldsymbol{y}_{i,p} \in \mathbb{R}^{n_{y_p}}$ of y_p if and only if $p \in \mathcal{N}^{\mathrm{E}}(i)$, and thus, $\overline{\mathcal{N}}^{\mathrm{E}}(p)$ is the set of agents that keeps an estimate of y_p;
6) P directed *design* graphs $\{\mathcal{G}_p^{\mathrm{D}}\}_{p \in \mathcal{P}}$, with $\mathcal{G}_p^{\mathrm{D}} = (\overline{\mathcal{N}}^{\mathrm{E}}(p), \mathcal{E}_p^{\mathrm{D}})$, to describe how the agents exchange their estimates: agent i can receive $\boldsymbol{y}_{j,p}$ from agent j if and only if $j \in \mathcal{N}_p^{\mathrm{D}}(i)$.

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

Example 1 (Partially separable costs): Consider a multiagent problem, where the agents can only exchange information over a communication network $\mathcal{G}^{\mathrm{C}}$, and each agent $i \in \mathcal{I}$ has a private cost function $f_i(x)$, depending on a global decision vector $x \in \mathbb{R}^{n_x}$, 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 $y = x$ 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 $\mathcal{G}^{\mathrm{I}}$: f_i depends on y_p if and only if $p \in \mathcal{N}^{\mathrm{I}}(i) \subseteq \mathcal{P}$. With some abuse of notation, we emphasize this fact by writing

$$f_i(y) = f_i((y_p)_{p \in \mathcal{N}^{\mathrm{I}}(i)}). \tag{4}$$

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 $\tilde{\boldsymbol{y}}_i := \operatorname{col}((\boldsymbol{y}_{i,p})_{p \in \mathcal{P}}) \in \mathbb{R}^{n_y}$ of the whole decision variable and to let the agents exchange their estimates with every neighbor over $\mathcal{G}^{\mathrm{C}}$. In END notation, we write this as

$$\mathcal{E}^{\mathrm{E}} = \mathcal{P} \times \mathcal{I}, \qquad \mathcal{G}_p^{\mathrm{D}} = \mathcal{G}^{\mathrm{C}} \ (\forall p \in \mathcal{P}) \tag{C0}$$

(we refer to (C0) as the "standard" setup, since it is the usually studied scenario). This choice of graphs $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p \in \mathcal{P}}$ ignores the possible structure in (4). Note that agent i only needs $(y_p)_{p \in \mathcal{N}^{\mathrm{I}}(i)}$ 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 $\mathcal{G}^{\mathrm{I}}$ is sparse and P is large. Are smarter choices of $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p \in \mathcal{P}}$ 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 $\overline{\mathcal{N}}^{\mathrm{I}}(p) \neq \varnothing$ for all $p \in \mathcal{P}$ (i.e., each component of y is indispensable to some agent).

(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 $\mathcal{G}^{\mathrm{C}}$ and $\mathcal{G}^{\mathrm{I}}$ shall be considered fixed a priori. In contrast, the graphs $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ 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 $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ so that it is possible to *distributedly* 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 $\mathcal{G}^{\mathrm{C}}$ and the interference graph $\mathcal{G}^{\mathrm{I}}$, design the estimate graph $\mathcal{G}^{\mathrm{E}}$ and the design graphs $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ such that:

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

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

iii) "additional requirements" on $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ are met. □

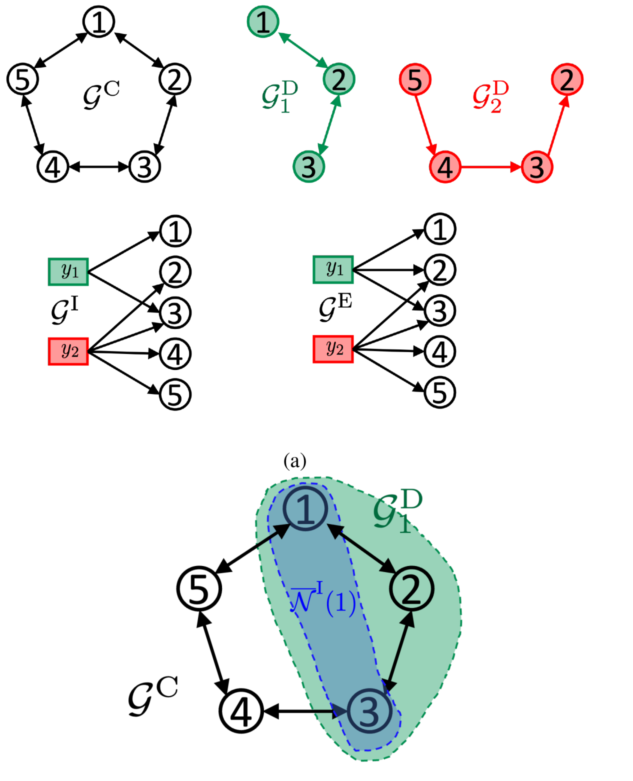
In particular, it must hold that $\mathcal{G}^{\mathrm{E}} \subseteq \mathcal{G}^{\mathrm{I}}$, namely, each agent estimates at least the components of y, which are indispensable for local computation (in fact, $\mathcal{G}^{\mathrm{I}}$ expresses the minimal information necessary for each agent). Moreover, since data transmission can only happen over the communication graph $\mathcal{G}^{\mathrm{C}}$, it must hold that $\mathcal{G}_p^{\mathrm{D}} \subseteq \mathcal{G}^{\mathrm{C}}$, 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 $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ 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 $\mathcal{G}^{\mathrm{E}}$ and $\mathcal{G}_p^{\mathrm{D}}$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 $\mathcal{G}^{\mathrm{I}}$ 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 $\mathcal{G}_p^{\mathrm{D}}$ straightforward, as shown in Appendix A-4. □


1
2
3
4
5
$\mathcal{G}^{\mathrm{C}}$
1
2
3
$\mathcal{G}_1^{\mathrm{D}}$
5
2
4
3
$\mathcal{G}_2^{\mathrm{D}}$
y_1
$\mathcal{G}^{\mathrm{I}}$
y_2
1
2
3
4
5
y_1
$\mathcal{G}^{\mathrm{E}}$
y_2
1
2
3
4
5
1
$\mathcal{G}_1^{\mathrm{D}}$
5
2
$\overline{\mathcal{N}}^{\mathrm{I}}(1)$
$\mathcal{G}^{\mathrm{C}}$
4
3

(a)

(b)

Fig. 1. (a) Simple example of END. On the left, the given communication and interference graphs, with $\mathcal{I} = \{1,2,3,4,5\}$ and $\mathcal{P} = \{1,2\}$. On the right, a possible choice for the design graphs and the corresponding estimate graphs; they solve Design Problem 1 with "(iii) $\mathcal{G}_1^{\mathrm{D}}$ is undirected and connected and with minimal number of nodes; $\mathcal{G}_2^{\mathrm{D}}$ is rooted at 5 and with minimal number of edges." (b) Let us focus on the design of $\mathcal{G}_1^{\mathrm{D}}$. The goal is to minimize the number of copies of y_1 (i.e., the number of nodes in $\mathcal{G}_1^{\mathrm{D}}$), but provided that $\mathcal{G}_1^{\mathrm{D}}$ is connected and the conditions in Design Problem 1(i) and (ii) are met (i.e., $\overline{\mathcal{N}}^{\mathrm{I}}(1) \subseteq \overline{\mathcal{N}}^{\mathrm{E}}(1)$ and $\mathcal{G}_1^{\mathrm{D}} \subseteq \mathcal{G}^{\mathrm{C}}$). Note that agent 2 has to estimate y_1 (i.e., $2 \in \overline{\mathcal{N}}^{\mathrm{E}}(1)$), even though agent 2 is not directly affected by y_1 (i.e., $2 \notin \overline{\mathcal{N}}^{\mathrm{I}}(1)$): otherwise, the information could not travel between nodes 1 and 3, which are not communication neighbors. In general, a solution to this design problem can be obtained by solving the UST problem UST($\mathcal{G}^{C}$,$\overline{\mathcal{N}}^{\mathrm{I}}(1)$) (see Section I-A), for which distributed off-the-shelf algorithms are available [26]. The problem is NP-hard in general, but can be approximated in polynomial time. Sufficient for the existence of a solution is that $\mathcal{G}^{\mathrm{C}}$ is undirected and connected.

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 $\mathcal{G}^{\mathrm{E}}$ and $\{\mathcal{G}_p^{\mathrm{D}}\}_{p\in\mathcal{P}}$ 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 $\mathcal{G}^{\mathrm{I}} \subseteq \mathcal{G}^{\mathrm{E}}$ and that $\mathcal{G}_p^{\mathrm{D}} \subseteq \mathcal{G}^{\mathrm{C}}$ 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 $\mathcal{G}_p^{\mathrm{D}}$ 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 $N_p := |\overline{\mathcal{N}}^{\mathrm{E}}(p)|$ be the overall number of copies of y_p kept by all the agents. We define

$$\boldsymbol{y}_p := \mathrm{col}((\boldsymbol{y}_{i,p})_{i\in\overline{\mathcal{N}}^{\mathrm{E}}(p)}) \in \mathbb{R}^{N_p n_{y_p}} \quad \forall p \in \mathcal{P} \tag{5}$$

$$\boldsymbol{y} := \mathrm{col}((\boldsymbol{y}_p)_{p\in\mathcal{P}}) \in \mathbb{R}^{n_{\boldsymbol{y}}} \tag{6}$$

where we recall that $\boldsymbol{y}_{i,p}$ is the estimate of the quantity y_p kept by agent i; $n_{\boldsymbol{y}} := \sum_{p\in\mathcal{P}} N_p n_{y_p}$. Note that $\boldsymbol{y}_p$ collects all the copies of y_p, kept by different agents.

