Distributed Constrained Online Learning

Santiago Paternain, Soomin Lee, Michael M. Zavlanos, and Alejandro Ribeiro

***Abstract*—In this article, we consider groups of agents in a network that select actions in order to satisfy a set of constraints that vary arbitrarily over time and minimize a time varying function of which they have only local observations. The selection of actions, also called a strategy, is causal and decentralized, i.e., the dynamical system that determines the actions of a given agent depends only on the constraints at the current time and on its own actions and those of its neighbors. To determine such a strategy, we propose a decentralized saddle point algorithm and show that the corresponding global fit and regret are bounded by functions of the order of $\sqrt{T}$ i.e., functions whose limit is a constant when divided by $\sqrt{T}$. Specifically, we define the global fit of a strategy as a vector that integrates over time the global constraint violations as seen by a given node. The fit is a performance loss associated with online operation as opposed to offline clairvoyant operation which can always select an action, if one exists, that satisfies the constraints at all times. If this fit grows sublinearly with the time horizon it suggests that the strategy approaches the feasible set of actions. Likewise, we define the regret of a strategy as the difference between its accumulated cost and that of the best fixed action that one could select knowing beforehand the time evolution of the objective function. Numerical examples support the theoretical conclusions.**

***Index Terms*—Distributed algorithms, gradient methods, learning (artificial intelligence).**

I. Introduction

DISTRIBUTED optimization has applications in several engineering problems such as source localization [1], resource allocation problems in multi cellular communication networks [2], machine learning [3], [4], multi-robot teams [5]–[7] and the internet of things [8], [9]. In these problems agents try to optimize collectively a common objective function that is separable in local objectives. In cases where a centralized solution of such problems is acceptable and the objectives and constraints are convex, the problem reduces to solve a classic convex optimization problem. Several methods to do so exist, notably the saddle point algorithm by Arrow and Hurwicz [10], which has the advantage of admitting a distributed implementation [11]–[14]. In this framework the most studied classes of problems are when the constraints and the objective function are constant with respect of the time and when their variation is according to a stationary probability distribution. Solutions to the former problem have been established [11]–[14]. In the latter, the problem that is studied is that of selecting actions that minimize the expectation of the objective while satisfying the constraints in average. When the problem is unconstrained, centralized and decentralized implementations of stochastic gradient descent converge to such solutions [15]–[18].

In this paper we consider online formulations in which the cost and constraints can vary arbitrarily over time, even strategically. In this case, cost minimization can be formulated in the language of regret [19]–[21] whereby agents select online actions that result in a cost chosen by nature. The cost functions are revealed to the agents after the action are selected and these values are used to adapt the future strategy. Regret is defined as the difference between the total cost incurred by each agent and the cost of the optimal fixed solution that a clairvoyant agent could select. In that sense, it measures the effect of not knowing the temporal evolution of the cost function and, therefore, it can be used as a performance measure for online operation. Likewise, the fit of a strategy [22], [23] is a vector that integrates over time the constraint violation incurred by each agent. In that sense the fit is a performance loss associated with online operation as opposed to offline clairvoyant operation which can always select an action that satisfies the constraints at all times, if such action exists. It is a remarkable fact that online versions of the Arrow-Hurwicz algorithm for centralized problems achieve regret bounded by a constant and fit bounded by a function that grows at a sublinear rate [22]–[24]; this suggests vanishing per play penalties and constraint violation of online plays with respect to clairvoyant agent. Since the fit compares the accumulation of the constraints it is possible to achieve small fit by alternating periods in which the constraints are satisfied with slack and periods in which the constraints are violated. This notion is appropriated for constraints that have a cumulative nature. However, when this is not the case it is possible to work with the notion of saturated fit, where only violations of the constraints are accumulated. A trajectory with small saturated fit is one in which the constraints are violated by a significant amount only for a short period of time.

The problem of distributed online constrained convex optimization in continuous time has been studied in [25]–[27]. Continuous time approaches have advantages in the context of distributed control systems whenever signals are inherently continuous. Moreover, discrete-time approaches can be characterized as the discretization of continuous-time dynamics.

Manuscript received March 18, 2019; revised December 21, 2019; accepted May 26, 2020. Date of publication June 5, 2020; date of current version June 18, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Elias Aboutanios. This work is supported in part by ARL DCIST CRA W911NF-17-2-0181, AFOSR under award #FA9550-19-1-0169 and by NSF under award CNS-1932011. *(Corresponding author: Santiago Paternain.)*

Santiago Paternain and Alejandro Ribeiro are with the Department of Electrical and System Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: spater@seas.upenn.edu; aribeiro@seas.upenn.edu).

Soomin Lee is with the Yahoo! Research Verizon Media, Sunnyvale, CA 94089 USA (e-mail: soominl@yahoo-inc.com).

Michael M. Zavlanos is with the Department of Mechanical Engeeniring and Material Science, Duke University, Durham, NC 27708 USA (e-mail: michael.zavlanos@duke.edu).

Digital Object Identifier 10.1109/TSP.2020.2999671

The work in [25] considers an unconstrained problem where agents minimize a time-varying convex loss. Problems with constraints, also the subject of this work, have been considered in [26], [27] where Saddle Point algorithms are shown to achieve sublinear bounds on the network disagreement and on the regret achieved by the strategy. This work extends the results in [26] by considering time-varying constraint. Likewise, instead of imposing exact consensus as in [26], [27], we allow for small disagreement. This idea has been used for unconstrained problems in [28]. With this modification we are able to establish that the trajectories that arise from the distributed online saddle point dynamics are such that the disagreement of agents, the regret and the fit are bounded by sublinear functions of the time horizon (Section IV). This suggests that our proposed algorithm achieves consensus, feasibility and optimality as the time goes to infinity. The feasibility bounds presented in this work are also an extension with respect to [26], [27]. Central to the development of this result are the definitions of global fit and regret (Section II). The former is defined as a vector that contains the time integrals of the constraints of all the agents evaluated across the trajectory of a specific agent. Having small global fit means that the agent is able to satisfy the constraints of every other agent, and thus if it is placed in a different position in the network its performance is maintained. Likewise, the global regret is the evaluation of the accumulated global cost of a specific agent's trajectory with respected to the optimal centralized clairvoyant solution.

To solve the constrained online optimization problem under consideration, we propose an online distributed saddle point algorithm to control the growth of the global fit and regret. Saddle point algorithms update the primal variables – the actions – along the negative of a weighted linear combination of the gradients of each constraints. Since the feasible set of actions is typically represented by the intersection of the sub level sets of the constraints functions, this linear combination pushes the actions towards feasibility. The weights of this linear combination are updated according to the current violation of the constraints. If an action violates a constraint by much, its corresponding weight is increased faster, whereas if a constraint is satisfied its corresponding weight is reduced. We start by showing that if an action that is feasible for all agents and for all times exists, then the network disagreement is bounded by a function that is sublinear with respect to the time horizon. Based on this result we establish sublinear global fit and global regret. We also illustrate our algorithm on a problem involving a team of robots driving through an urban environment to perform real-time texture classification for the purpose of mapping and object recognition. We show that the team of robots succeeds in training a common classifier that allows them to distinguish between grass and pavement images even when some of the agents have only observed one of the classes. The rest of this paper is organized as follows. In Section II we formalize the online distributed optimization problem. Later we present the Distributed Online Saddle Point Algorithm (Section III) and we establish that it achieves consensus, feasibility and optimality in Section IV. Other than concluding remarks the paper finishes with numerical examples in section V and VI.

II. Constrained Online Learning in Networks

We consider a group of N agents linked by an undirected connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{1, \dots, N\}$ is a set of nodes and $\mathcal{E}$ is a set of edges so that $(i,j) \in \mathcal{E}$ means that i and j are connected to each other. The set $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$ contains all nodes that are connected to i and is called the neighborhood of i. Note that since the graph is undirected, node j is in the neighborhood of i if and only if node i is in the neighborhood of j.

We are interested in situations where the agents in $\mathcal{G}$ have access to arbitrarily time varying local constraints and local objective functions and continuously select actions that are good not only for their local constraints and costs but for the local constraints and costs of other agents. To explain this formally, let $t \in \mathbb{R}^+$ be a continuous time index, and $\mathcal{X} \subset \mathbb{R}^n$ a convex set, $f_i(t,\cdot) : \mathcal{X} \to \mathbb{R}^{m_i}$ be a set of m_i convex constraints at agent i and $f_{0i}(t,\cdot) : \mathcal{X} \to \mathbb{R}$ be a local convex cost incurred at node i. A *local* goal of node i is to select an action $\mathbf{x}_i \in \mathcal{X}$ that satisfies local constraints $f_i(t,\mathbf{x}_i) \preceq 0$ across a time interval $[0,T]$ while minimizing the cost $f_{0i}(t,\mathbf{x}_i)$ integrated over the same interval. This is tantamount to defining the locally optimal solution $\mathbf{x}_i^\ell$ over the interval $[0,T]$ as

$$\mathbf{x}_i^\ell := \underset{\mathbf{x}_i \in \mathcal{X}}{\operatorname{argmin}} \quad \int_0^T f_{0i}(t,\mathbf{x}_i)\, dt \qquad \text{s.t.} \quad f_i(t,\mathbf{x}_i) \preceq 0,\ \forall\, t \in [0,T]. \tag{1}$$

Problem (1) models situations in which each of the agents is acting independently since the optimal action of i depends on its local cost and constraints, and not on those of other agents. Instead, here we are interested in situations where the actions of agents are coordinated, so that an action $\mathbf{x}_i$ of agent i can affect the costs and constraints of other agents. This results in a *global* formulation in which the optimal action of each agent $\mathbf{x}_i^\star$ is defined as the one that satisfies the constraints of all agents and minimizes the integral of the sum cost,

$$\mathbf{x}_i^\star := \underset{\mathbf{x}_i \in \mathcal{X}}{\operatorname{argmin}} \quad \int_0^T \sum_{j=1}^N f_{0j}(t,\mathbf{x}_i)\, dt, \qquad \text{s.t.} \quad f_j(t,\mathbf{x}_i) \preceq 0, \quad \forall\, j \text{ and } t \in [0,T]. \tag{2}$$

We say that the problem in (2) is global because the action $\mathbf{x}_i$ is evaluated at the constraint and costs of all nodes. This readily implies that $\mathbf{x}_i^\star = \mathbf{x}_j^\star$ and that there is a single global action that is optimal for all nodes. For future reference we define the sum cost function $f_0(t,\mathbf{x}_i) := \sum_{j=1}^N f_{0j}(t,\mathbf{x}_i)$ and the aggregate constraint $f(t,\mathbf{x}_i) := [f_1(t,\mathbf{x}_i); \dots; f_N(t,\mathbf{x}_i)]$. With this notation, the problem in (2) can be rewritten as

$$\mathbf{x}_i^\star = \underset{\mathbf{x}_i \in \mathcal{X}}{\operatorname{argmin}} \quad \int_0^T f_0(t,\mathbf{x}_i)\, dt, \qquad \text{s.t.} \quad f(t,\mathbf{x}_i) \preceq 0, \quad \forall\, t \in [0,T]. \tag{3}$$

While (1) models agents acting in isolation, (2) and (3) model agents acting in concert. The latter situation arises when local functions are related to a common variable. E.g., the costs and constraints can represent local observations of a parameter to

be estimated [29] or local observations and costs of a plant to be controlled [30]. Problems having the form of (3) also arise in large scale optimization where costs and constraints are *not* acquired locally but are distributed over several servers to reduce computation and storage [31].