We denote

$$\mathbf{W}^{\mathrm{D}} := \mathrm{diag}((\mathrm{W}_p^{\mathrm{D}} \otimes \mathrm{I}_{n_{y_p}})_{p\in\mathcal{P}}) \tag{7}$$

$$\mathbf{L}^{\mathrm{D}} := \mathrm{diag}((\mathrm{L}_p^{\mathrm{D}} \otimes \mathrm{I}_{n_{y_p}})_{p\in\mathcal{P}}) \tag{8}$$

where $\mathrm{W}_p^{\mathrm{D}}$ is the weight matrix of $\mathcal{G}_p^{\mathrm{D}}$, and $\mathrm{L}_p^{\mathrm{D}}$ is its Laplacian. For all $p \in \mathcal{P}$, let

$$\boldsymbol{\mathcal{C}}_p := \{\boldsymbol{y}_p \in \mathbb{R}^{N_p n_{y_p}} \mid \boldsymbol{y}_p = \mathbf{1}_{N_p} \otimes v, v \in \mathbb{R}^{n_{y_p}}\} \tag{9}$$

$$\boldsymbol{\mathcal{C}} := \textstyle\prod_{p\in\mathcal{P}} \boldsymbol{\mathcal{C}}_p \tag{10}$$

be the consensus space for $\boldsymbol{y}_p$, namely, the subspace where all the estimates of y_p are equal, and the overall consensus space, respectively. Given $y = \mathrm{col}((y_p)_{p\in\mathcal{P}}) \in \mathbb{R}^{n_y}$, we also define

$$\boldsymbol{\mathcal{C}}(y) := \mathrm{col}((\mathbf{1}_{N_p} \otimes y_p)_{p\in\mathcal{P}}) \in \boldsymbol{\mathcal{C}}. \tag{11}$$

We denote by $\boldsymbol{\mathcal{C}}_{\perp}$ the complementary subspace of $\boldsymbol{\mathcal{C}}$, and by $\Pi_{\parallel} := \mathrm{diag}((\mathbf{1}_{Np}\mathbf{1}_{N_p}^{\top} \otimes I_{n_{y_p}}/N_p)_{p\in\mathcal{P}})$ and $\Pi_{\perp} := \mathrm{I} - \Pi_{\parallel}$ the projection matrices onto $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{C}}_{\perp}$, respectively.

For each $p \in \mathcal{P}$, for each $i \in \overline{\mathcal{N}}^{\mathrm{E}}(p)$, we denote by

$$i_p := \textstyle\sum_{j\in\overline{\mathcal{N}}^{\mathrm{E}}(p),\, j\le i} 1 \tag{12}$$

the position of i in the ordered set of nodes $\overline{\mathcal{N}}^{\mathrm{E}}(p)$. For all $i \in \overline{\mathcal{N}}^{\mathrm{E}}(p)$, we denote by $\mathrm{R}_{i,p} \in \mathbb{R}^{n_{y_p} \times N_p n_{y_p}}$ the matrix that selects $\boldsymbol{y}_{i,p}$ from $\boldsymbol{y}_p$, i.e., $\boldsymbol{y}_{i,p} = \mathrm{R}_{i,p}\boldsymbol{y}_p$.

Sometimes, it is useful to define agentwise quantities, which we indicate with a tilde. Let

$$\tilde{\boldsymbol{y}}_i := \mathrm{col}((\boldsymbol{y}_{i,p})_{p\in\mathcal{N}^{\mathrm{E}}(i)}) \quad \forall i \in \mathcal{I} \tag{13}$$

$$\tilde{\boldsymbol{y}} := \mathrm{col}((\tilde{\boldsymbol{y}}_i)_{i\in\mathcal{I}}) \in \mathbb{R}^{n_{\boldsymbol{y}}} \tag{14}$$

where $\tilde{\boldsymbol{y}}_i$ collects all the estimates kept by agent i. Let $\mathrm{P} \in \mathbb{R}^{n_{\boldsymbol{y}} \times n_{\boldsymbol{y}}}$ be the permutation matrix such that $\mathrm{P}\boldsymbol{y} = \tilde{\boldsymbol{y}}$; the graph structures corresponding to $\tilde{\boldsymbol{y}}$ can be defined via permutations, e.g., $\tilde{\mathbf{W}}^{\mathrm{D}} := \mathrm{P}\mathbf{W}^{\mathrm{D}}\mathrm{P}^{\top}$, $\tilde{\mathbf{L}}^{\mathrm{D}} := \mathrm{P}\mathbf{L}^{\mathrm{D}}\mathrm{P}^{\top}$, and $\tilde{\boldsymbol{\mathcal{C}}} := \mathrm{P}\boldsymbol{\mathcal{C}}$.

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 $\mathcal{G}_p^{\mathrm{D}}$ is rooted at r_p. Then, $\mathrm{null}(\mathbf{L}^{\mathrm{D}}) = \boldsymbol{\mathcal{C}}$. Moreover, $\mathrm{null}(\tilde{\mathbf{L}}^{\mathrm{D}}) = \tilde{\boldsymbol{\mathcal{C}}}$. □

Lemma 2: Assume that $\mathcal{G}_p^{\mathrm{D}}$ is strongly connected and that $\mathrm{W}_p^{\mathrm{D}}$ is balanced, for all $p \in \mathcal{P}$. Then, for any $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$, $\langle \boldsymbol{y}, \mathbf{L}^{\mathrm{D}}\boldsymbol{y}\rangle \ge \frac{\bar{\lambda}}{2}\|(\mathrm{I} - \Pi_{\parallel})\boldsymbol{y}\|^2$, where $\bar{\lambda} := \min_{p\in\mathcal{P}}\{\lambda_2({\mathrm{L}_p^{\mathrm{D}}}^{\top} + \mathrm{L}_p^{\mathrm{D}})\} > 0$. □

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 $f_i(x_i, x_{-i})$, $f_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{x_{-i}}} \to \mathbb{R}$, which depends both on its local action (decision variable) $x_i \in \mathbb{R}^{n_{x_i}}$ and on the actions of the other agents $x_{-i} := \mathrm{col}((x_j)_{j\in\mathcal{I}\setminus\{i\}}) \in \mathbb{R}^{n_{x_{-i}}}$. Each agent chooses its action in a local feasible set $\Omega_i \subseteq \mathbb{R}^{n_{x_i}}$; let $x := \mathrm{col}((x_i)_{i\in\mathcal{I}}) \in \Omega$ be the overall action, with $\Omega := \prod_{i\in\mathcal{I}} \Omega_i \subseteq \mathbb{R}^{n_x}$. The agents' decisions are also coupled via shared constraints: specifically, the overall feasible set is $\mathcal{X} := \Omega \cap \{x \in \mathbb{R}^{n_x} \mid \mathrm{A}x \le a\}$, where $\mathrm{A} \in \mathbb{R}^{n_\lambda \times n_x}$, $a \in \mathbb{R}^{n_\lambda}$. We call *generalized game* the following set of interdependent optimization problems:

$$(\forall i \in \mathcal{I}) \ \underset{x_i\in\mathbb{R}^{n_{x_i}}}{\mathrm{minimize}} \ f_i(x_i, x_{-i}) \ \text{s.t.} \ (x_i, x_{-i}) \in \mathcal{X}. \tag{15}$$

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

Definition 1: A GNE is an N-tuple $x^\star = \mathrm{col}((x_i^\star)_{i\in\mathcal{I}}) \in \mathcal{X}$ such that, for all $i \in \mathcal{I}$, $f_i(x_i^\star, x_{-i}^\star) \le \inf_{x_i}\{f_i(x_i, x_{-i}^\star) \mid (x_i, x_{-i}^\star) \in \mathcal{X}\}$. □

Let us define the *pseudogradient* operator $\mathsf{F} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, as follows:

$$\mathsf{F}(x) := \mathrm{col}((\nabla_{x_i} f_i(x_i, x_{-i}))_{i\in\mathcal{I}}). \tag{16}$$

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 $i \in \mathcal{I}$, Ω_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 μ-strongly monotone and θ-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, $x^\star$ is the v-GNE of the game in (15) if and only if there exists a dual variables $\lambda^\star \in \mathbb{R}^{n_\lambda}$ satisfying the following Karush–Kuhn–Tucker (KKT) conditions [2, Th. 4.8] (we recall that $\mathrm{N}_{\mathcal{S}}$ denotes the normal cone of a set $\mathcal{S}$)

$$\mathbf{0}_{n_x} \in \mathsf{F}(x^\star) + \mathrm{N}_{\Omega}(x^\star) + \mathrm{A}^{\top}\lambda^\star \tag{17a}$$

$$\mathbf{0}_{n_\lambda} \in -(Ax^\star - a) + \mathrm{N}_{\mathbb{R}_{\ge 0}^{n_\lambda}}. \tag{17b}$$

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 $\mathcal{G}^{\mathrm{C}} = (\mathcal{I}, \mathcal{E}^{\mathrm{C}})$. 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., $\mathcal{X} = \Omega$): 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., $y = x$, $\mathcal{P} = \mathcal{I}$, and $y_i = x_i$ for all $i \in \mathcal{I}$ (finer partitions are also possible). Hence, we can describe the cost coupling via an interference graph $\mathcal{G}^{\mathrm{I}} = (\mathcal{P}, \mathcal{I}, \mathcal{E}^{\mathrm{I}}) = (\mathcal{I}, \mathcal{I}, \mathcal{E}^{\mathrm{I}})$, where $(p, i) \in \mathcal{E}^{\mathrm{I}}$ if and only if f_i depends on x_p, for all $i \neq p$, and $(i, i) \in \mathcal{E}^{\mathrm{I}}$ for all $i \in \mathcal{I}$; we also write

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

Assume that an estimate graph $\mathcal{G}^{\mathrm{E}}$ and design graphs $\mathcal{G}^{\mathrm{D}}$s are chosen according to Assumption 1. Then, each agent i keeps and sends the copies $\{\boldsymbol{y}_{i,p}, p \in \mathcal{N}^{\mathrm{E}}(i)\}$, 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 $\boldsymbol{y}_{i,i} := x_i$ (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 $i \in \mathcal{I}$ performs:

$$\begin{aligned} \hat{\boldsymbol{y}}^k_{i,p} &:= \textstyle\sum_{j \in \mathcal{N}^{\mathrm{D}}_p(i)} [\mathrm{W}^{\mathrm{D}}_p]_{i_p, j_p} \boldsymbol{y}^k_{j,p} && (\forall p \in \mathcal{N}^{\mathrm{E}}(i)) \\ \boldsymbol{y}^{k+1}_{i,p} &= \hat{\boldsymbol{y}}^k_{i,p} && (\forall p \in \mathcal{N}^{\mathrm{E}}(i) \backslash \{i\}) \\ \boldsymbol{y}^{k+1}_{i,i} &= \mathrm{proj}_{\Omega_i}\left(\hat{\boldsymbol{y}}^k_{i,i} - \alpha \nabla_{x_i} f_i((\hat{\boldsymbol{y}}^k_{i,p})_{p \in \mathcal{N}^{\mathrm{I}}(i)})\right). && (18) \end{aligned}$$

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

$$\mathbf{F}(\boldsymbol{y}) := \mathrm{col}((\nabla_{x_i} f_i((\boldsymbol{y}_{i,p})_{p \in \mathcal{N}^{\mathrm{I}}(i)}))_{i \in \mathcal{I}}) \tag{19}$$

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

$$\boldsymbol{y}^{k+1} = \mathrm{proj}_{\boldsymbol{\Omega}}\left(\mathbf{W}^{\mathrm{D}} \boldsymbol{y}^k - \alpha \mathrm{R}^\top \mathbf{F}(\mathbf{W}^{\mathrm{D}} \boldsymbol{y}^k)\right). \tag{20}$$

We now postulate additional conditions on the design graphs.

Assumption 3: For each $i \in \mathcal{I}$, $\mathcal{G}^{\mathrm{D}}_i$ is rooted at i and $\mathrm{W}^{\mathrm{D}}_i \mathbf{1}_{N_i} = \mathbf{1}_{N_i}$; we denote by $q^{\mathrm{D}}_i \in \mathbb{R}^{N_i}$ the unique nonnegative vector such that $q^{\mathrm{D}\top}_i \mathrm{W}^{\mathrm{D}}_i = q^{\mathrm{D}\top}_i$, $\mathbf{1}^\top_{N_i} q^{\mathrm{D}}_i = 1$. □

Assumption 3 is very mild: rootedness is necessary for the consensus of the estimates; row stochasticity is immediately satisfiable whenever the agents have access to their own in-degree. One major technical complication—with respect to the usual, more restrictive strongly connectedness assumption—is that the (Perron) eigenvectors q^{D}_is might have zero elements. In addition, we require one technical condition.

Assumption 4: For all $i \in \mathcal{I}$, there is a matrix $\mathrm{Q}_i \succ 0$ such that $\sigma_i := \|\mathrm{W}^{\mathrm{D}}_i - \mathbf{1}_{N_i} q^{\mathrm{D}\top}_i\|_{\mathrm{Q}_i} < 1$, $[\mathbf{1}^\top_{N_i} \mathrm{Q}_i]_{i_i} = 1$, $1^\top_{N_i} \mathrm{Q}_i \mathrm{W}^{\mathrm{D}}_i (\mathrm{I}_{N_i} - \mathbf{1}_{N_i} q^{\mathrm{D}\top}_i) = \mathbf{0}^\top_{N_i}$, and either (i) Q_i is diagonal or (ii) $\Omega_i = \mathbb{R}^{n_{x_i}}$. □

Remark 2: Assumption 4(i) alone is general enough to comprise all the cases considered in the existing literature [corresponding to the choice $\mathcal{G}^{\mathrm{D}}_p = \mathcal{G}^{\mathrm{C}}$ in (C0)].

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

ii) If $\mathcal{G}^{\mathrm{D}}_i$ is the directed star graph (namely, there are all and only the edges from node i to every node in $\overline{\mathcal{N}}^{\mathrm{E}}(i)$), then Assumption 4(i) holds with $\mathrm{Q}_i = \mathrm{I}$ (and $\sigma_i = 0$, q_i with only one nonzero element $[q_i]_{i_i} = 1$); note that having this structure for all $i \in \mathcal{I}$ 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

$$\Xi := \mathrm{diag}((\mathrm{Q}_i \otimes \mathrm{I}_{n_{x_i}})_{i \in \mathcal{I}}), \qquad \bar{\sigma} := \max_{i \in \mathcal{I}}\{\sigma_i\}$$

$$\bar{\theta} := \theta \sqrt{\max_{i \in \mathcal{I}}\{[\mathrm{Q}_i]_{i_i, i_i}\} / \lambda_{\min}(\Xi)}$$

$$\underline{\gamma} := \sqrt{1/\max_{i \in \mathcal{I}}\{\mathbf{1}^\top \mathrm{Q}_i \mathbf{1}\}}, \quad \bar{\gamma} := \sqrt{1/\min_{i \in \mathcal{I}}\{\mathbf{1}^\top \mathrm{Q}_i \mathbf{1}\}}$$

$$\mathrm{M}_\alpha := \begin{bmatrix} 1 - 2\alpha\mu\underline{\gamma}^2 + \alpha^2\theta^2\bar{\gamma}^2 & \bar{\sigma}(\alpha(\bar{\theta} + \theta\bar{\gamma}) + \alpha^2\bar{\theta}\theta\bar{\gamma}) \\ \bar{\sigma}(\alpha(\bar{\theta} + \theta\bar{\gamma}) + \alpha^2\bar{\theta}\theta\bar{\gamma}) & \bar{\sigma}^2(1 + 2\alpha\bar{\theta} + \alpha^2\bar{\theta}^2) \end{bmatrix}.$$

Let $\alpha > 0$ be chosen such that

$$\rho_\alpha = \lambda_{\max}(\mathrm{M}_\alpha) < 1. \tag{21}$$

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

Proof: See Appendix B-A. ■

The condition (21) always holds for α 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 $\mathcal{G}^{\mathrm{C}}$ 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 $\mathcal{G}^{\mathrm{C}}$ related to $\mathcal{G}^{\mathrm{I}}$, each agent must only estimate the actions that directly affect its cost.[2] 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 $\mathcal{G}_i^{\mathrm{D}}$s that are not strongly connected. For instance, if $\{\mathcal{G}_i^{\mathrm{D}}\}_{i\in\mathcal{I}}$ are all star graphs [i.e., Remark 2(ii)], the action update in (20) is

$$x_i^{k+1} = \mathrm{proj}_{\Omega_i}(x_i^k - \alpha\nabla_{x_i} f_i(x^k)) \tag{22}$$

which is the standard pseudogradient method for the full-information scenario, where estimates and true values coincide. In particular, when $\mathcal{G}^{\mathrm{E}}$ is complete, Theorem 1 retrieves the well-known bound $\alpha < 2\mu/\theta^2$ (since $\bar{\sigma} = 0$, $\underline{\gamma}^2 = \bar{\gamma}^2 = \frac{1}{N}$) [34, Prop. 26.16].

4) Another (not strongly-connected) case not addressed before is that of a matrix $\mathrm{W}_1^{\mathrm{D}} = \begin{bmatrix} 1 & \mathbf{0}_{N_i-1}^\top \\ c & W \end{bmatrix}$, $c \in \mathbb{R}_{\geq 0}^{N_i-1}$, representing a leader–follower protocol (with agent 1 as the leader for ease of notation). If $\Omega_1 = \mathbb{R}^{n_{x_1}}$ and Assumption 3 holds, it can be checked that Assumption 4 is verified with $q_1^{\mathrm{D}} = e_1 = \mathrm{col}(1, 0, \ldots, 0)$ and $\mathrm{Q}_1 = \begin{bmatrix} 1 + \mathbf{1}^\top \mathrm{X}_{22}\mathbf{1} & -\mathbf{1}^\top \mathrm{X}_{2,2} \\ -\mathrm{X}_{22}\mathbf{1}_{N_i-1} & X_{22} \end{bmatrix}$, where $\mathrm{X} = \begin{bmatrix} \mathrm{X}_{1,1} & \mathrm{X}_{1,2} \\ X_{1,2}^\top & \mathrm{X}_{2,2} \end{bmatrix} \succ 0$ is any matrix such that $\|\mathrm{W}_1^{\mathrm{D}} - \mathbf{1}q_1^{\mathrm{D}^\top}\|_{\mathrm{X}} < 1$. As a special case, if $[c]_{j_1} = 1$, we have $\hat{y}_{j,1}^k = x_1^k$, namely, agent j can use the real action x_1 received by agent 1 when evaluating its cost. We can also model a scenario in which the *exact* information on x_1 propagates over $\mathcal{G}_1^{\mathrm{D}}$ but with some delay, by choosing $\mathcal{G}_1^{\mathrm{D}}$ as a directed tree.