If the functions $f_{0i}(t,\cdot)$ and $f_i(t,\cdot)$ are available for all times $t\in[0,T]$, solving (3) reduces to solving a distributed convex optimization problem for which a number of standard algorithms are applicable; see e.g., [32], [33]. In this paper we consider problems in which the constraints $f_i(t,\cdot)$ and costs $f_0(t,\cdot)$ are arbitrary and observed causally and locally by node i. In this setting it makes sense to consider time varying strategies $\mathbf{x}_i(t)$ that adapt the action of agent i to the information that is revealed at time t. In this context the optimal argument in (3) is a clairvoyant action that would be chosen when agents have knowledge of the future evolution of the system at time $t=0$. The appropriate figures of merit in this case are the notions of regret [20], [34], [35] and fit [22] that we generalize to network settings in the following section. These quantities compare the performance of the online distributed operation with the offline centralized solution of (3).

Before proceeding with definitions of network regret and fit, notice that for the definition in (3) to be valid the function $f_0(t,\mathbf{x})$ has to be integrable with respect to the time variable t. In subsequent definitions and analyses we further require the network to be connected and the constraints f_i for all $i\in\mathcal{V}$, to be integrable, convex and Lipschitz continuous with respect to $\mathbf{x}$ for all times t. We formally state these assumptions next.

Assumption 1: The network is connected with diameter D, i.e., the shortest distance between the two most distant nodes in the network is D.

Assumption 2: Let $\mathcal{X}$ be a compact convex set and the functions $f_{0i}(t,\mathbf{x})$ and $f_i(t,\mathbf{x})$ be integrable with respect to t and convex with respect to $\mathbf{x}\in\mathcal{X}$ for all $t\in[0,T]$. We further assume that cost and constraints are Lipschitz continuous over $\mathcal{X}$ with respective constants $L_0>0$ and $L_f>0$. I.e., for any $\mathbf{x},\mathbf{y}\in\mathcal{X}$ and all $t\in[0,T]$ the cost functions satisfy

$$\big|f_{0i}(t,\mathbf{x})-f_{0i}(t,\mathbf{y})\big|\leq L_0\big\|\mathbf{x}-\mathbf{y}\big\|, \tag{4}$$

and the constraint functions satisfy

$$\big|f_{kj}(t,\mathbf{x})-f_{kj}(t,\mathbf{y})\big|\leq L_f\big\|\mathbf{x}-\mathbf{y}\big\|, \tag{5}$$

where $f_{kj}(t,\cdot)$ denotes the jth component of the vector valued constraint function $f_k(t,\cdot)$.

We remark that integrability with respect to t is a weak condition. We do not require differentiability, not even continuity. This entails a fundamental difference with time varying optimization problems that strive to track a time varying optimal argument under the assumption of smooth time varying costs and constraints [29], [36], [37]. The goal here is to design an algorithm that can adapt to unexpected changes in the system, including, indeed, most importantly, to those that arise because of discontinuities in the cost and constraint functions. Likewise, note that we require the functions to be Lipschitz—essentially gradient bounded—on the compact set $\mathcal{X}$. This assumption means therefore continuously differentiable almost everywhere in $\mathcal{X}$. Another requirement for $\mathbf{x}_i^\star$ to be well defined is existence of an action $\mathbf{x}^\dagger\in\mathcal{X}$ that satisfies the constraints at all times and for all nodes as we formally state next.

Assumption 3: There exists an action $\mathbf{x}^\dagger\in\mathcal{X}$ that satisfies the constraints of all agents for all times $t\in[0,T]$,

$$f_i(t,\mathbf{x}^\dagger)\prec 0,\qquad \forall\, i \text{ and } t\in[0,T]. \tag{6}$$

We say that $\mathcal{X}^\dagger:=\{\mathbf{x}^\dagger\in\mathcal{X}: f_i(t,\mathbf{x}^\dagger)\preceq 0,\forall i \text{ and } t\in[0,T].\}$ is the set of feasible actions.

We require as well that the minimum of the objective function does not become progressively smaller with time so that a uniform bound K holds for all times $t\in[0,T]$. The existence of the bound in (7) is a mild requirement. Since the function $f_0(t,\mathbf{x})$ is convex, for any time t it is lower bounded for compact set of actions $\mathcal{X}$. The only restriction imposed is that $\min_{\mathbf{x}\in\mathcal{X}^N} f_0(t,\mathbf{x})$ does not become progressively smaller with time so that a uniform bound K holds for all times $t\in[0,T]$.

Assumption 4: There exists $K>0$ independent of the time horizon T such that for all $t\in[0,T]$ it holds that

$$f_0(t,\mathbf{x}^\star)-\min_{\mathbf{x}\in\mathcal{X}^N} f_0(t,\mathbf{x})\leq K, \tag{7}$$

where $\mathbf{x}^\star$ is the solution to the problem (3).

A. Network Regret and Network Fit

To evaluate the cost performance of such trajectories we define the notions of network regret and network fit. Begin then by considering a trajectory $\mathbf{x}_i(t)$ chosen by agent i and the total accumulated cost $\int_0^T f_{0j}(t,\mathbf{x}_i(t))dt$ that this trajectory incurs for a possibly different agent j. We define the regret $\mathcal{R}^i_{Tj}$ as the difference of this accumulated cost relative to the corresponding cost that would be incurred by the optimal trajectory $\mathbf{x}_i^\star$ of (3)

$$\mathcal{R}^i_{Tj}:=\int_0^T f_{0j}(t,\mathbf{x}_i(t))\,dt-\int_0^T f_{0j}(t,\mathbf{x}_i^\star)\,dt. \tag{8}$$

Likewise, we consider the accumulation $\int_0^T f_j(t,\mathbf{x}_i(t))dt$ of constraint of agent j incurred by the trajectory of agent i. The fit $\mathcal{F}^i_{Tj}$ is defined as the comparison of this constraint accumulation relative to the corresponding constraint accumulation of the optimal trajectory $\mathbf{x}_i^\star$

$$\mathcal{F}^i_{Tj}:=\int_0^T f_j(t,\mathbf{x}_i(t))\,dt-\int_0^T f_j(t,\mathbf{x}_i^\star)\,dt. \tag{9}$$

The action $\mathbf{x}_i^\star$ can be considered as an offline reference that would be chosen by an entity that is clairvoyant, because it observes the future, and omniscient, because it observes the costs and constraints of all nodes. Our objective is to consider trajectories that are chosen online by agents that are causal, because they observe the past, and local, because they observe their local costs and exchange information with neighboring nodes only. In this context regret and fit can be interpreted as performance losses associated with online causal and local operation as opposed to offline clairvoyant and omniscient operation. If $\mathcal{F}^i_{Tj}$ is positive we are in a situation in which, had the constraints of all agents be known beforehand, we could have selected an action $\mathbf{x}^\dagger$ to satisfy all constraints. The fit measures how far the trajectory $\mathbf{x}(t)$ is from achieving that goal. Due to its cumulative nature, it

is possible to achieve small (local) fit by alternating between actions for which the constraints take positive and negative values. This is an appropriate model for quantities that can be stored – such as energy budgets enforced through average power constrains. However, in other settings this formulation can be a limitation. This drawback can be overcome by defining the saturated fit in which constraints slacks are saturated to a small negative constant. We discuss this in Section IV-A.

Analogously, if the regret $\mathcal{R}^i_{Tj}$ is large we are in a situation in which prior knowledge of the objective functions and constraints would had resulted in selecting a strategy $\mathbf{x}^\star$ that achieves much better performance than the one achieved by $\mathbf{x}_i(t)$. In that sense $\mathcal{R}^i_{Tj}$ indicates how much we regret not having had that information a priori.

A good learning strategy is one that achieves small regret and fit as that would be an indication that the trajectory $\mathbf{x}(t)$ approaches $\mathbf{x}^\star$. Notice however that since the objective function and the constraints are integrated over a time horizon T, it is natural to expect the cost and constraints to grow linearly with T. Thus, having regret and fit that grow at a sublinear rate is sufficient indication of a good learning strategy. This intuition motivates the following definitions of feasible and optimal trajectories.

Definition 1: We define an environment as a set of constraints $f_j : \mathbb{R} \times \mathcal{X} \to \mathbb{R}^{m_j}$ and costs $f_{0j} : \mathbb{R} \times \mathcal{X} \to \mathbb{R}$ for all $j \in \mathcal{V}$. For a trajectory $\mathbf{x}_i(t)$ we consider the regret and fit definitions in (8) and (9) and further define the sum regret $\mathcal{R}^i_T := \sum_{j\in\mathcal{V}} \mathcal{R}^i_{Tj}$ and the network wide fit $\mathcal{F}^j_T = [\mathcal{F}^{j\top}_{T1}, \dots, \mathcal{F}^{j\top}_{TN}]^\top$. We say that:

Feasibility: The trajectories are feasible in the environment if all the local fits $\mathcal{F}^i_T$ with $i \in \mathcal{V}$ grow sublinearly with T. I.e., there exist a function $h(T)$ with $\limsup_{T\to\infty} h(T)/T = 0$ and a constant vector C_f such that for all times T it holds,

$$\mathcal{F}^i_T := \int_0^T f(t, \mathbf{x}_i(t))\, dt \leq C_f h(T). \tag{10}$$

Optimality: The trajectories are optimal in the environment if all regrets $\mathcal{R}^i_T$ grow sublinearly for all $i \in \mathcal{V}$ and T. I.e. there exist a function $h(T)$ with $\limsup_{T\to\infty} h(T)/T = 0$ and a constant C such that for all times T it holds,

$$\mathcal{R}^i_T := \int_0^T f_0(t, \mathbf{x}_i(t))\, dt - \int_0^T f_0(t, \mathbf{x}^\star)\, dt \leq C h(T). \tag{11}$$

In the next section we develop the details of a distributed and online version of the Arrow-Hurwicz algorithm, such that its generated trajectories are feasible and optimal in the sense of Definition 1. The latter is formally stated and proved in Section IV along with an intermediate result that claims that the disagreement across agents is sublinear with respect to the time horizon, hence suggesting consensus. Before doing so, we make a pertinent remark highligthing the differences between the results in this work and those achieved for the centralized algorithm.

Remark 1: The centralized version of the problem here considered (3), can be solved – when affordable – by an online saddle-point algorithm [22]. The trajectories that arise from such algorithm achieve (i) regret bounded by a constant independent of the time horizon T for the unconstrained problem; (ii) fit bounded by a constant independent of the time horizon T if the objective function is constant with respect to the action, i.e., feasibility problems and (iii) regret bounded by a constant independent of the time horizon T and fit bounded by a sublinear function of T otherwise. By considering the distributed version of the previous algorithm we cannot establish regret or fit bounded by constants independent of T because of the disagreement across agents.