5) Each graph $\mathcal{G}_i^{\mathrm{D}}$ can be chosen independently. For instance, one variable x_i might be publicly available (choose $\mathcal{G}_i^{\mathrm{D}}$ 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 M row blocks, i.e., $\mathrm{A} = [\mathrm{A}_{m,i}]_{m\in\mathcal{M},i\in\mathcal{I}}$ and $a = \mathrm{col}((\sum_{i\in\mathcal{I}} a_{m,i})_{m\in\mathcal{M}})$, where, for all $m \in \mathcal{M} := \{1, 2, \ldots, M\}$, $\mathrm{A}_{m,i} \in \mathbb{R}^{n_{\lambda_m}\times n_{x_i}}$ and $a_{m,i} \in \mathbb{R}^{n_{\lambda_m}}$ are local data kept by agent i. The coupling constraint sparsity pattern is described by the interference graph $\mathcal{G}^{\mathrm{I},\lambda} = (\mathcal{M}, \mathcal{I}, \mathcal{E}^{\mathrm{I},\lambda})$, where $(m, i) \in \mathcal{E}^{\mathrm{I},\lambda}$ if agent i is involved in the constraints block indexed by m; in other terms

$$(\forall (m,i) \notin \mathcal{E}^{\mathrm{I},\lambda}) \quad \mathrm{A}_{m,i} = \mathbf{0},\ a_{m,i} = \mathbf{0} \tag{23}$$

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

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 $\sigma : \mathbb{R}^{n_x} \to \mathbb{R}^{n_\sigma}$, so that for all $i \in \mathcal{I}$ and for some function $\bar{f}_i$, we have

$$f_i(x_i, x_{-i}) = \bar{f}_i(x_i, \sigma(x)). \tag{24}$$

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

$$\sigma_q(x) = \sigma_q((x_i)_{i\in\mathcal{N}^{\mathrm{I},\sigma}(q)}) \tag{25}$$

$$\bar{f}_i(x_i, \sigma(x)) = \bar{f}_i(x_i, (\sigma_q(x))_{q\in\mathcal{N}^{\mathrm{I},\sigma}(i)}) \tag{26}$$

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

$$\sigma_q(x) := \textstyle\sum_{i\in\mathcal{N}^{\mathrm{I},\sigma}(q)} \mathrm{B}_{q,i}x_i + b_{q,i} \tag{27}$$

$$\sigma(x) \ = \mathrm{col}((\sigma_q)_{q\in\mathcal{Q}}) = \mathrm{B}x + b \tag{28}$$

$\mathrm{B}_{q,i} \in \mathbb{R}^{n_{\sigma_q}\times n_{x_i}}$ and $b_{q,i} \in \mathbb{R}^{n_{\sigma_q}}$ being local data of agent i, $\mathrm{B} := [\mathrm{B}_{q,i}]_{q\in\mathcal{Q},i\in\mathcal{I}}$, $b = \sum_{i\in\mathcal{I}} \mathrm{col}((b_{q,i})_{q\in\mathcal{Q}})$, with

$$(\forall (q,i) \notin \mathcal{E}^{\mathrm{I},\sigma}) \quad \mathrm{B}_{q,i} = \mathbf{0},\ b_{q,i} = \mathbf{0}. \tag{29}$$

If $M = Q = 1$, we recover the standard generalized aggregative games [7], [36], where each agent must estimate the whole dual variable $\lambda \in \mathbb{R}^{n_\lambda}$ and the whole aggregative value $\sigma(x) \in \mathbb{R}^{n_\sigma}$. 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[3] estimate graphs $\mathcal{G}^{\mathrm{E},\lambda} = (\mathcal{M}, \mathcal{I}, \mathcal{E}^{\mathrm{E},\lambda})$, $\mathcal{G}^{\mathrm{E},\sigma} = (\mathcal{Q}, \mathcal{I}, \mathcal{E}^{\mathrm{E},\sigma})$. We postpone the discussion of a motivating application to Section IV.

[2] This is also achieved in Theorem 1 by the choice $\mathcal{G}_i^{\mathrm{D}} = (\mathcal{N}^{\mathrm{I}}(i), \mathcal{E}^{\mathrm{C}} \cap (\mathcal{N}^{\mathrm{I}}(i) \times \mathcal{N}^{\mathrm{I}}(i)))$ if all the resulting $\{\mathcal{G}_i^{\mathrm{D}}\}_{i\in\mathcal{I}}$ are strongly connected, which is a much weaker assumption than [13, Asm. 6].

[3] Here, we consider END problem for two variables of interest λ and σ. The notation in Section II is recovered by setting $y = \mathrm{col}(\lambda, \sigma(x))$, with $P = M + Q$. However, for readability, here, we treat the two variables separately, as two independent instances of END, where the corresponding quantities in Section II-D are distinguished via the superscripts λ and σ (e.g., $\mathcal{G}_m^{\mathrm{D},\lambda}$ and $\mathcal{G}_q^{\mathrm{D},\sigma}$ describe how the agents exchange the estimates of λ_m and σ_q, respectively). We also indicate the estimates of λ_m and $\sigma_q(x)$ kept by agent i with $\boldsymbol{\lambda}_{i,m}$ and $\boldsymbol{\sigma}_{i,q}$ (in place of $\boldsymbol{y}_{i,m}^\lambda$ and $\boldsymbol{y}_{i,q}^\sigma$) and analogously for the stacked vectors—e.g., $\boldsymbol{\sigma}_q = \mathrm{col}((\boldsymbol{\sigma}_{i,q})_{i\in\mathcal{N}^{\mathrm{E},\sigma}(q)})$, $\boldsymbol{\sigma} = \mathrm{col}((\boldsymbol{\sigma}_q)_{q\in\mathcal{Q}})$ and $\boldsymbol{\lambda}_m = \mathrm{col}((\boldsymbol{\lambda}_{i,m})_{i\in\mathcal{N}^{\mathrm{E},\lambda}(m)})$, $\boldsymbol{\lambda} = \mathrm{col}((\boldsymbol{\lambda}_m)_{m\in\mathcal{M}})$.

Let us define the *extended pseudogradient mapping*

$$\bar{\mathsf{F}}(x,\boldsymbol{\sigma}) := \mathrm{col}((\nabla_{x_i}\bar{f}_i(x_i,\tilde{\boldsymbol{\sigma}}_i) + \tilde{\mathrm{B}}_i^\top \nabla_{\tilde{\boldsymbol{\sigma}}_i}\bar{f}_i(x_i,\tilde{\boldsymbol{\sigma}}_i))_{i\in\mathcal{I}}) \quad (30)$$

where $\tilde{\mathrm{B}}_i := \mathrm{col}((\mathrm{B}_{q,i})_{q\in\mathcal{N}^{\mathrm{E},\sigma}(i)})$, and we recall that $\tilde{\boldsymbol{\sigma}}_i$ are the estimates of the components of σ kept by agent i and (with the customary overloading) $\bar{f}_i(x_i,\tilde{\boldsymbol{\sigma}}_i) = f_i(x_i,(\boldsymbol{\sigma}_{i,q})_{q\in\mathcal{N}^{\mathrm{E},\sigma}(i)}) := f_i(x_i,(\boldsymbol{\sigma}_{i,q})_{q\in\mathcal{N}^{\mathrm{I},\sigma}(i)})$. Note that $\bar{\mathsf{F}}$ coincides with the pseudo-gradient mapping when the estimates are exact and at consensus, i.e., $\bar{\mathsf{F}}(x,\boldsymbol{\mathcal{C}}^\sigma(\sigma(x))) = \mathsf{F}(x)$. We study the following distributed iteration:

$$x^{k+1} = \mathrm{proj}_\Omega\left(x^k - \beta\left(\alpha\bar{\mathsf{F}}(x^k,\boldsymbol{\sigma}^k) + \mathbf{B}^\top\mathbf{L}^{\mathrm{D},\sigma}\boldsymbol{\sigma}^k + \mathbf{A}^\top\boldsymbol{\lambda}^k\right)\right) \quad (31a)$$

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^k - \beta\mathbf{L}^{\mathrm{D},\sigma}\boldsymbol{\sigma}^k + \mathbf{B}(x^{k+1} - x^k) \quad (31b)$$

$$\boldsymbol{z}^{k+1} = \boldsymbol{z}^k + \beta\mathbf{L}^{\mathrm{D},\lambda}\boldsymbol{\lambda}^k \quad (31c)$$

$$\boldsymbol{\lambda}^{k+1} = \mathrm{proj}_{\mathbb{R}^{n_\lambda}_{\geq 0}}\left(\boldsymbol{\lambda}^k - \beta\left(\mathbf{L}^{\mathrm{D},\lambda}(2\boldsymbol{z}^{k+1} - \boldsymbol{z}^k) - \mathbf{A}(2x^{k+1} - x^k) + \boldsymbol{a}\right)\right) \quad (31d)$$

where $\alpha,\beta > 0$ are step sizes; $\boldsymbol{z} = \mathrm{col}((\boldsymbol{z}_m)_{m\in\mathcal{M}}) \in \mathbb{R}^{n_\lambda}$ and $\boldsymbol{z}_m = \mathrm{col}((\boldsymbol{z}_{i,m})_{i\in\overline{\mathcal{N}}^{\mathrm{E},\lambda}(m)})$, where, for all $m \in \mathcal{N}^{\mathrm{E},\lambda}(i)$, $\boldsymbol{z}_{i,m} \in \mathbb{R}^{n_{\lambda m}}$ is an auxiliary dual variable kept by agent i;

$$\mathbf{A} := \mathrm{P}^{\lambda\top}\mathrm{diag}((\tilde{\mathrm{A}}_i)_{i\in\mathcal{I}}), \qquad \tilde{\mathrm{A}}_i := \mathrm{col}((\mathrm{A}_{m,i})_{m\in\mathcal{N}^{\mathrm{E},\sigma}(i)})$$

$$\boldsymbol{a} := \mathrm{P}^{\lambda\top}\mathrm{col}((\tilde{a}_i)_{i\in\mathcal{I}}), \qquad \tilde{a}_i := \mathrm{col}((a_{m,i})_{m\in\mathcal{N}^{\mathrm{E},\lambda}(i)})$$

$$\mathbf{B} := \mathrm{diag}((N_q^\sigma I_{n_{\boldsymbol{\sigma}_q}})_{q\in\mathcal{Q}})\,\mathrm{P}^{\sigma\top}\mathrm{diag}((\tilde{\mathrm{B}}_i)_{i\in\mathcal{I}})$$

$$\boldsymbol{b} := \mathrm{diag}((N_q^\sigma I_{n_{\boldsymbol{\sigma}_q}})_{q\in\mathcal{Q}})\,\mathrm{P}^{\sigma\top}\mathrm{col}((\tilde{b}_i)_{i\in\mathcal{I}})$$

$$\tilde{b}_i := \mathrm{col}((b_{q,i})_{q\in\mathcal{N}^{\mathrm{E},\sigma}(i)})$$

and we recall that $N_q^\sigma = |\overline{\mathcal{N}}^{\mathrm{E},\sigma}(q)|$, that with $\tilde{}$ we indicate agentwise stacked quantities, and that P^λ and P^σ are permutation matrices (i.e., $\mathrm{P}^{\lambda\top}\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$ and $\mathrm{P}^{\sigma\top}\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$). We impose

$$\boldsymbol{\sigma}^0 = \mathbf{B}x^0 + \boldsymbol{b} \quad (32)$$

or, agentwise, $\tilde{\boldsymbol{\sigma}}_i^0 = \mathrm{col}((N_q^\sigma\mathrm{B}_{q,i}x_i^0 + N_q^\sigma b_{q,i})_{q\in\mathcal{N}^{\mathrm{E},\sigma}(i)})$; the other variables are initialized arbitrarily.