III. Distributed Online Saddle point

The problem defined in (3) is a centralized optimization problem in which all agents should select the same action. Since each agent $i \in \mathcal{V}$ has access only to the local cost and constraints, a more natural representation of the problem (3) is one where each agent selects a local decision vector $\mathbf{x}_i \in \mathbb{R}^n$. Nodes then try to achieve the minimum of their local objective functions $f_{0i}(t, \mathbf{x}_i)$ while satisfying the local constraints $f_i(t, \mathbf{x}_i) \preceq 0$ and keeping their variables equal to the variables $\mathbf{x}_j$ of neighboring nodes $j \in \mathcal{N}_i$. Defining $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top$, this alternative formulation can be written as

$$\begin{aligned} \mathbf{x}^\star := \underset{\mathbf{x}\in\mathcal{X}^N}{\operatorname{argmin}} \quad & \int_0^T f_0(t, \mathbf{x})\, dt \\ \text{s.t.} \quad & f_i(t, \mathbf{x}_i) \preceq 0, \forall t \in [0, T], \forall i \in \mathcal{V}, \\ & \mathbf{x}_i = \mathbf{x}_j, \forall i \in \mathcal{V}, j \in \mathcal{N}_i. \end{aligned} \tag{12}$$

Since the network is assumed to be connected (cf., Assumption 1), the constraints $\mathbf{x}_i = \mathbf{x}_j$ for all i and $j \in \mathcal{N}_i$ imply that (3) and (12) are equivalent. The previous problem can solved in a distributed manner with a variety of methods, one of which is he saddle point algorithm of Arrow and Hurwicz [10]. In this work we aim to extend the saddle point algorithm to control the growth of regret and fit. In doing so it is convenient to relax the consensus constraints $\mathbf{x}_i = \mathbf{x}_j$ in (12) to allow for some controlled disagreement. A common alternative to do so it is to add the disagreement to the objective function weighted by some positive real number. The main drawback of this approach is that, the relationship between the regularizer and the disagreement is not explicit. Instead, we accomplish this by defining the set of constraints

$$g_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \gamma \leq 0, \tag{13}$$

where γ is a positive constant limiting how much constraint violation is allowed. Notice that the parameter γ could be set to be arbitrarily small. The advantage of using a controlled disagreement is that it allows for agents to achieve a good global performance without damaging excessively the local performance, which in some applications might be important as well. By allowing larger values of γ, we allow more disagreement and therefore we prioritize the local performance, whereas by making γ closer to zero the goal is set in the centralized performance. This is, if γ is large enough, the coupling of the variables is given only by the constraints that each agent needs to satisfy. Whereas if γ is small we are closer to imposing the exact consensus constraint. The same relaxation is considered

in [28] in the case of unconstrained distributed optimization. The proximity constraints allows us to write the problem of interest in the following form

$$\begin{aligned}\tilde{\mathbf{x}}^\star := \underset{\mathbf{x}\in\mathcal{X}^N}{\operatorname{argmin}} \quad & \int_0^T f_0(t,\mathbf{x})\, dt \\ \text{s.t.} \quad & f_i(t,\mathbf{x}_i) \preceq 0, \forall t \in [0,T], \forall i \in \mathcal{V}, \\ & g_{ij}(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \gamma \leq 0, \forall i,j \in \mathcal{V}.\end{aligned} \tag{14}$$

For the above online optimization problem we can construct the following time varying Lagrangian

$$\mathcal{L}(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f_0(t,\mathbf{x}) + \sum_{i=1}^N \left(\boldsymbol{\lambda}_i^\top f_i(t,\mathbf{x}_i) + \boldsymbol{\mu}_i^\top g_i(\mathbf{x})\right), \tag{15}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}_+^{m_i}$ for $i = 1 \dots N$ and $\boldsymbol{\mu}_i \in \mathbb{R}_+^{|\mathcal{N}_i|}$ for $i = 1 \dots N$ are the Lagrange multipliers and where $g_i(\mathbf{x}) \in \mathbb{R}^{|\mathcal{N}_i|}$ is the vector with components $g_{ij}(\mathbf{x}_i,\mathbf{x}_j)$ for all $j \in \mathcal{N}_i$. Saddle point methods rely on the fact that for a constrained convex optimization problem, a pair is a primal-dual solution if and only if it is a saddle point of the Lagrangian associated with the problem, see e.g. [38]. This is the case in problem (14) since $f_0(\cdot,\mathbf{x})$, $f(\cdot,\mathbf{x})$ and $g(\cdot,\mathbf{x})$ are convex (c.f. Assumption 2). Notice that $\boldsymbol{\lambda}, \boldsymbol{\mu} \succeq 0$, hence the Lagrangian is convex with respect to $\mathbf{x}$ and therefore the subgradient with respect to $\mathbf{x}$ exists for all time $t \geq 0$, let us denote it by $\mathcal{L}_x(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})$. The Lagrangian is linear with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and therefore its partial derivatives with respect to these variables exist. Let us denote them by $\mathcal{L}_\lambda(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})$ and $\mathcal{L}_\mu(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})$ respectively. The actions $\mathbf{x}$ are updated – as in the classic Arrow-Hurwicz algorithm – by following the negative subgradient of the Lagrangian with respect to $\mathbf{x}$

$$\begin{aligned}\dot{\mathbf{x}} = & -\mathcal{L}_x(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = -f_{0,x}(t,\mathbf{x}) \\ & - \sum_{i=1}^N f_{i,x}(t,\mathbf{x}_i)^\top \boldsymbol{\lambda}_i - \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \mu_{ij} g_{ij,x}(\mathbf{x}),\end{aligned} \tag{16}$$

where $\mathcal{N}_i$ is the set of neighbors of node i. The primal update interprets the constraints as a potentials with corresponding weights $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and descends along a linear combination of the gradients of said potentials. The multipliers are then updated by following the subgradient of the Lagrangian with respect to them

$$\dot{\boldsymbol{\lambda}} = \mathcal{L}_\lambda(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\mathbf{x}), \tag{17a}$$

$$\dot{\boldsymbol{\mu}} = \mathcal{L}_\mu(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = g(\mathbf{x}). \tag{17b}$$

The intuition behind the latter update is that if a constraint is violated, for instance $f_{1,1}(\mathbf{x}) > 0$ the corresponding multiplier, $\boldsymbol{\lambda}_{1,1}$ will be increased, thus weighting more the corresponding gradient in the linear combination in (16). Which in turn pushes the action towards satisfying the constraint. On the other hand, if the constraint is satisfied, the weight of that potential will be reduced, thus making the direction of the corresponding gradient less important in the weighted linear combination. Observe that the multipliers need to remain positive at all time to ensure the convexity of the Lagrangian with respect to $\mathbf{x}$, yet if a multiplier takes the value zero and its corresponding constraint is satisfied, the previous update turns the multiplier negative. To avoid this issue, we will require a projection over the positive orthant. We formalize this idea next, after making the observation that the update (16)–(17) is indeed distributed. To see this, write the Lagrangian as a sum of the following local Lagrangians

$$\begin{aligned}\mathcal{L}^i(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = & f_0^i(t,\mathbf{x}_i) + \boldsymbol{\lambda}_i^\top f_i(t,\mathbf{x}_i) \\ & + \sum_{j\in\mathcal{N}_i} \mu_{ij}\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \gamma\right),\end{aligned} \tag{18}$$

where to compute each local Lagrangian, agent i needs only information regarding its variables and those of its neighbors. Then, each agent can compute locally the gradient of the Lagrangian with respect to its local variable $\mathbf{x}_i$ and perform the update described in (16)–(17), with the caveat that to ensure that the multipliers are always positive we need to consider a projected dynamical system. We formalize this idea next.

Definition 2 (Projection of a vector at a point): Let $\mathcal{K} \subset \mathbb{R}^n$ be a compact convex set. Then, for any $\mathbf{y} \in \mathcal{K}$ and $\mathbf{v} \in \mathbb{R}^n$, we defined the projection of $\mathbf{v}$ over the set K at the point $\mathbf{y}$ as

$$\Pi_{\mathcal{K}}[\mathbf{y},\mathbf{v}] = \lim_{\xi\to 0^+} \frac{P_{\mathcal{K}}(\mathbf{y}+\xi\mathbf{v}) - \mathbf{y}}{\xi}, \tag{19}$$

where the standard projection $P_{\mathcal{K}}(\mathbf{z}) = \operatorname{argmin}_{\mathbf{y}\in\mathcal{K}} \|\mathbf{y}-\mathbf{z}\|^2$ is always well defined because $\mathcal{K}$ is convex.

The intuition behind the projection is that, if the point $\mathbf{y}$ is in the interior of the set $\mathcal{K}$ then the projection of the vector $\mathbf{v}$ is the vector itself. In cases where $\mathbf{y}$ is in the boundary of the set $\mathcal{K}$, the projection of $\mathbf{v}$ is its component tangential to the boundary of $\mathcal{K}$. With this definition at hand, and by defining the gain of the controller to be $\varepsilon > 0$ we define the distributed online saddle point controller as follows. Each agent updates its action by following the negative subgradient of the Lagrangian with respect to its local copy of the action $\mathbf{x}_i$

$$\dot{\mathbf{x}}_i = \Pi_{\mathcal{X}}\left[\mathbf{x}_i, -\varepsilon\mathcal{L}_{x_i}(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})\right], \tag{20}$$

Likewise, the multipliers $\boldsymbol{\lambda}_i$ and $\boldsymbol{\mu}_i$ are updated by ascending along the direction of the gradient of the Lagrangian with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ respectively, i.e.,

$$\dot{\boldsymbol{\lambda}}_i = \Pi_+\left[\boldsymbol{\lambda}_i, \varepsilon\mathcal{L}_{\lambda_i}(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})\right], \tag{21a}$$

$$\dot{\mu}_{ij} = \Pi_+\left[\mu_{ij}, \varepsilon\mathcal{L}_{\mu_{ij}}(t,\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})\right]. \tag{21b}$$

The three gradients can be computed in a distributed fashion since they only depend on each agent's own variables and those of their neighbors. In [22] it was shown that in the centralized case, a saddle point algorithm akin to the one described by (20) and (21) achieves feasible and strongly optimal trajectories, i.e., fit bounded by a sublinear function of the time horizon and regret bounded by function that is constant with respect to the time horizon. In this work we show that the distributed version of the aforementioned algorithm (c.f. (20) and (21)) achieves feasible and optimal trajectories in the sense of Definition 1. Moreover, the network disagreement is bounded by a function

that is sublinear with respect to the time horizon. These results are the subject of the next section.

IV. Feasible and Optimal trajectories

Let us consider an energy-like function which will be used in subsequent analysis. Let $M = \sum_{i=1}^{N} m_i$ and $A = \sum_{i=1}^{N} |\mathcal{N}_i|$. Let $\tilde{\mathbf{x}} \in \mathcal{X}^N$, $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}_+^M$, $\tilde{\boldsymbol{\mu}} \in \mathbb{R}_+^A$, where we denote by $|\mathcal{N}_i|$ the carnality of the set of neighbors of node i, and define the function

$$V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = \frac{1}{2}\left(\|\mathbf{x}-\tilde{\mathbf{x}}\|^2 + \|\boldsymbol{\lambda}-\tilde{\boldsymbol{\lambda}}\|^2 + \|\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}}\|^2\right). \tag{22}$$

By considering the time derivative of the previous function along the dynamics (20)–(21) we establish that the integral of the difference of the Lagrangian evaluated at $(\mathbf{x}(t), \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})$ and the Lagrangian evaluated at $(\tilde{\mathbf{x}}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$ is bounded by a constant independent of the time horizon T. The following lemma – key to establish that the saddle point dynamics yield feasible and optimal trajectories – formalizes this result.

Lemma 1: Let Assumptions 1–3 hold. Then for any $T \geq 0$ the solutions of the dynamical system (20)–(21) satisfy

$$\int_0^T \mathcal{L}(t,\mathbf{x}(t),\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}) - \mathcal{L}(t,\tilde{\mathbf{x}},\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t))\,dt \leq \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}, \tag{23}$$

where $V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))$ is the energy-function defined in (22) with arbitrary $\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}$ and evaluated at the actions and multipliers at time zero.

Proof: See Appendix A. ∎

By analyzing the expression (23) for different choices of $\tilde{\mathbf{x}}$, $\tilde{\boldsymbol{\lambda}}$ and $\tilde{\boldsymbol{\mu}}$ it is possible to establish that the saddle point dynamics (20)–(21) yields sublinear network disagreement for all $T > 0$. We formalize this result in Proposition 1.

Proposition 1 (Sublinear Network Disagreement): Let Assumptions 1–4 hold. Then for any $T \geq 0$ the solutions of the dynamical system (20)–(21) are such that the network disagreement is sublinear with respect to T. In particular for $\boldsymbol{\lambda}(0) = 0$ and $\boldsymbol{\mu}(0) = 0$, for any $i, j \in \mathcal{V}$ we have that

$$\int_0^T \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|\,dt \leq D\sqrt{(K+\gamma)T + \frac{1}{2\varepsilon}\left(1 + \|\mathbf{x}^\star - \mathbf{x}(0)\|^2\right)}, \tag{24}$$

where D is the network diameter defined in Assumption 1.

Proof: Let us consider the expression (23) with $\tilde{\mathbf{x}} = \mathbf{x}^\star$, the solution of the problem (12), $\tilde{\boldsymbol{\lambda}} = \mathbf{0}$, and $\tilde{\boldsymbol{\mu}}_{ij} = 1$ for some $i \in \mathcal{V}$ and $j \in \mathcal{N}_i$ and $\tilde{\boldsymbol{\mu}}_{ik} = 0$ for all $k \neq j$ and $\tilde{\boldsymbol{\mu}}_l = \mathbf{0}$ for all $l \neq i$. For this selection of variables the Lagrangian evaluated at $(t, \mathbf{x}(t), \tilde{\lambda}, \tilde{\boldsymbol{\mu}})$ yields

$$\mathcal{L}(t,\mathbf{x}(t),\tilde{\lambda},\tilde{\boldsymbol{\mu}}) = f_0(\mathbf{x}(t)) + \left(\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2 - \gamma\right). \tag{25}$$

Applying Lemma 1 for this particular choice of $\tilde{\mathbf{x}}$, $\tilde{\lambda}$ and $\tilde{\boldsymbol{\mu}}$, (23) reduces to

$$\begin{aligned}\int_0^T f_0(\mathbf{x}(t)) + \left(\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2 - \gamma\right) dt \\ - \int_0^T \left(f_0(\mathbf{x}^\star) + \boldsymbol{\lambda}(t)^\top f(\mathbf{x}^\star)\right) dt \\ - \int_0^T \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \mu_{ij}(t)\left(\left\|\mathbf{x}_i^\star - \mathbf{x}_j^\star\right\|^2 - \gamma\right) dt \\ \leq \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}.\end{aligned} \tag{26}$$

Since $\mathbf{x}^\star$ is the solution to (12) it holds that $\mathbf{x}_i^\star = \mathbf{x}_j^\star$ and that $f(\mathbf{x}^\star) \preceq \mathbf{0}$. Since $\boldsymbol{\lambda}(t)$ and $\boldsymbol{\mu}(t)$ are always in the positive orthant, due to the projection in their update (c.f. (21a) and (21b)), we have that $\boldsymbol{\lambda}(t)^\top f(\mathbf{x}^\star) \leq 0$ and that $\boldsymbol{\mu}_{ij}(t)(\|\mathbf{x}_i^\star - \mathbf{x}_j^\star\|^2 - \gamma) = -\boldsymbol{\mu}_{ij}(t)\gamma \leq 0$ for all $t \in [0, T]$. These observations imply that

$$\int_0^T \left(f_0(\mathbf{x}(t)) - f_0(\mathbf{x}^\star) + \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2 - \gamma\right) dt \leq \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}. \tag{27}$$

From the definition of minimum and Assumption 4 it follows that

$$f_0(t,\mathbf{x}(t)) - f_0(t,\mathbf{x}^\star) \geq \min_{x\in\mathcal{X}^N} f_0(t,\mathbf{x}) - f_0(t,\mathbf{x}^\star) \geq -K. \tag{28}$$

Substituting (28) into (27) yields

$$\int_0^T -(K+\gamma) + \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2\,dt \leq \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}. \tag{29}$$

Rearranging the terms in the previous expression it holds that

$$\int_0^T \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2\,dt \leq (K+\gamma)T + \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}. \tag{30}$$

Because the square function is convex, by virtue of Jensen's inequality we have that

$$\left(\int_0^T \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|\,dt\right)^2 \leq \int_0^T \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2\,dt. \tag{31}$$

The previous inequality provides a lower bound for (30), hence we have that

$$\left(\int_0^T \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|\,dt\right)^2 \leq (K+\delta)T + \frac{V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))}{\varepsilon}. \tag{32}$$

By taking the square root of the previous inequality we observe that the disagreement among neighbors is sublinear. In particular, evaluating $V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0))$ with the particular selection of $\tilde{\mathbf{x}}$, $\tilde{\boldsymbol{\lambda}}$ and $\tilde{\boldsymbol{\mu}}$ yields

$$\int_0^T \|\mathbf{x}_i(t)-\mathbf{x}_j(t)\|\,dt \leq \sqrt{(K+\delta)T+\frac{1}{2\varepsilon}\left(1+\|\mathbf{x}^\star-\mathbf{x}(0)\|^2\right)}. \tag{33}$$

The latter establishes a sublinear disagreement among one hop neighbors on the network. To show that the result holds for every pair of nodes use the triangular inequality and the fact that the diameter of the network is D (Assumption 1). ■

The sublinear network disagreement that the previous proposition establishes, along with the result of Lemma 1, allows us to prove that the local Fit and Regret are bounded by sublinear functions with respect to the time horizon T. The latter means that the trajectories that arise from the Distributed Online Saddle Point Dynamics (20)–(21) are feasible and optimal in the sense of definition 1. We formalize these results in theorems 1 and 2 respectively.

Theorem 1 (Feasibility): Let Assumptions 1–4 hold. Then for any $T \geq 0$ the solutions of the dynamical system (20)–(21), with $\varepsilon > 1/2$, are such that the k-th component of the local fit $\mathcal{F}^i_{Tj}$ for any $i,j \in \mathcal{V}$ is bounded by $(\mathcal{F}^i_{Tj})_k \leq O(\sqrt{T})$.

Proof: We evaluate the expression (23) for the particular choice of $\tilde{\mathbf{x}} = \mathbf{x}^\star$ and $\tilde{\boldsymbol{\mu}} = 0$

$$\begin{aligned}\int_0^T f_0(t,\mathbf{x}(t))-f_0(t,\mathbf{x}^\star)dt &\times \sum_{i=1}^N \int_0^T \left[\tilde{\boldsymbol{\lambda}}_i^\top f_i(t,\mathbf{x}_i(t)) - \boldsymbol{\lambda}_i^\top(t) f_i(t,\mathbf{x}_i^\star)\right] dt \\ &- \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \int_0^T \mu_{ij}(t)\left(\|\mathbf{x}_i^\star-\mathbf{x}_j^\star\|^2-\gamma\right) dt \\ &\leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star,\tilde{\boldsymbol{\lambda}},\mathbf{0}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0)).\end{aligned} \tag{34}$$

By virtue of Assumption 4 and the bound derived in (28), we can lower bound the difference of the integrals of the objective functions by

$$\int_0^T f_0(t,\mathbf{x}(t))-f_0(t,\mathbf{x}^\star)dt \geq -\int_0^T K dt = -KT. \tag{35}$$

Substituting the previous bound in (34) yields

$$\begin{aligned}\sum_{i=1}^N \int_0^T \left[\tilde{\boldsymbol{\lambda}}_i^\top f_i(t,\mathbf{x}_i(t)) - \boldsymbol{\lambda}_i^\top(t) f_i(t,\mathbf{x}_i^\star)\right] dt &- \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \int_0^T \mu_{ij}(t)\left(\|\mathbf{x}_i^\star-\mathbf{x}_j^\star\|^2-\gamma\right) dt \\ &\leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star,\tilde{\boldsymbol{\lambda}},\mathbf{0}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0)) + KT.\end{aligned} \tag{36}$$

Observe that, since $\mathbf{x}^\star$ is the solution of the decentralized problem (12), we have that $f_i(t,\mathbf{x}_i^\star) \preceq \mathbf{0}$. Moreover, because the Lagrange Multipliers $\boldsymbol{\lambda}_i(t)$ are in the positive orthant (cf., (21a)) the product $-\boldsymbol{\lambda}_i^\top(t) f_i(t,\mathbf{x}_i^\star)$ is always positive. Likewise, the product $\boldsymbol{\mu}_{ij}(t)(\gamma-\|\mathbf{x}_i^\star-\mathbf{x}_j^\star\|^2) = \boldsymbol{\mu}_{ij}(t)\gamma \geq 0$, for all $t \geq 0$. With these considerations the following bound holds

$$\sum_{i=1}^N \int_0^T \tilde{\boldsymbol{\lambda}}_i^\top f_i(t,\mathbf{x}_i(t))\,dt \leq \frac{1}{\varepsilon} V_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0)) + KT. \tag{37}$$

Denote by $[\cdot]^+$ the projection over the positive orthant. Then, for a particular $i \in \mathcal{V}$ choose $\tilde{\boldsymbol{\lambda}}_i = [\mathcal{F}^i_{Ti}]^+$ and $\tilde{\boldsymbol{\lambda}}_j = \mathbf{0}$ for all $j \in \mathcal{V}$ such that $j \neq i$. Then, (37) reduces to

$$\left\|\left[\mathcal{F}^i_{Ti}\right]^+\right\|^2 \leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star,\tilde{\boldsymbol{\lambda}},\mathbf{0}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0)) + KT. \tag{38}$$

Without loss of generality assume $\boldsymbol{\lambda}(0) = 0$ and $\boldsymbol{\mu}(0) = 0$. Then, the right hand side of the previous expression reduces to

$$V_{\mathbf{x}^\star,\tilde{\boldsymbol{\lambda}},\mathbf{0}}(\mathbf{x}(0),\boldsymbol{\lambda}(0),\boldsymbol{\mu}(0)) = \frac{\|\mathbf{x}(0)-\mathbf{x}^\star\|^2 + \left\|\left[\mathcal{F}^i_{Ti}\right]^+\right\|^2}{2}, \tag{39}$$

and (38) can be written as

$$\left(1-\frac{1}{2\varepsilon}\right)\left\|\left[\mathcal{F}^i_{Ti}\right]^+\right\|^2 \leq \frac{1}{2\varepsilon}\left(\|\mathbf{x}(0)-\mathbf{x}^\star\|^2\right) + KT. \tag{40}$$

Then, for any $\varepsilon > 1/2$, the previous expression yields

$$\left\|\left[\mathcal{F}^i_{Ti}\right]^+\right\|^2 \leq \frac{\left(\|\mathbf{x}(0)-\mathbf{x}^\star\|^2 + 2\varepsilon KT\right)}{2\varepsilon-1}. \tag{41}$$

Taking the square root of both sides of the previous inequality shows that the norm of the projection of the fit is bounded by a sublinear function. In particular we have that each component $k = 1 \ldots m_i$ of $\mathcal{F}^i_{Ti}$ is also upper bounded by the a sublinear function that grows as $O(\sqrt{T})$. We are left to show that for any $j \neq i$ we have fit that is bounded by a function of the order of $\sqrt{T}$. The latter is a consequence of the Lipschitz continuity and the sublinear network disagreement as we show next. Add and subtract $f_{i,k}(t,\mathbf{x}_i(t))$ to the definition of the local fit to write

$$\begin{aligned}\left(\mathcal{F}^i_{Tj}\right)_k = \int_0^T f_{i,k}(t,\mathbf{x}_j(t))dt &= \int_0^T f_{i,k}(t,\mathbf{x}_i(t))dt \\ &+ \int_0^T f_{i,k}(t,\mathbf{x}_j(t)) - f_{i,k}(t,\mathbf{x}_i(t))dt.\end{aligned} \tag{42}$$

The first term on the right hand side of the previous expression is, by definition, $(\mathcal{F}^i_{Ti})_k$. This term is bounded by a function of the order as $\sqrt{T}$ as previously shown. On the other hand, the second integral can be bounded using the Lipschitz continuity of $f_{i,k}(t,\mathbf{x})$ (c.f. Assumption 2) by

$$\begin{aligned}\left|\int_0^T f_{i,k}(t,\mathbf{x}_j(t)) - f_{i,k}(t,\mathbf{x}_i(t))dt\right| &\leq \int_0^T L_f\|\mathbf{x}_i(t)-\mathbf{x}_j(t)\|dt\end{aligned} \tag{43}$$

Then, the result of Proposition 1 completes the proof. ■

The previous theorem establishes that the local fit achieved by a system that follows saddle point dynamics (20)–(21) is bounded by a function whose rate of growth is sublinear, thus suggesting vanishing penalties. However, as discussed previously this can be achieved by solutions that oscillate, i.e., trajectories that violate the constraints at some times and that satisfy them with slack at other periods. Since this is not desirable in some applications, we overcome this limitation by showing that the saturated global fit has the same property. We address this in Section IV-A. The fact that the fit grows sublinearly is equivalent to achieving trajectories that are feasible in the sense of Definition 1.