The algorithm is based on primal-dual pseudogradient dynamics, where the update of each $\boldsymbol{\sigma}_q$ represents a dynamic tracking of the aggregation function σ_q, over the graph $\mathcal{G}_q^{\mathrm{D},\sigma}$. 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 $m \in \mathcal{M}$, $\mathcal{G}_m^{\mathrm{D},\lambda}$ is undirected and connected and $\mathrm{W}_m^{\mathrm{D},\lambda}$ is symmetric. For each $q \in \mathcal{Q}$, $\mathcal{G}_q^{\mathrm{D},\sigma}$ is strongly connected and $\mathrm{W}_q^{\mathrm{D},\sigma}$ is balanced. □

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

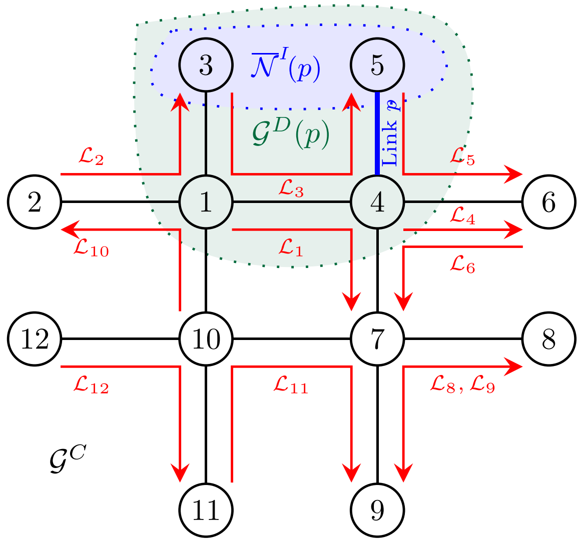


Fig. 2. Unicast rate allocation game. The main complication is that some users send data over the same link but cannot communicate: for instance, link (4,5)—labeled p—is used by users 3 and 5 (i.e., $\overline{\mathcal{N}}^{\mathrm{I}}(p) = \{3,5\}$), which are not communication neighbors.

Proof: See Appendix B-B, where we also provide explicit bounds for α and β. ■

To our knowledge, we are the first to consider the partial coupling in the constraints or aggregation in generalized games. For nongeneralized games, an approach is studied in [24], where the cost of each agent is only affected by some of the components of an aggregation function: nevertheless, this algorithm requires strong conditions on the (undirected) communication network—necessary to allow the choice $\mathcal{G}^{\mathrm{I}} = \mathcal{G}^{\mathrm{E}}$, see also Example 3 in Appendix A. Besides avoiding this limitation, we also considered the presence of coupling constraints and a more general aggregation function.

The efficacy and improved scalability guaranteed by the method in (31) are demonstrated numerically in the next section.

IV. Illustrative Example: Unicast Rate Allocation

We study a bandwidth allocation problem with fixed routing [11], [24], modeled as a GNE problem—see Fig. 2 for an illustration. Consider an undirected connected communication network $\mathcal{G}^{\mathrm{C}} = (\mathcal{I},\mathcal{E}^{\mathrm{C}})$. From each node $i \in \mathcal{I}$, a user sends data with rate $x_i \in [0,1]$, over a path $\mathcal{L}_i$—i.e., a sequence of consecutive edges over the graph $\mathcal{G}^{\mathrm{C}}$. User i aims to choose x_i to minimize the cost function

$$J_i(x_i,x_{-i}) = J_i(x_i,\{\sigma_l\}_{l\in\mathcal{L}i}) = -U(x_i) + \textstyle\sum_{l\in\mathcal{L}_i} c_l(x_i,\sigma_l)$$

$$\sigma_l := \textstyle\sum_{j|l\in\mathcal{L}_j} x_j$$

where U_i is an utility function, σ_l represents the aggregative rate over the link (i.e., edge) l, and c_l is a penalty related to link l (e.g., quantifying the loss of service quality due to congestion [11] or a tax imposed by a network manager [24]). Furthermore, the capacity of each link l is bounded by the coupling constraints

$\sigma_l \leq a_l$, for some $a_l > 0$. The objective is to seek a v-GNE of the resulting generalized game, when the users can only communicate over the graph $\mathcal{G}^{\mathrm{C}}$. The capacity of each link can change over time [11] (e.g., because some external primary users have priority on the bandwidth use); thus, the users might have to solve the problem multiple times. Therefore, it is important for the users to solve the problem as efficiently as possible to quickly adapt to the changes.

Let us relabel the "active" edges $\bigcup_{i\in\mathcal{I}}\{l \in \mathcal{L}_i\}$ (i.e., the edges of $\mathcal{G}^{\mathrm{C}}$ hosting at least one path) as $\{1, 2, \dots, P\} =: \mathcal{P}$; further, let us define for all $p \in \mathcal{P}$ and $i \in \mathcal{I}$

$$\mathrm{A}_{p,i} = \mathrm{B}_{p,i} = \begin{cases} 1, & \text{if } p \in \mathcal{L}_i \text{ (user } i \text{ uses link } p\text{)} \\ 0, & \text{otherwise} \end{cases}$$

the so-called *routing* matrix [11]. With these definitions, the problem retrieves an aggregative game of the type described in Section III-B, with $\mathcal{M} = \mathcal{Q} = \mathcal{P}$. In particular, only the aggregative variables and constraints relative to the links in $\mathcal{L}_i$ directly affect agent i, i.e., $\mathcal{N}^{\mathrm{I},\sigma}(i) = \mathcal{N}^{\mathrm{I},\lambda}(i) = \mathcal{L}_i$.[4]

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 $\{\mathcal{G}^{\mathrm{C}}|_{\overline{\mathcal{N}}^{\mathrm{I}}(p)}\}_{p\in\mathcal{P}}$ 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 $\mathrm{W}_p^{\mathrm{D},\sigma} = \mathrm{W}_p^{\mathrm{D},\lambda} =: \mathrm{W}_p^{\mathrm{D}}$ 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:* $\mathrm{W}_p^{\mathrm{D}}$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:* $\mathrm{W}_p^{\mathrm{D}}$s are chosen to minimize the per-iteration communication cost, by solving Design Problem 1 with: "(iii) For all $p \in \mathcal{P}$, $\mathcal{G}_p^{\mathrm{D}}$ is connected and has minimal number of edges."

We note that, for the customized algorithm, the design of each graph $\mathrm{W}_p^{\mathrm{D}}$ corresponds to solving an ST problem as in Fig. 1. 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).

We choose $U_i(x_i) = 10 * \log(x_i + 1)$, $c_l(x_i, \sigma_l) = \psi_l \frac{x_i}{1+e^{-\sigma_l}}$ with ψ_l 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 $x^\star$ 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 with the problem dimension, we expect the gap between the two methods to increase for larger networks. Simulations with $I = 7, 12, 16, 20$ ($P = 7, 16, 18, 26$) confirm this intuition: for the case $I = 20$, the customized algorithm saves 99% of the communication cost (where sending one scalar value to *one* neighbor on $\mathcal{G}^{\mathrm{C}}$ costs 1, in unicast fashion). Finally, for the selected parameters and $I \geq 24$, the standard algorithm fails to converge in our simulations, while the customized algorithm converges (at least) up to $I = 50$—suggesting tolerance to larger step sizes, with upper bounds less affected by the problem dimension.

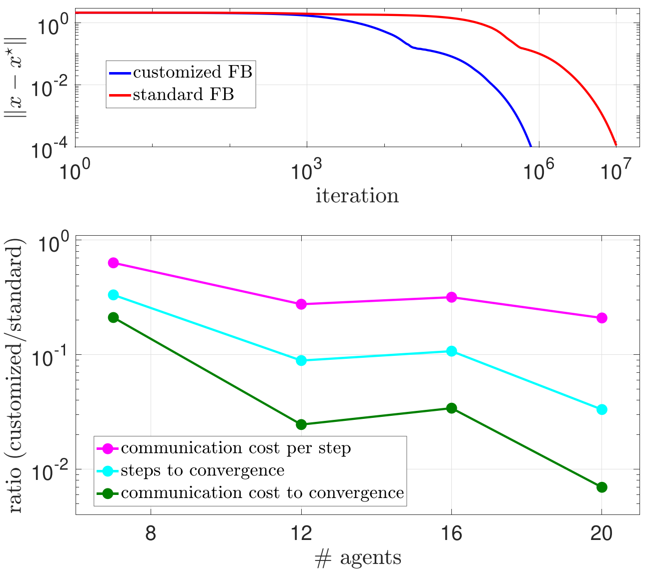


Fig. 3. Unicast rate allocation via algorithm (31), for the scheme in Fig. 2 (top) and for different randomly generated networks, with maximal path length 4 and stopping criterion $\|x - x^\star\| \leq 10^{-2}$ (bottom).

In conclusion, while requiring some initial design effort, the customized method can result in substantial efficiency improvement, especially if the v-GNE problem is solved multiple times due to time-varying cost parameters or link capacities.

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].

[4] The setup also fits more general cases where from a node i, multiple distinct flows $x_{i,1}, \dots, x_{1,n_{x_i}}$ are sent over distinct paths $\mathcal{L}_{i,1}, \dots, \mathcal{L}_{i,n_{x_i}}$, either by a single user (adjusting the cost function) or by multiple competitive users (considering an augmented communication graph $\mathcal{G}^{\mathrm{C}'} \supset \mathcal{G}^{\mathrm{C}}$).

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 $\mathcal{P}$, the number of copies of y_p, provided that the conditions in Design Problem 1(i) and (ii) are satisfied and the graphs $\{\mathcal{G}^{\mathrm{D}}_p\}_{p\in\mathcal{P}}$ enjoy some connectivity properties. In particular, consider Design Problem 1 with (iii) for each $p \in \mathcal{P}$, $\mathcal{G}^{\mathrm{D}}_p$ is rooted at $r_p \in \mathcal{I}$ (respectively, $\mathcal{G}^{\mathrm{D}}_p$ is strongly connected); the number $|\overline{\mathcal{N}}^{\mathrm{E}}(p)|$ of nodes in $\mathcal{G}^{\mathrm{D}}_p$ 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 $\mathcal{G}^{\mathrm{D}}_p$ 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 $\mathcal{G}^{\mathrm{C}}$ 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 $\mathcal{G}'$ for $\mathcal{G}^{\mathrm{D}}_p$, any graph $\mathcal{G}''$ such that $\mathcal{G}' \subseteq \mathcal{G}'' \subseteq \mathcal{G}^{\mathrm{C}}$ is is also a solution for Design Problem 1. In simple terms, we can add edges to $\mathcal{G}'$, 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 $\mathcal{G}^{\mathrm{D}}_p$ (see Fig. 1).

B. Fair Allocation and Bandwidth Constraints: Instead of minimizing the overall dimension of $\boldsymbol{y}$, 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 $\mathcal{G}^{\mathrm{D}}_p$s as solutions of an ST problem (see Section I-A), but sequentially penalizing unbalanced allocations by opportunely choosing the edge weights for each p. Bandwidth constraints can be addressed similarly, to avoid overloading some channels of the communication network $\mathcal{G}^{\mathrm{C}}$.

C. Designing the Communication Graph: In this article, we consider the graph $\mathcal{G}^{\mathrm{C}}$ 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 $\mathcal{G}^{\mathrm{C}}$ is complete; then, the graphs $\{\mathcal{G}^{\mathrm{D}}_p\}_{p\in\mathcal{P}}$ can be chosen to fulfill some specifications (e.g., minimize the number of active edges in $\mathcal{G}^{\mathrm{C}}$—which determines the physical channels/edges actually needed).

D. Straightforward Designs: 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

$$\min_{y_i\in\mathbb{R}^{n_{y_i}},i\in\mathcal{I}} \textstyle\sum_{i\in\mathcal{I}} f_i(y_i,(y_j)_{j\in\mathcal{N}^{\mathrm{C}}(i)}) \tag{33}$$

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 $\mathcal{G}^{\mathrm{C}}$: this is a special case of Example 1, with $\mathcal{P}=\mathcal{I}$, $\overline{\mathcal{N}}^{\mathrm{I}}(i)=\mathcal{N}^{\mathrm{C}}(i)\cup\{i\}$. Consider Design Problem 1, with "(iii) $\forall i\in\mathcal{I}$, $\mathcal{G}^{\mathrm{D}}_i$ is connected; $\mathcal{G}^{\mathrm{D}}_i$ has minimum number of nodes (provided that all other specifications are met)." A solution is to fix each $\mathcal{G}^{\mathrm{D}}_i$ as the undirected star graph centered in i with vertices $\mathcal{V}^{\mathrm{D}}_i=\overline{\mathcal{N}}^{\mathrm{I}}(i)$: 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], and [40]. □

Example 3 ($\mathcal{G}^{\mathrm{E}}=\mathcal{G}^{\mathrm{I}}$): With the goal of minimizing the overall memory allocation, consider the choice

$$\mathcal{G}^{\mathrm{D}}_p=\mathcal{G}^{\mathrm{C}}|_{\mathcal{N}^{\mathrm{I}}(p)} \qquad \forall p\in\mathcal{P}. \tag{34}$$

In this case, $\mathcal{G}^{\mathrm{E}}=\mathcal{G}^{\mathrm{I}}$, 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 $\{\mathcal{G}^{\mathrm{D}}_i\}_{i\in\mathcal{I}}$ ensure the desired connectedness properties in Design Problem 1(iii), which usually not verified (see $\mathcal{G}^{\mathrm{D}}_1$ in Fig. 1), but holds in some particular cases (e.g., Example 2, $\mathcal{G}^{\mathrm{D}}_2$ in Fig. 1; see also [10, Asm. 5], [13, 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], for NE problems arising in social networks, where the cost of agent i is $f_i(y_i,(y_j)_{j\in\mathcal{N}^{\mathrm{C}}(i)},(y_\ell)_{l\in\mathcal{N}^{\mathrm{C}}(j),j\in\mathcal{N}^{\mathrm{C}}(i)})$, i.e., the cost of each agent depends on its own action y_i and on the actions of its (in-)neighbors and neighbors' neighbors. □

Appendix B
Proofs

A. Proof of Theorem 1

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

Lemma 3: Let $\mathcal{F}(\boldsymbol{y}):=\mathbf{W}^{\mathrm{D}}\boldsymbol{y}-\alpha\mathrm{R}^\top\mathbf{F}(\mathbf{W}^{\mathrm{D}}\boldsymbol{y})$. Then, for any $\boldsymbol{y}\in\mathbb{R}^{n_{\boldsymbol{y}}}$, it holds that

$$\|\mathcal{F}(\boldsymbol{y})-\mathcal{F}(\boldsymbol{y}^\star)\|_\Xi\leq\sqrt{\rho_\alpha}\|\boldsymbol{y}-\boldsymbol{y}^\star\|_\Xi.\square$$

Proof: Let $\boldsymbol{y}=\boldsymbol{y}_{\parallel}+\boldsymbol{y}_{\perp}$, where $\boldsymbol{y}_{\parallel}:=\operatorname{diag}(((\mathbf{1}_{N_i}q_i^{\mathrm{D}\top})\otimes\mathrm{I}_{n_{x_i}})_{i\in\mathcal{I}})\boldsymbol{y}\in\boldsymbol{\mathcal{C}}$, and thus, $\boldsymbol{y}_{\parallel}=\boldsymbol{\mathcal{C}}(y_{\parallel})$ for some $y_{\parallel}\in\mathbb{R}^{n_y}$. Let $\hat{\boldsymbol{y}}:=\mathbf{W}^{\mathrm{D}}\boldsymbol{y}=\boldsymbol{y}_{\parallel}+\hat{\boldsymbol{y}}_{\perp}$, where $\hat{\boldsymbol{y}}_{\perp}:=\mathbf{W}^{\mathrm{D}}\boldsymbol{y}_{\perp}=\mathbf{W}^{\mathrm{D}}\operatorname{diag}(((\mathrm{I}-\mathbf{1}_{N_i}q_i^{\mathrm{D}\top})\otimes\mathrm{I}_{n_{x_i}})_{i\in\mathcal{I}})\boldsymbol{y}$ and we used that $\mathbf{W}^{\mathrm{D}}\boldsymbol{y}_{\parallel}=\boldsymbol{y}_{\parallel}=\boldsymbol{\mathcal{C}}(y_{\parallel})$ (by row stochasticity). By Assumption 4, we have $q_i^{\mathrm{D}}\mathbf{1}_{N_i}^\top\mathrm{Q}_i\mathrm{W}_i^{\mathrm{D}}(\mathrm{I}-\mathbf{1}_{N_i}q_i^{\mathrm{D}\top})=\mathbf{0}$ and hence $\langle\boldsymbol{y}_{\parallel},\hat{\boldsymbol{y}}_{\perp}\rangle_\Xi=0$. Therefore

$$\begin{aligned}
&\|(\hat{\boldsymbol{y}}-\alpha\mathrm{R}^\top\mathbf{F}(\hat{\boldsymbol{y}}))-(\boldsymbol{y}^\star-\alpha\mathrm{R}^\top\mathbf{F}(\boldsymbol{y}^\star))\|_\Xi^2\\
&=\|\boldsymbol{y}_{\parallel}-\boldsymbol{y}^\star\|_\Xi^2+\|\hat{\boldsymbol{y}}_{\perp}\|_\Xi^2\\
&\quad+\alpha^2\|\mathrm{R}^\top(\mathbf{F}(\hat{\boldsymbol{y}})-\mathbf{F}(\boldsymbol{y}_{\parallel})+\mathbf{F}(\boldsymbol{y}_{\parallel})-\mathbf{F}(\boldsymbol{y}^\star))\|_\Xi^2
\end{aligned}$$

$$\begin{aligned}
&- 2\alpha\langle \hat{\boldsymbol{y}}_{\perp}, \mathrm{R}^{\top}(\mathbf{F}(\hat{\boldsymbol{y}}) - \mathbf{F}(\boldsymbol{y}^{\star}))\rangle_{\Xi} \\
&- 2\alpha\langle \boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}, \mathrm{R}^{\top}(\mathbf{F}(\hat{\boldsymbol{y}}) - \mathbf{F}(\boldsymbol{y}_{\parallel}))\rangle_{\Xi} \\
&- 2\alpha\langle \boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}, \mathrm{R}^{\top}(\mathbf{F}(\boldsymbol{y}_{\parallel}) - \mathbf{F}(\boldsymbol{y}^{\star}))\rangle_{\Xi} \\
\leq{}& \|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi}^2 + \|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi}^2 + \alpha^2(\bar{\theta}\|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi} + \theta\bar{\gamma}\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}_{\parallel}\|_{\Xi})^2 \\
&+ 2\alpha\bar{\theta}\|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi}(\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi} + \|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi}) \\
&+ 2\alpha\theta\bar{\gamma}\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi}\|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi} - 2\alpha\mu\underline{\gamma}^2\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi}^2. \qquad (35)
\end{aligned}$$