We next focus in establishing the optimality of the trajectories generated by the saddle point dynamics (20)–(21). In the next theorem we show that this is the case by proving that the growth of the regret is bounded by a sublinear function of the time horizon.

Theorem 2 (Optimality): Let Assumptions 1–4 hold. Then for any $T \geq 0$ the solutions of the dynamical system (20),(21a) and (21b) are such that the local regret $\mathcal{R}_T^i$ for any $i \in \mathcal{V}$ is bounded by a function of $O(\sqrt{T})$. In particular, for $\lambda(0) = 0$ and $\boldsymbol{\mu}(0) = 0$ we have that

$$\mathcal{R}_T^i \leq \frac{\left(1 + \|\mathbf{x}^\star - \mathbf{x}(0)\|^2\right)}{\varepsilon} + (N-1)L_0 D\sqrt{(K+\gamma)T + \frac{1}{2\varepsilon}\left(1 + \|\mathbf{x}^\star - \mathbf{x}(0)\|^2\right)} \tag{44}$$

Proof: Let us consider the local regret of agent j, with $j \in \mathcal{V}$, defined in (11)

$$\mathcal{R}_T^j = \int_0^T \sum_{i=1}^N \left(f_0^i(t, \mathbf{x}_j(t)) - f_0^i(t, \mathbf{x}_i^\star)\right) dt. \tag{45}$$

Add and subtract $\sum_{i=1, i\neq j}^N f_0^i(t, \mathbf{x}_i(t))$ to the previous equation and rewrite the previous expression as

$$\mathcal{R}_T^j = \int_0^T f_0(t, \mathbf{x}(t)) - f_0(t, \mathbf{x}^\star)\, dt + \int_0^T \sum_{i=1, i\neq j}^N f_0^i(t, \mathbf{x}_j(t)) - f_0^i(t, \mathbf{x}_i(t))\, dt. \tag{46}$$

From Lemma 1 and by choosing $\tilde{\mathbf{x}} = \mathbf{x}^\star$, $\tilde{\boldsymbol{\lambda}} = 0$, $\tilde{\boldsymbol{\mu}} = 0$ it follows that

$$\begin{aligned} &\int_0^T \sum_{i=1}^N \left(f_0^i(t, \mathbf{x}_i(t)) - f_0^i(t, \mathbf{x}_i^\star)\right) dt \\ &- \sum_{i=1}^N \int_0^T \boldsymbol{\lambda}_i^\top(t) f_i(t, \mathbf{x}_i^\star)\, dt \\ &- \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \int_0^T \boldsymbol{\mu}_{ij}(t)\left(\left\|\mathbf{x}_i^\star - \mathbf{x}_j^\star\right\|^2 - \gamma\right) dt \\ &\leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star, \mathbf{0}, \mathbf{0}}\left(\mathbf{x}(0), \boldsymbol{\lambda}(0), \boldsymbol{\mu}(0)\right). \end{aligned} \tag{47}$$

As it was previously argued, because $\mathbf{x}^\star$ is the solution of (12) it follows that $f_i(\mathbf{x}^\star) \preceq 0$ and that $\mathbf{x}_i^\star = \mathbf{x}_j^\star$. Combining these observations with the fact that the multipliers always lie in the positive orthant due to the projections introduced in their updates (21), we have that $-\lambda_i(t)^\top f_i(t, \mathbf{x}_i^\star) \geq 0$ and $\boldsymbol{\mu}_{ij}(t)(\|\mathbf{x}_i^\star - \mathbf{x}_j^\star\|^2 - \gamma) \geq 0$. Thus it follows that

$$\begin{aligned} &\int_0^T \sum_{i=1}^N f_0^i(t, \mathbf{x}_i(t)) - f_0^i(t, \mathbf{x}_i^\star)\, dt \\ &\leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star, \mathbf{0}, \mathbf{0}}\left(\mathbf{x}(0), \boldsymbol{\lambda}(0), \boldsymbol{\mu}(0)\right). \end{aligned} \tag{48}$$

Notice that the difference in the left hand side of the previous expression is equal to the first term in (46). Thus, the local regret in (46) can be upper bounded by

$$\mathcal{R}_T^j \leq \frac{1}{\varepsilon} V_{\mathbf{x}^\star, \mathbf{0}, \mathbf{0}}\left(\mathbf{x}(0), \boldsymbol{\lambda}(0), \boldsymbol{\mu}(0)\right) + \int_0^T \sum_{i=1, i\neq j}^N f_0^i(t, \mathbf{x}_j(t)) - f_0^i(t, \mathbf{x}_i(t))\, dt. \tag{49}$$

To bound the second term in the previous expression, use the Lipschitz continuity of the objective function (c.f. Assumption 2)

$$\left|\int_0^T \sum_{i=1, i\neq j}^N f_0^i(t, \mathbf{x}_j(t)) - f_0^i(t, \mathbf{x}_i(t))\, dt\right| \leq \sum_{i=1, i\neq j}^N \int_0^T L_0 \left\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\right\| dt. \tag{50}$$

The latter is bounded by a function of the order of $\sqrt{T}$ as a consequence of the sublinear network disagreement established in Proposition 1. Since $V_{\mathbf{x}^\star, \mathbf{0}, \mathbf{0}}(\mathbf{x}(0), \boldsymbol{\lambda}(0), \boldsymbol{\mu}(0))/\varepsilon$ is a constant, it holds that (49) is upper bounded by a function of the order of $\sqrt{T}$. To complete the proof of the bound in (44) it suffices to replace in the previous expression the network disagreement by the result of Proposition 1. ■

We have established that agents operating distributedly achieve sublinear network disagreement, fit and regret. In the next section we discuss the case where the constraint is lower bounded and therefore it cannot be satisfied with slack. This prevents the fit to be bounded due to trajectories that alternate between feasibility and large periods of infeasibility.

A. Saturated Fit

We define a saturated function to prevent the constraint to take negative values smaller than a given threshold. Formally, let $\delta > 0$ and define the function $\tilde{f}_\delta(t, \mathbf{x}) = \max\{f(t, \mathbf{x}), -\delta\}$. Then we can define the notion of saturated local fit as

$$\tilde{\mathcal{F}}_T^{ij} = \int_0^T \tilde{f}_{\delta, i}(t, \mathbf{x}_j(t))\, dt, \tag{51}$$

By taking small values of δ we can arbitrarily reduce the negative portion of the fit. Ideally one would like to set $\delta = 0$. We next establish that the sublinear bound for the fit established in Theorem 1 holds as well for the saturated fit.

Corollary 1: Let Assumptions 1–4 hold. Then for any $T \geq 0$ the solutions of the dynamical system (20)–(21), with $\varepsilon > 1/2$, are such that the k-th component of the saturated fit $\tilde{\mathcal{F}}^i_{\delta,Tj}$ for any $i, j \in \mathcal{V}$ is bounded by $(\tilde{\mathcal{F}}^i_{\delta,Tj})_k \leq O(\sqrt{T})$.

Proof: Recall that $\tilde{f}_{\delta,i}(t,\mathbf{x}) = \max\{f_i(t,\mathbf{x}), -\delta\}$. Because $\tilde{f}_{\delta,i}(t,\mathbf{x})$ is the point-wise maximum of two convex functions in $\mathbf{x}$ it is also convex. Thus, it satisfies the hypotheses of Theorem 1 and it follows that the local saturated fit is also bounded by a function of $O(\sqrt{T})$. ■

In the next section we present numerical examples that support the theoretical conclusions. Before doing so, we discuss the bound on the network disagreement established in Proposition 1.

Remark 2: In Proposition 1 we showed that the network disagreement depends on T as $O(\sqrt{T})$. However, the bound achieved is a direct consequence of the choice of the relaxation of the consensus constraint. This choice being arbitrary, it is possible to chose different relaxations to obtain different bounds on the network disagreement. If one desires to bound this quantity by a sublinear function $h(T)$, it suffices to impose the constraint $g(\|\mathbf{x}_i - \mathbf{x}_j\|) - \gamma < 0$ for any $g(y) = h^{-1}(y)$ as long as $g(y)$ is a convex function. The latter holds because the main component of the proof of the proposition is Jensen's inequality.

V. Numerical Examples: Robotic Team

In this section we consider a team of N robots tasked with classifying in real time and in a distributed manner the different objects and terrains that compose the environment in which they are deployed. This problem, has been studied in [5], although the method presented here is different. Each robot has only access to information about the environment based on the path it has traversed and the images gathered. Therefore, its local information may not be enough to achieve the task of classification since the information gathered may omit regions of the feature space that are crucial. See for instance Fig. 1 where we depict random trajectories of twenty agents driving around an intersection. When the agent is on the pavement i.e., the absolute value of its horizontal or vertical coordinate is less than five, then it observes pavement images. On the other hand, outside this region it observes grass. As it can be observed in that figure only some of the agents visit both regions and the interest is that the whole team can learn a common classifier. The advantage of learning such classifier is that a robot can identify if it is on grass even if it has not seen grass in the training process. In particular we consider a problem in which each robot receives features $\mathbf{z}_i(t) \in \mathbb{R}^n$ from the scene and corresponding labels $y_i(t) \in \{-1, 1\}$ depending on whether the terrain is grass or pavement. The details of the feature extraction from image data is provided in Section V-A. The common objective of the agents can be formulated as training a common linear classifier $\mathbf{x} \in \mathbb{R}^n$ that minimizes a loss function. The loss function is such that its value is small when the classification is accurate and it takes large values in the opposite case. In particular for this problem we consider logistic regression

$$f_i(t,\mathbf{x}) = \log\left(1 + e^{-y_i(t)\mathbf{x}^\top \mathbf{z}_i(t)}\right). \tag{52}$$

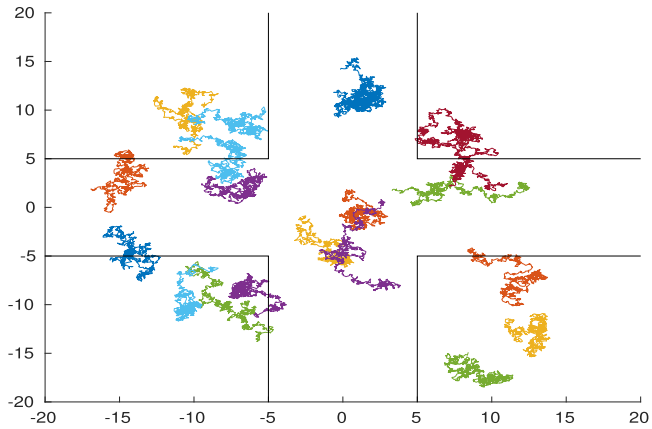

Fig. 1. Example of 20 robots driving randomly at an intersection. When the robot is on the street it is observing pavement images, whereas when it outside of the intersection it has access to grass images.

The classifier is designed so that the prediction is defined by the sign of the inner product between the classifier $\mathbf{x}$ and the feature vector $\mathbf{z}_i(t)$ observed by robot i at time t. This is, the predicted label is given by $\hat{y}_i(t) = \text{sign}(\mathbf{x}^\top \mathbf{z}_i(t))$. Notice that if the prediction is correct, both $\hat{y}_i(t)$ and $y_i(t)$ have the same sign and thus, the exponential in (52) takes a small value. Which in turn results in $f_i(t,\mathbf{x})$ being small. On the other hand, if the classification is incorrect, the sign of the exponential is positive, which results in a large value of $f_i(t,\mathbf{x})$. Hence, the expression in (52) is a surrogate of the error function since it results in small values when the prediction is correct and on large values otherwise. Notice that agents need to exchange their current actions $\mathbf{x}(t)$ with their neighbors to solve the minimization of (52) using the algorithm defined by (20) and (21). Since the dimensionality of the actions is as large as the feature vector we want to find a sparse classifier in order to reduce the communication cost. A way of doing so is to include a ℓ_1 norm regularization in the cost. This regularizer is known to promote sparsity. Let, $\alpha > 0$ and define the following local cost

$$\tilde{f}_i(t,\mathbf{x}) = f_i(t,\mathbf{x}) + \alpha \|\mathbf{x}\|_1 . \tag{53}$$

The previous objective introduces a trade-off between classification performance and sparsity. Instead, one can define a desired tolerance for the classification error – by imposing $f_i(t,\mathbf{x})$ to be smaller than a given tolerance $\delta > 0$ for all $i = 1 \dots N$– and by minimizing the objective $\|\mathbf{x}\|_1$, so to get the sparsest of the solutions. With this idea we define the following centralized problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & f_i(t,\mathbf{x}) - \delta \leq 0 \quad \forall i = 1 \dots N. \end{aligned} \tag{54}$$

To solve this problem in a distributed manner, we define – as done in Section II – local copies of the classifier $\mathbf{x}_i$ for each agent. The decentralized version of the previous problem then

yields

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\ldots\mathbf{x}_N} \quad \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i\|_1$$
$$\text{s.t.} \quad f_i(t,\mathbf{x}_i)-\delta\leq 0 \quad \forall i=1\ldots N$$
$$\|\mathbf{x}_i-\mathbf{x}_j\|^2-\gamma\leq 0 \quad \forall i=1\ldots N. \qquad (55)$$

We evaluate the performance of the saddle-point algorithm (20)–(21) by solving the problem (55) applied to the team of robots navigating around the intersection depicted in Fig. 1. The positions of the N agents is initialized by drawing it from a uniform distribution on the square $[-L,L]^2$ and their paths are random walks updated every T_s seconds, where each step is drawn from a two-dimensional Gaussian variable, with zero mean and covariance matrix $\text{diag}(\sigma_w,\sigma_w)$. Ever T_s seconds each agent has observed I images in the IRA[1] database [5] of either grass or pavement. Do notice that even though the algorithm proposed is derived in continuous time, for this application we propose to work with a discrete time system. In Section V-B we present the results achieved by the saddle-point algorithm in the previously described problem. Before doing so, we describe in the next section the feature extraction from the images.

A. Data from image database

The feature extraction is done as in [5], a procedure inspired in the two-dimensional texton [39]. We describe it next for completeness. The texture features $\mathbf{z}_i(t)$ are generated as the sum of a sparse dictionary representation of sub-patches of size 24-by-24. This is, each robot classifies images patches of size 24-by-24 by first extracting the nine non-overlapping 8-by-8 sub-patches within it. Each sub-patch is then vectorized, the sample mean subtracted off and divided by its norm. Such that the resulting sub-patch j observed by agent i, yields a zero-mean vector $\mathbf{z}_i^j$ with norm one. The 9 vectors resulting from each sub-patch are stacked as columns in a matrix $\mathbf{Z}_i=[\mathbf{z}_i^1;\ldots;\mathbf{z}_i^9]$. On the other hand, the agents have a dictionary of textures that has been trained offline following [5]. An example of this dictionary can be observed in Fig. 2. The dictionary can be represented by a matrix $\mathbf{D}\in\mathcal{M}^{n\times 64}$, where n is the number of features that one wants to extract. The feature used for classification by agent i is the aggregate sparse coding $\mathbf{z}_i(t)$, defined as $\mathbf{z}_i(t)=\sum_{j=1}^{9}\mathbf{z}^\star(\mathbf{D};\mathbf{z}_i^j(t))$, where $\mathbf{z}^\star(\mathbf{D};\mathbf{z}_i^j(t))$ is the solution to the following optimization problem.

$$\mathbf{z}^\star(\mathbf{D};\mathbf{z}_i^j(t))=\underset{\mathbf{z}}{\text{argmin}}\ \frac{1}{2}\|\mathbf{z}_i^j(t)-\mathbf{D}\mathbf{z}\|_2^2+\zeta\|\mathbf{z}\|_1, \qquad (56)$$

where $\zeta>0$ is the regularization coefficient.

B. Results

In this section we present the behavior of the Online Distributed Online Algorithm (20)–(21) for a team of robots that drive in the intersection as the one depicted in Fig. 1. For this particular example we consider $N=20$ agents, $L=15$, σ_w and $T_s=1$. The parameters of the feature extraction are set to $\zeta=0.125$, $n=128$. We chose $\delta=0.001$, $\gamma=10$ and the algorithm step size to be $\eta=0.02$, and we consider that each agent has access to 24 images per sampling period, in this case, 24 images per second.

[1]Integrated Research Assessment for the U.S. Army's Robotics Collaborative Technology Alliance.

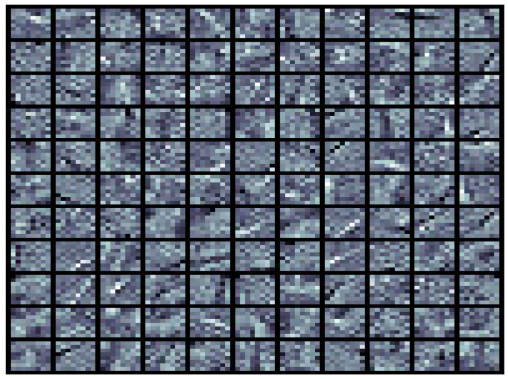

Fig. 2. Example of dictionary for 8-by-8 gray scale patches.

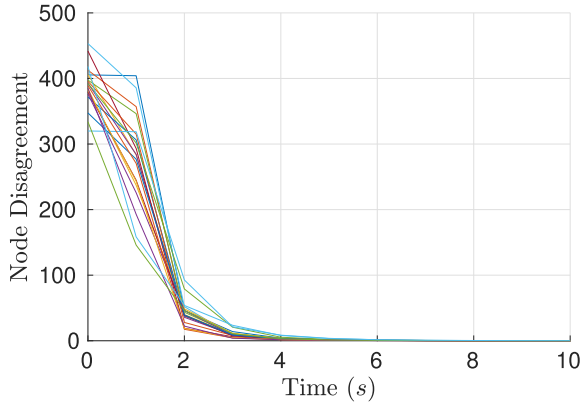

Fig. 3. Network disagreement per node for a network of 20 agents that follow the dynamics (20)–(21). The feature vectors $\mathbf{z}_i(t)\in\mathbb{R}^n$ are extracted from images of the IRA texture database as described in section V-A. The disagreement is sublinear as expected by virtue of Proposition 1.

As it can be observed in Fig. 3 the network disagreement converges to zero in approximately 6 seconds, which implies consensus among the agents. This observation supports the theoretical result in Proposition 1. In Fig. 4 we depict the network fit for one randomly selected node. As predicted by Theorem 1 the fit is sublinear. The effectiveness of the algorithm can be observed in the classification accuracy achieved by the agents in Fig. 5. Notice that the classification error of all the agents is bellow 30%. It can be observed as well, that some agents classify with accuracy above 90%. The latter is the case for agents that are observing grass. There seems to be an intrinsic difficulty in classifying pavement in the current data set. To support this claim we compute the covariance matrix of 512 features of images selected randomly. We then project the 192-dimensional feature vector onto the first two principal components. This projection is depicted in Fig. 6. As it can be observed the points corresponding to pavement cannot be separated from points corresponding to grass, yet there is a cluster of points corresponding to grass that is

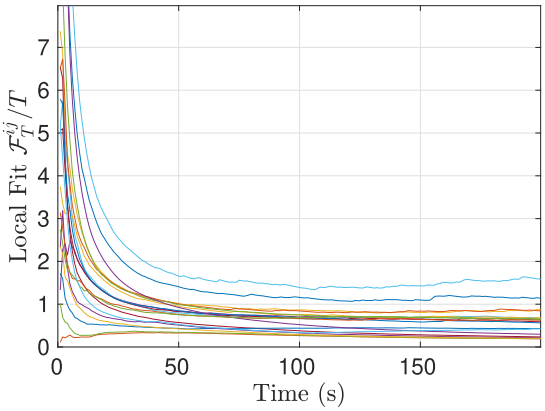

Fig. 4. Local fit $\mathcal{F}_T^j/T$ for a random node in a network of $N = 20$ agents that follow the dynamics (20)–(21). The feature vectors $\mathbf{z}_i(t) \in \mathbb{R}^n$ are extracted from images of the IRA texture database as described in Section V-A. As predicted by Theorem 1 the fit is sublinear.

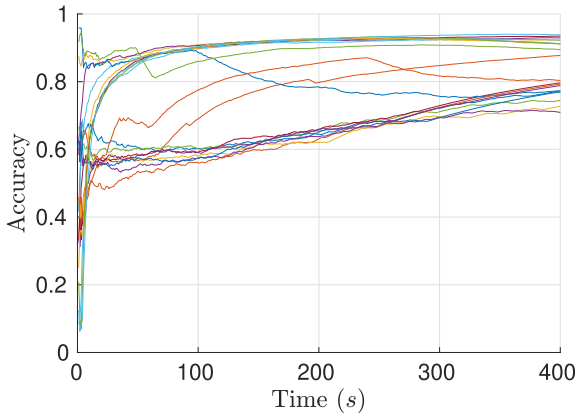

Fig. 5. The accuracy of the prediction per node reaches a minimum of 70% for a network of $N = 20$ agents that follow the dynamics (20)–(21). The feature vectors $\mathbf{z}_i(t) \in \mathbb{R}^n$ are extracted from images of the IRA texture database as described in Section V-A. As it can be observed some agents achieve accuracy of 90%. This agents are observing grass images, whereas those that perform worst are classifying pavement. The fact that one of the classes is poorly classified can be understood as not having a cluster of points where there is no grass as it can be seen in the study of the two principal components of the data set in Fig. 6.

spearated. This suggests that it is harder to classify the pavement images.