where the last inequality follows by the Cauchy–Schwarz inequality and using that $\|\mathrm{R}^{\top}v\|_{\Xi}^2 \leq \max_{i\in\mathcal{I}}\{[\mathrm{Q}_i]_{i_i,i_i}\}\|v\|^2$ for all $v \in \mathbb{R}^{n_x}$, because $\mathrm{R}\Xi\mathrm{R}^{\top} = \mathrm{diag}(([\mathrm{Q}_i]_{i_i,i_i}\mathrm{I})_{i\in\mathcal{I}})$, that $\mathbf{F}$ is θ-Lipschitz continuous if F is (see [32, Lem. 1]), and, finally, that $\|\mathbf{F}(\boldsymbol{y}_{\parallel}) - \mathbf{F}(\boldsymbol{y}^{\star})\|^2 = \|\mathsf{F}(y_{\parallel}) - \mathsf{F}(x^{\star})\|^2 \leq \theta^2\|y_{\parallel} - x^{\star}\| = \theta^2\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\mathrm{diag}((\mathrm{Q}_i\otimes\mathrm{I}/(\mathbf{1}^{\top}\mathrm{Q}_i\mathbf{1}))_{i\in\mathcal{I}})}$ (the last equality due to $(\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}) \in \boldsymbol{\mathcal{C}}$) and similarly that $\langle \boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}, \mathrm{R}^{\top}(\mathbf{F}(\boldsymbol{y}_{\parallel}) - \mathbf{F}(\boldsymbol{y}^{\star}))\rangle_{\Xi} = \langle y_{\parallel} - x^{\star}, \mathsf{F}(y_{\parallel}) - \mathsf{F}(x^{\star})\rangle \geq \mu\|y_{\parallel} - x^{\star}\|^2 = \mu\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|^2_{\mathrm{diag}((\mathrm{Q}_i\otimes\mathrm{I}/(\mathbf{1}^{\top}\mathrm{Q}_i\mathbf{1}))_{i\in\mathcal{I}})}$, (by using the normalization $[\mathbf{1}^{\top}\mathrm{Q}_i]_{i_i} = 1$ in Assumption 4). In addition, by Assumption 3, we have $\|\hat{\boldsymbol{y}}_{\perp}\|_{\Xi} = \|\mathbf{W}^{\mathrm{D}}\boldsymbol{y}_{\perp}\|_{\Xi} \leq \bar{\sigma}\|\boldsymbol{y}_{\perp}\|_{\Xi}$; together with (35), this yields

$$\begin{aligned}
\|\mathcal{F}(\boldsymbol{y}) - \mathcal{F}(\boldsymbol{y}^{\star}))\|_{\Xi} &\leq \begin{bmatrix} \|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi} \\ \|\boldsymbol{y}_{\perp}\|_{\Xi} \end{bmatrix}^{\top} \mathrm{M}_{\alpha} \begin{bmatrix} \|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi} \\ \|\boldsymbol{y}_{\perp}\|_{\Xi} \end{bmatrix} \\
&\leq \lambda_{\max}(\mathrm{M}_{\alpha})(\|\boldsymbol{y}_{\parallel} - \boldsymbol{y}^{\star}\|_{\Xi}^2 + \|\boldsymbol{y}_{\perp}\|_{\Xi}^2) \\
&= \lambda_{\max}(\mathrm{M}_{\alpha})\|\boldsymbol{y} - \boldsymbol{y}^{\star}\|_{\Xi}^2.
\end{aligned}$$

■

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

$$\begin{aligned}
\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^{\star}\|_{\Xi} &= \|\,\mathrm{prox}_{\boldsymbol{g}}^{\Xi}(\mathcal{F}(\boldsymbol{y})) - \mathrm{prox}_{\boldsymbol{g}}^{\Xi}(\mathcal{F}(\boldsymbol{y}^{\star}))\|_{\Xi} \\
&\leq \|\mathcal{F}(\boldsymbol{y}) - \mathcal{F}(\boldsymbol{y}^{\star})\|_{\Xi}
\end{aligned}$$

and the conclusion follows by Lemma 3. ■

B. Proof of Theorem 2

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

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

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

$$\mathfrak{A}(\boldsymbol{\omega}) := \underbrace{\begin{bmatrix} \alpha\bar{\mathbf{F}}(x,\boldsymbol{\sigma}) + \mathbf{B}^{\top}\mathbf{L}^{\mathrm{D},\sigma}\boldsymbol{\sigma} \\ \mathbf{L}^{\mathrm{D},\sigma}\boldsymbol{\sigma} \\ \mathbf{0} \\ \boldsymbol{b} \end{bmatrix}}_{:=\mathfrak{A}_1} + \underbrace{\begin{bmatrix} \mathbf{A}^{\top}\boldsymbol{\lambda} \\ \mathbf{0} \\ -\mathbf{L}^{\mathrm{D},\lambda}\boldsymbol{\lambda} \\ \mathbf{L}^{\mathrm{D},\lambda}\boldsymbol{z} - \mathbf{A}x \end{bmatrix}}_{:=\mathfrak{A}_2} + \underbrace{\begin{bmatrix} \mathsf{N}_{\Omega} \\ \mathbf{0} \\ \mathbf{0} \\ \mathsf{N}_{\mathbb{R}_{\geq 0}^{n_{\lambda}}} \end{bmatrix}}_{:=\mathfrak{A}_3} \qquad (37)$$

$$\Phi := \begin{bmatrix} \beta^{-1}\mathrm{I} & \mathbf{0} & \mathbf{0} & -\mathbf{A}^{\top} \\ \mathbf{0} & \beta^{-1}\mathrm{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \beta^{-1}\mathrm{I} & \mathbf{L}^{\mathrm{D},\lambda} \\ -\mathbf{A} & \mathbf{0} & \mathbf{L}^{\mathrm{D},\lambda} & \beta^{-1}\mathrm{I} \end{bmatrix} \qquad (38)$$

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

Lemma 4 (Invariance): For all $k \in \mathbb{N}$, $\Pi_{\parallel}^{\sigma}\boldsymbol{s}^k = \mathbf{0}$. □

Proof: Via induction and by (36), since $\Pi_{\parallel}^{\sigma}\mathbf{L}^{\mathrm{D},\sigma} = \mathbf{0}$. ■

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

$$\mathfrak{A}_2(\boldsymbol{\omega}^{k+1}) + \mathfrak{A}_3(\boldsymbol{\omega}^{k+1}) + \Phi(\boldsymbol{\omega}^{k+1} - \boldsymbol{\omega}^k) \ni -\mathfrak{A}_1(\boldsymbol{\omega}^k) \qquad (39)$$

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

Proof: The iteration in (31a), (31c), (31d), and (36) is retrieved expanding the terms in (39) and canceling identical terms, by noting that $\mathfrak{A}_3$ is the normal cone of the set $\mathcal{S} = \Omega \times \mathbb{R}^{n_{\sigma}} \times \mathbb{R}^{n_{\lambda}} \times \mathbb{R}_{\geq 0}^{n_{\lambda}}$, and hence, $\boldsymbol{\omega}^{k+1} + \mathfrak{A}_3(\boldsymbol{\omega}^{k+1}) \ni \boldsymbol{\omega}'$ is equivalent to $\boldsymbol{\omega}^{k+1} = \mathrm{proj}_{\mathcal{S}}(\boldsymbol{\omega}')$ for any $\boldsymbol{\omega}'$. ■

Lemma 6 (Fixed points): The fixed points of the iteration in (39) coincide with $\mathrm{zer}(\mathfrak{A})$. The set $\mathrm{zer}(\mathfrak{A}) \cap \Sigma := \{\boldsymbol{\omega} \mid \Pi_{\parallel}^{\sigma}\boldsymbol{s} = \mathbf{0}\}$ is nonempty. Moreover, for any $\boldsymbol{\omega}^{\star} = (x^{\star}, \boldsymbol{s}^{\star}, \boldsymbol{z}^{\star}, \boldsymbol{\lambda}^{\star}) \in \mathrm{zer}(\mathfrak{A}) \cap \Sigma$, we have that $\boldsymbol{\sigma}^{\star} := \boldsymbol{s}^{\star} + \mathbf{B}x^{\star} + \boldsymbol{b} = \boldsymbol{\mathcal{C}}^{\sigma}(\sigma(x^{\star}))$, $\boldsymbol{\lambda}^{\star} = \boldsymbol{\mathcal{C}}^{\lambda}(\lambda^{\star})$, where $(x^{\star}, \lambda^{\star})$ solves (17); hence, $x^{\star}$ is the v-GNE of the game in (15). □