VI. Numerical Example: Synthetic Data

In this section we consider a random network with N agents with probability of connection p. The constraints are defined by a time-varying quadratic function

$$f_i(t, \mathbf{x}) = \mathbf{x}^\top Q_i^C \mathbf{x} - r_i(t)^2, \tag{57}$$

for all $i = 1, \dots, N$. The matrices $Q_i^C \in \mathbb{S}_+^{n \times n}$ are randomly selected and we choose $r_i(t) = (r_{Ti}(T-t) + r_{0i}t)/T$ where T is the horizon of the problem and r_{T_i}, r_{0i} are randomly drawn from a uniform distribution over the interval $[1, 2]$. The selection of the constraints guarantees that $0 \in \mathcal{X}^\dagger$, i.e., the origin is a strictly feasible solution for all times $t \in [0, T]$. The objective

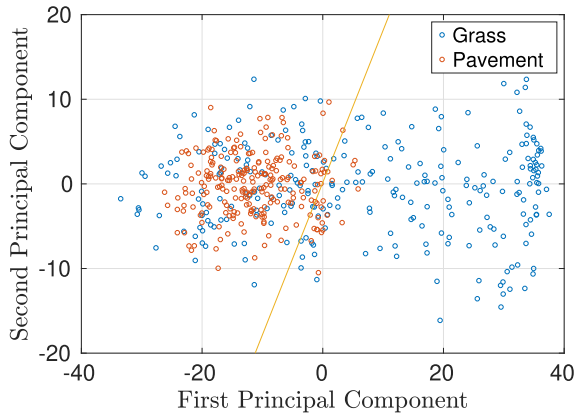

Fig. 6. We depict the projection of the features extracted from 512 images onto the two principal components of the data set. As it can be observed in this picture the images of grass are easier to classify since they present a distinct cluster without any pavement images. On the other hand, the cluster of points corresponding to pavement are intertwined with grass images, which makes its classification harder. We depict as well the projection of the classifier trained by node 1 after 400 seconds.

functions are quadratic functions of the following form

$$f_{0i}(t, \mathbf{x}) = (\mathbf{x} - \mathbf{c}_i(t))^\top Q_i (\mathbf{x} - \mathbf{c}_i(t)), \tag{58}$$

where $Q_i \in \mathbb{S}_+^{n \times n}$ are randomly selected for all $i = 1, \dots, N$ and $\mathbf{c}_i(t)$ satisfy

$$\mathbf{c}_i(t) = \mathbf{c}_{0i} cos(\theta_i + w_i t), \tag{59}$$

where θ_i are uniformly drawn over the interval [0,1]. Note that the clairvoyant solution can be computed since the objective to be minimized

$$f_{0T}(\mathbf{x}) = \int_0^T \sum_{i=1}^N (\mathbf{x} - \mathbf{c}_i(t))^\top Q_i (\mathbf{x} - \mathbf{c}_i(t)) \, dt \tag{60}$$

can be computed in closed form. Moreover, the constraints $f_i(t, \mathbf{x}) \leq 0$ for all $t \in [0, T]$ can be enforced by imposing the tightest of the constraints. This is,

$$\tilde{f}_i(\mathbf{x}) := \mathbf{x}^\top Q_i^C \mathbf{x} - \min\{r_{0i}, r_{Ti}\}^2 \leq 0. \tag{61}$$

Thus, the clairvoyant solution is computed by solving the problem

$$\begin{aligned} 2\mathbf{x}^\star \; := \; & \underset{\mathbf{x}_i \in \mathcal{X}}{\operatorname{argmin}} \quad f_{0T}(\mathbf{x}), \\ & \text{s.t.} \quad \tilde{f}_i(\mathbf{x}) \leq 0, \quad \forall\, i = 1, \dots, N. \end{aligned} \tag{62}$$

For the numerical example in this section we choose $N = 10, n = 1, p = 0.2, w_i = 0.01$ for all $i = 1 \dots N$ and $\mathbf{c}_{0i} = 1$ for all $i = 1 \dots N$. The network disagreement parameter is set to $\gamma = 0.1$.

Fig. 7 depicts the network disagreement in the setup previously described. As it can be observed, the disagreement vanishes which implies consensus among the agents. This observation supports the theoretical result of Proposition 1. In Fig. 8 we depict the network fit for one randomly selected node. The fit for said node is sublinear since when divided by the time horizon it converges to zero. This result is in accordance with

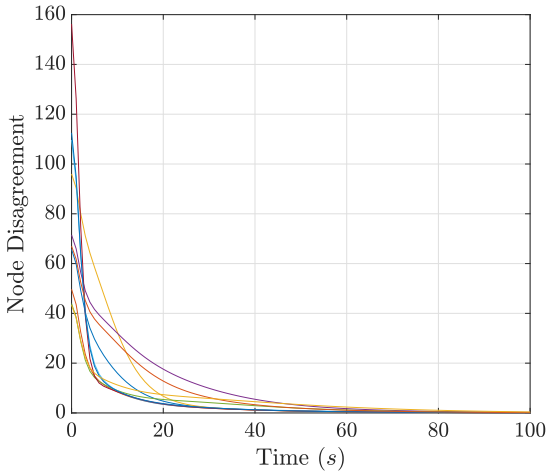

Fig. 7. Network disagreement per node for a network of 10 agents that follow the dynamics (20)–(21) for the problem described in section VI. The disagreement is sublinear as expected by virtue of Proposition 1.

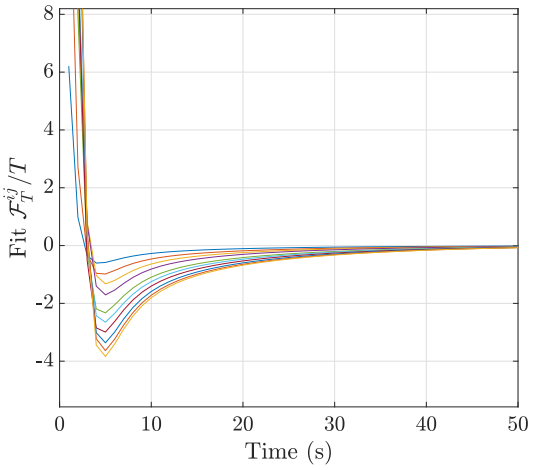

Fig. 8. Local fit $\mathcal{F}_T^j/T$ for a random node in a network of $N = 10$ agents that follow the dynamics (20)–(21) for the problem described in section VI. As predicted by Theorem 1 the fit is a sublinear function of the time-horizon.

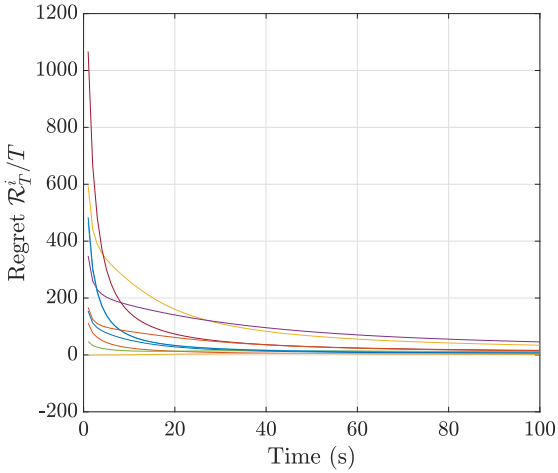

Fig. 9. Regret $\mathcal{F}_T^j/T$ for all the nodes in a network of $N = 10$ agents that follow the dynamics (20)–(21) for the problem described in section VI. As predicted by Theorem 1 the regret is a sublinear function of the time-horizon.

the theoretical result established in Theorem 1. Likewise, the regret for all nodes is sublinear as it can be observed in Fig. 9.

VII. Conclusion

We considered the problem of constrained distributed online learning. Each agent only has access to its local constraints and objective function, and the aim is to coordinate the actions among the agents such that the resulting trajectories are feasible and optimal for the team as a whole. We showed that a distributed online version of the saddle point algorithm achieves global fit, regret and network disagreement bounded by functions whose growth rate is bounded by $\sqrt{T}$. The latter result suggests vanishing constraint violation, optimality and network agreement in average as time evolves. We evaluate the performance of the algorithm for a team of robots driving through an urban environment to perform real time texture classification for the purpose of terrain recognition. Likewise, we evaluate the performance of said algorithm in a synthetic example.

Appendix

A. Proof of Lemma 1

Let us start by considering the time derivative of the energy function defined in (22). Using the chain rule yields

$$\dot{V}_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t)) = \sum_{i=1}^{N} \left(\mathbf{x}_i(t)-\tilde{\mathbf{x}}_i\right)^\top \dot{\mathbf{x}}_i(t) + \sum_{i=1}^{N} \left(\boldsymbol{\lambda}_i(t)-\tilde{\boldsymbol{\lambda}}_i\right)^\top \dot{\boldsymbol{\lambda}}_i(t) + \sum_{i=1}^{N} \left(\boldsymbol{\mu}_i(t)-\tilde{\boldsymbol{\mu}}_i\right)^\top \dot{\boldsymbol{\mu}}_i(t). \tag{63}$$

Substituting the time derivatives by those given by the Distributed Online Saddle Point dynamics (20)–(21) in the previous expression yields

$$\begin{aligned}
&\dot{V}_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\lambda}},\tilde{\boldsymbol{\mu}}}(\mathbf{x}(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t)) \\
&= \sum_{i=1}^{N} \left(\mathbf{x}_i(t)-\tilde{\mathbf{x}}_i\right)^\top \Pi_{\mathcal{X}}\left[\mathbf{x}_i(t), -\varepsilon \mathcal{L}_{x_i}(t,\mathbf{x}(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t))\right] \\
&+ \sum_{i=1}^{N} \left(\boldsymbol{\lambda}_i(t)-\tilde{\boldsymbol{\lambda}}_i\right)^\top \Pi_{+}\left[\boldsymbol{\lambda}_i(t), \varepsilon \mathcal{L}_{\lambda_i}(t,\mathbf{x}_i(t),\boldsymbol{\lambda}_i(t),\boldsymbol{\mu}_i(t))\right] \\
&+ \sum_{i=1}^{N} \left(\boldsymbol{\mu}_i(t)-\tilde{\boldsymbol{\mu}}_i\right)^\top \Pi_{+}\left[\boldsymbol{\mu}(t), \varepsilon \mathcal{L}^i_{\mu}(t,\mathbf{x}_i(t),\boldsymbol{\lambda}_i(t),\boldsymbol{\mu}_i(t))\right].
\end{aligned} \tag{64}$$

Since both $\mathbf{x}_i(t)$ and $\tilde{\mathbf{x}}_i$ belong to the convex set $\mathcal{X}$, the inner product between $\mathbf{x}_i(t)-\tilde{\mathbf{x}}_i$ and the projected vector can be upper bounded by the the product with the field without the projection (cf., Lemma 1 [22])

$$\begin{aligned}
&\left(\mathbf{x}_i(t)-\tilde{\mathbf{x}}_i\right)^\top \Pi_{\mathcal{X}}\left[\mathbf{x}_i(t), -\varepsilon \mathcal{L}_{x_i}(t,\mathbf{x}(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t))\right] \\
&\quad \leq -\left(\mathbf{x}_i(t)-\tilde{\mathbf{x}}_i\right)^\top \varepsilon \mathcal{L}_{x_i}(t,\mathbf{x}(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t)).
\end{aligned} \tag{65}$$