Proof: By the definition of inverse operator, (39) is equivalent to $\boldsymbol{\omega}^{k+1} = (\mathsf{Id} + \Phi^{-1}(\mathfrak{A}_2 + \mathfrak{A}_3))^{-1}(\mathsf{Id} - \Phi^{-1}\mathfrak{A}_3)(\boldsymbol{\omega}^k)$, i.e., the FB algorithm [34, Sec. 26.5] applied to the operator $\Phi^{-1}\mathfrak{A}$; its fixed points are the zeros of $\Phi^{-1}\mathfrak{A}$ [34, Prop. 26.1(iv)], and clearly, $\mathrm{zer}(\Phi^{-1}\mathfrak{A}) = \mathrm{zer}(\mathfrak{A})$. Now, consider any $\boldsymbol{\omega}^{\star} \in \mathrm{zer}(\mathfrak{A}) \cap \Sigma$. Note that $\boldsymbol{\sigma}^{\star} = \Pi_{\parallel}^{\sigma}\boldsymbol{\sigma}^{\star} = \Pi_{\parallel}^{\sigma}\boldsymbol{s}^{\star} + \Pi_{\parallel}^{\sigma}\mathbf{B}x^{\star} + \Pi_{\parallel}^{\sigma}\boldsymbol{b} = \boldsymbol{\mathcal{C}}^{\sigma}(\sigma(x^{\star}))$, where the first equality follows by the second row in (37) and Assumption 5 (and Lemma 1), and the second by definition of $\mathbf{B}$, $\boldsymbol{b}$, and $\Pi_{\parallel}^{\sigma}$. By the third row in (37), $\boldsymbol{\lambda}^{\star} \in \boldsymbol{\mathcal{C}}^{\lambda}$; 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 $\Pi_{\parallel}^{\lambda}$. Conversely, it can be shown as in [7, Lem. 10] that, for any $(x^{\star}, \lambda^{\star})$ solving (17) (at least one such pair exists by Assumption 2), there exists $\boldsymbol{z}^{\star} \in \mathbb{R}^{n_{\lambda}}$ such that $(x^{\star}, \boldsymbol{s}^{\star} := \boldsymbol{\mathcal{C}}^{\sigma}(\sigma(x^{\star})) - \mathbf{B}x^{\star} - \boldsymbol{b}, \boldsymbol{z}^{\star}, \boldsymbol{\mathcal{C}}^{\lambda}\lambda^{\star}) \in \mathrm{zer}(\mathfrak{A})$; therefore, $\mathrm{zer}(\mathfrak{A}) \cap \Sigma \neq \varnothing$, since $\Pi_{\parallel}^{\sigma}\boldsymbol{s}^{\star} = \mathbf{0}$. ■

Lemma 7 (Monotonicity properties): The operator $\mathfrak{A}_2 + \mathfrak{A}_3$ is maximally monotone [34, Def. 20.20]. Let $\bar{\lambda} := \lambda_2(\frac{\mathrm{L}^{\mathrm{D},\sigma} + \mathrm{L}^{\mathrm{D},\sigma^\top}}{2})$; then, for any $0 < \alpha < \frac{4\mu\bar{\lambda}}{\bar{\theta}^2}$, the operator $\mathfrak{A}_1$ is η_α-restricted cocoercive, for some $\eta_\alpha > 0$ depending on α: for all $\boldsymbol{\omega} \in \Sigma$ and all $\boldsymbol{\omega}^\star \in \mathrm{zer}(\mathfrak{A}) \cap \Sigma$, $\langle \mathfrak{A}_1(\boldsymbol{\omega}) - \mathfrak{A}_1(\boldsymbol{\omega}^\star), \boldsymbol{\omega} - \boldsymbol{\omega}^\star \rangle \geq \eta_\alpha \| \mathfrak{A}_1(\boldsymbol{\omega}) - \mathfrak{A}_1(\boldsymbol{\omega}^\star)\|^2$. □

Proof: $\mathfrak{A}_2$ is maximally monotone because it is a skew-symmetric linear operator [34, Ex. 20.35], the normal cone $\mathfrak{A}_3$ is maximally monotone by [34, Ex. 20.26]; then, since $\mathfrak{A}_2$ has full domain, $\mathfrak{A}_2 + \mathfrak{A}_3$ is maximally monotone by [34, Th. 25.2]. For the second statement, for all $\boldsymbol{\omega} \in \Sigma$, we have $\langle \boldsymbol{\omega} - \boldsymbol{\omega}^\star, \mathfrak{A}_1(\boldsymbol{\omega}) - \mathfrak{A}_1(\boldsymbol{\omega}^\star) \rangle = \alpha \langle x - x^\star, \bar{\mathbf{F}}(x, \boldsymbol{\sigma}) - \bar{\mathbf{F}}(x^\star, \boldsymbol{\sigma}^\star) \rangle + \langle \boldsymbol{\sigma} - \boldsymbol{\sigma}^\star, \mathbf{L}^{\mathrm{D},\sigma}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^\star) \rangle$, and the result follows identically to [36, Lemma 5] by using Lemma 2, that $\bar{\mathbf{F}}$ is $\bar{\theta}$-Lipschitz, that $\bar{\mathbf{F}}(x, \Pi^{\sigma}_{\parallel} \boldsymbol{\sigma}) = F(x)$, and $F(x)$ is μ-strongly monotone (we refer to [36] for the expression of η_α). ■

Lemma 5 recasts (31) as a preconditioned FB algorithm [6], applied to the operators $\mathfrak{A}_1$ and $\mathfrak{A}_2 + \mathfrak{A}_3$, with preconditioning matrix $\Phi \succ 0$. In turn, Lemma 7 ensures the conditions on the operators $\mathfrak{A}_1$ and $\mathfrak{A}_2 + \mathfrak{A}_3$ that guarantee the convergence of the preconditioned FB method to a fixed point in $\mathrm{zer}(\mathfrak{A}) \cap \Sigma$, provided that the step size β is chosen such that $0 < \beta < 2\eta_\alpha \delta$ (this can be proven by showing the decrease of the Lyapunov function $\|\boldsymbol{\omega}^k - \boldsymbol{\omega}^\star\|_\Phi$, with analysis restricted to the invariant subspace Σ, 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 C. Kanzow, "Generalized Nash equilibrium problems," *4OR*, vol. 5, pp. 461–464, 2007.

[3] H.-B. Dürr, M. S. Stanković, and K. H. Johansson, "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. no. 110080.

[8] D. Gadjov 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. Salehisadaghiani and L. Pavel, "Distributed Nash equilibrium seeking in networked graphical games," *Automatica*, vol. 87, pp. 17–24, 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.

[13] F. Salehisadaghiani and L. Pavel, "Nash equilibrium seeking with non-doubly stochastic communication weight matrix," *EAI Endorsed Trans. Collaborative Comput.*, vol. 4, pp. 3–15, 2020.

[14] I. Necoara and D. Clipici, "Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds," *SIAM J. Optim.*, vol. 26, no. 1, pp. 197–226, 2016.

[15] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "Distributed optimization with local domains: Applications in MPC and network flows," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 2004–2009, Jul. 2015.

[16] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A proximal diffusion strategy for multiagent optimization with sparse affine constraints," *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4554–4567, Nov. 2020.

[17] S. A. Alghunaim and A. H. Sayed, "Distributed coupled multiagent stochastic optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 175–190, Jan. 2020.

[18] I. Notarnicola, R. Carli, and G. Notarstefano, "Distributed partitioned big-data optimization via asynchronous dual decomposition," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1910–1919, Dec. 2018.

[19] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[20] D. Fullmer and A. S. Morse, "A distributed algorithm for computing a common fixed point of a finite family of paracontractions," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 2833–2843, Sep. 2018.

[21] F. Salehisadaghiani and L. Pavel, "Distributed Nash equilibrium seeking: A gossip-based algorithm," *Automatica*, vol. 72, pp. 209–216, 2016.

[22] M. Bianchi and S. Grammatico, "Nash equilibrium seeking under partial-decision information over directed communication networks," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, pp. 3555–3560.

[23] T. Tatarenko, W. Shi, and A. Nedic, "Geometric convergence of gradient play algorithms for distributed Nash equilibrium seeking," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5342–5353, Nov. 2021.

[24] E. Eslami, H. Kebriaei, H. Atrianfar, and F. Farhadi, "Incentive mechanism design for unicast service provisioning with network aggregative game," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 572–579, Jan. 2023.

[25] M. Bianchi and S. Grammatico, "The END: Estimation network design for games under partial-decision information," 2023, *arXiv:2208.11377*.

[26] P. Chalermsook and J. Fakcharoenphol, "Simple distributed algorithms for approximating minimum Steiner trees," in *Computing and Combinatorics*, L. Wang, Ed. Berlin, Germany: Springer, 2005, pp. 380–389.

[27] M. Bianchi, E. Benenati, and S. Grammatico, "Linear convergence in time-varying generalized Nash equilibrium problems," in *Proc. IEEE 62nd Conf. Decis. Control*, 2023, pp. 7220–7226.

[28] J. Xu, Y. Tian, Y. Sun, and G. Scutari, "Distributed algorithms for composite optimization: Unified framework and convergence analysis," *IEEE Trans. Signal Process.*, vol. 69, pp. 3555–3570, 2021.

[29] J. Koshal, A. Nedić, and U. V. Shanbhag, "Distributed algorithms for aggregative games on graphs," *Oper. Res.*, vol. 64, pp. 680–704, 2016.

[30] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York, NY, USA: Springer, 2007.

[31] B. Wang, Y. Wu, and K. R. Liu, "Game theory for cognitive radio networks: An overview," *Comput. Netw.*, vol. 54, no. 14, pp. 2537–2561, 2010.

[32] M. Bianchi and S. Grammatico, "Fully distributed Nash equilibrium seeking over time-varying communication networks with linear convergence rate," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 499–504, Apr. 2021.

[33] D. T. A. Nguyen, D. T. Nguyen, and A. Nedić, "Distributed Nash equilibrium seeking over time-varying directed communication networks," 2023, *arXiv:2201.02323*.

[34] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer, 2017.

[35] F. Parise, B. Gentile, and J. Lygeros, "A distributed algorithm for almost-Nash equilibria of average aggregative games with coupling constraints," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 2, pp. 770–782, Jun. 2020.

[36] D. Gadjov and L. Pavel, "Single-timescale distributed GNE seeking for aggregative games over networks via forward-backward operator splitting," *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3259–3266, Jul. 2021.

[37] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 1, pp. 438–449, Mar. 2022.

[38] M. Charikar et al., "Approximation algorithms for directed Steiner problems," *J. Algorithms*, vol. 33, pp. 73–91, 1999.

[39] T. Erseghe, "A distributed and scalable processing method based upon ADMM," *IEEE Signal Process. Lett.*, vol. 19, no. 9, pp. 563–566, Sep. 2012.

[40] 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.