Summing across agents on both sides of the previous expression allows us to upper bound the first term summation in (64)

$$\sum_{i=1}^{N} \left(\mathbf{x}_i(t) - \tilde{\mathbf{x}}_i\right)^\top \Pi_{\mathcal{X}} \left[\mathbf{x}_i(t), -\varepsilon \mathcal{L}_{x_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$\leq -\varepsilon \sum_{i=1}^{N} \left(\mathbf{x}(t) - \tilde{\mathbf{x}}\right)^\top \mathcal{L}_x(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)). \tag{66}$$

In addition, since the Lagrangian is a convex function with respect to $\mathbf{x}_i(t)$ we can further upper bound the sum of inner products by the difference between the Lagrangian evaluated at $\tilde{\mathbf{x}}_i$ and $\mathbf{x}_i(t)$. Proceeding in this way for all $i \in \mathcal{V}$ yields

$$\sum_{i=1}^{N} \left(\mathbf{x}_i(t) - \tilde{\mathbf{x}}_i\right)^\top \Pi_{\mathcal{X}} \left[\mathbf{x}_i(t), -\varepsilon \mathcal{L}_{x_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$\leq \varepsilon \left(\mathcal{L}(t, \tilde{\mathbf{x}}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) - \mathcal{L}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right). \tag{67}$$

Likewise, for the multipliers the following relationships hold by virtue of [22, Lemma 1]

$$\left(\boldsymbol{\lambda}_i(t) - \tilde{\boldsymbol{\lambda}}_i\right)^\top \Pi_{+} \left[\boldsymbol{\lambda}_i(t), \varepsilon \mathcal{L}_{\lambda_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$\leq \left(\boldsymbol{\lambda}_i(t) - \tilde{\boldsymbol{\lambda}}_i\right)^\top \varepsilon \mathcal{L}_{\lambda_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)), \tag{68}$$

and

$$\left(\boldsymbol{\mu}_i(t) - \tilde{\boldsymbol{\mu}}_i\right)^\top \Pi_{+} \left[\boldsymbol{\mu}_i(t), \varepsilon \mathcal{L}_{\mu_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$\leq \left(\boldsymbol{\mu}_i(t) - \tilde{\boldsymbol{\mu}}_i\right)^\top \varepsilon \mathcal{L}_{\mu_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)). \tag{69}$$

Because the Lagrangian is linear with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ (cf., (15)) we can write the above inner products as differences of the Lagrangian evaluated at $\boldsymbol{\lambda}_i(t)$ and $\bar{\boldsymbol{\lambda}}_i$ and as differences of the Lagrangian evaluated at $\boldsymbol{\mu}_i(t)$ and $\bar{\boldsymbol{\mu}}_i$

$$\sum_{i=1}^{N} \left(\boldsymbol{\lambda}_i(t) - \tilde{\boldsymbol{\lambda}}_i\right)^\top \Pi_{+} \left[\boldsymbol{\lambda}_i(t), \varepsilon \mathcal{L}_{\lambda_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$+ \sum_{i=1}^{N} \left(\boldsymbol{\mu}_i(t) - \tilde{\boldsymbol{\mu}}_i\right)^\top \left[\boldsymbol{\mu}_i(t), \varepsilon \mathcal{L}_{\mu_i}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]$$
$$\leq \varepsilon \left(\mathcal{L}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) - \mathcal{L}(t, \mathbf{x}(t), \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})\right). \tag{70}$$

Substituting the expressions (67) and (70) in (64) reduces to

$$\dot{V}_{\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$$
$$\leq \varepsilon \left(\mathcal{L}(t, \tilde{\mathbf{x}}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) - \mathcal{L}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right)$$
$$+ \varepsilon \left(\mathcal{L}(t, \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) - \mathcal{L}(t, \mathbf{x}(t), \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})\right). \tag{71}$$

Observe that the second and third terms on the right hand side of the previous expression cancel out for all t. Hence, the previous bound reduces to

$$\dot{V}_{\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$$
$$\leq \varepsilon \left(\mathcal{L}(t, \tilde{\mathbf{x}}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) - \mathcal{L}(t, \mathbf{x}(t), \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}})\right). \tag{72}$$

Rearranging the terms in the previous inequality and integrating both sides from $t = 0$ until the time horizon $t = T$ yields

$$\int_0^T \mathcal{L}(t, \mathbf{x}(t), \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}) - \mathcal{L}(t, \tilde{\mathbf{x}}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\, dt$$
$$\leq -\frac{1}{\varepsilon} \int_0^T \dot{V}_{\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\, dt. \tag{73}$$

By virtue of the Fundamental Theorem of Calculus, the right hand side of the previous expression reduces to the difference between $V_{\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$ evaluated at time T and 0.

The proof if completed by observing that for any T, $V_{\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}}(\mathbf{x}(T), \boldsymbol{\lambda}(T), \boldsymbol{\mu}(T))$ is non-negative and thus, the right hand side can be upper bounded by the energy function evaluated at $t = 0$.

REFERENCES

[1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, ACM, 2004, pp. 20–27.

[2] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2988–3003, 2012.

[3] Z. John Lu, "The elements of statistical learning: Data mining, inference, and prediction," *J. Roy. Statistical Soc.: Ser. A (Statistics in Society)*, vol. 173, no. 3, pp. 693–694, 2010.

[4] R. L. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2762–2774, 2009.

[5] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4L: Decentralized dynamic discriminative dictionary learning," in *Proc. Intell. Robots Syst. IEEE/RSJ Int. Conf.*, 2015, pp. 2966–2973.

[6] A. Koppel, S. Paternain, C. Richard, and A. Ribeiro, "Decentralized online learning with kernels," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3240–3255, Jun. 2018.

[7] C. Freundlich, S. Lee, and M. M. Zavlanos, "Distributed active state estimation with user-specified accuracy," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 418–433, 2018.

[8] A. Ghosh and S. Sarkar, "Pricing for profit in Internet of Things," in *Inf. Theory (ISIT), IEEE Int. Symp.*, 2015, pp. 2211–2215.

[9] K. Gatsis and G. J. Pappas, "Wireless control for the IOT: Power, spectrum, and security challenges," in *Proc. Second Int. Conf. Internet-of-Things Des. Implementation*, 2017, pp. 341–342, ACM.

[10] K. J. Arrow and L. Hurwicz, *Studies in Linear and Nonlinear Programming*. Standford, CA, USA: Stanford Univ. Press, 1958.

[11] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, 2012.

[12] D. Yuan, S. Xu, and H. Zhao, "Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms," *IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics)*, vol. 41, no. 6, pp. 1715–1724, 2011.

[13] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[14] D. Feijer and F. Paganini, "Stability of primal–dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, no. 12, pp. 1974–1981, 2010.

[15] H. Robbins and S. Monro, "A stochastic approximation method," *The Ann. Math. Statist.*, 1951, pp. 400–407.

[16] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, no. 1-2, pp. 83–112, 2017.

[17] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 15, pp. 3924–3938, 2014.

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

[19] D. Blackwell *et al.*, "An analog of the minimax theorem for vector payoffs," *Pacific J. Math.*, vol. 6, no. 1, pp. 1–8, 1956.
[20] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 928–936.
[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., New York, NY, USA: Springer, 1995.
[22] S. Paternain and A. Ribeiro, "Online learning of feasible strategies in unknown environments," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2807–2822, 2017.
[23] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.
[24] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, no. Sep., pp. 2503–2528, 2012.
[25] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *Proc. Decis. Control (CDC), IEEE 52nd Annu. Conf.*, 2013, pp. 1484–1489.
[26] S. Lee and M. M. Zavlanos, "Distributed primal-dual methods for online constrained optimization," in *Proc. Amer. Control Conf., 2016*, pp. 7171–7176.
[27] S. Lee, A. Ribeiro, and M. M. Zavlanos, "Distributed continuous-time online optimization using saddle-point methods," in *Proc. Decis. Control (CDC), 2016 IEEE 55th Conf.*, 2016, pp. 4314–4319.
[28] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," in *Acoust., Speech Signal Process. (ICASSP), IEEE Int. Conf.*, 2016, pp. 3726–3730.
[29] F. Y. Jakubiec and A. Ribeiro, "D-MAP: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 450–466, 2013.
[30] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, 2010.
[31] R. D. Braun, "Collaborative optimization: An architecture for large-scale distributed design," 1996.
[32] 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, 2011.
[33] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, 2015.
[34] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.
[35] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2-3, pp. 169–192, 2007.
[36] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591, 2016.
[37] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1973–1986, 2018.
[38] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, U.K.: Cambridge Univ. Press, 2004.
[39] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Comput. Vision*, vol. 43, no. 1, pp. 29–44, 2001.

Santiago Paternain received the B.Sc. degree in electrical engineering from the Universidad de la Repblica Oriental del Uruguay, Montevideo, Uruguay in 2012, the M.Sc. degree in statistics from the Wharton School in 2018 and the Ph.D. degree in electrical and systems engineering from the Department of Electrical and Systems Engineering, the University of Pennsylvania in 2018. He is a Postdoctoral Researcher at the University of Pennsylvania. He was the recipient of the 2017 CDC Best Student Paper Award and the 2019 Joseph and Rosaline Wolfe Best Doctoral Dissertation Award from the Electrical and Systems Engineering Department at the University of Pennsylvania. His research interests include optimization and control of dynamical systems.

Soomin Lee received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, and two master's degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) and in computer science from the University of Illinois at Urbana-Champaign. After graduation, she worked as a Postdoc Associate in Duke Robotics Group and Industrial and Systems Engineering at Georgia Tech. She is a Senior Research Scientist at Yahoo! Research / Verizon Media. She is a recipient of NSF South Data Hub PEPI fellowship. Her research interests include theoretical optimization, control and optimization of distributed systems interconnected over complex networks, deep learning, and large-scale machine learning algorithms for big data analysis.

Michael M. Zavlanos (Senior Member, IEEE) received the diploma in mechanical engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2002, and the M.S.E. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, in 2005 and 2008, respectively.

He is currently an Associate Professor in the Department of Mechanical Engineering and Materials Science at Duke University, Durham, NC. He also holds a secondary appointment in the Department of Electrical and Computer Engineering and the Department of Computer Science. Prior to joining Duke University, Dr. Zavlanos was an Assistant Professor in the Department of Mechanical Engineering at the Stevens Institute of Technology, Hoboken, NJ, and a Postdoctoral Researcher in the GRASP Lab, University of Pennsylvania, Philadelphia, PA. His research focuses on control theory, optimization, and learning and, in particular, autonomous systems and robotics, networked and distributed control systems, and cyber-physical systems.

Dr. Zavlanos is a recipient of various awards including the 2014 Naval Research Young Investigator Program (YIP) Award and the 2011 National Science Foundation Faculty Early Career Development (CAREER) Award.

Alejandro Ribeiro received the B.Sc. degree in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis in 2005 and 2007. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D studies, in 2008 he joined the University of Pennsylvania (Penn), Philadelphia, where he is currently the Rosenbluth Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His focus is on structured representations of networked data structures, graph signal processing, network optimization, robot teams, and network control. Dr. Ribeiro received the 2014 O. Hugo Schuck Best Paper Award, and paper awards at the 2016 SSP Workshop, 2016 SAM Workshop, 2015 Asilomar SSC Conference, ACC 2013, ICASSP 2006, and ICASSP 2005. His teaching has been recognized with the 2017 Lindback award for distinguished teaching and the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching. Dr. Ribeiro is a Fulbright Scholar class of 2003 and a Penn Fellow class of 2015.