Asynchronous Online Learning in Multi-Agent Systems with Proximity Constraints

Amrit Singh Bedi*, *Student Member, IEEE,* Alec Koppel†, *Member, IEEE,* and Ketan Rajawat*, *Member, IEEE*

***Abstract*—We consider the problem distributed learning from sequential data via online convex optimization. A multi-agent system is considered where each agent has a private objective but is willing to cooperate in order to minimize the network cost, which is the sum of local cost functions. Different from the classical distributed settings, where the agents coordinate through the use of consensus constraints, we allow the neighboring agent actions to be related via a non-linear proximity function. A decentralized saddle point algorithm is proposed that is capable of handling gradient delays arising from communication or computational issues. The proposed online asynchronous algorithm is analyzed under adversarial settings by developing bounds on the regret of $\mathcal{O}(\sqrt{T})$, that measures the cumulative loss incurred by the online algorithm against a clairvoyant, and network discrepancy of $\mathcal{O}(T^{3/4})$, that measures the cumulative constraint violation or agent disagreement. By allowing the agents to utilize stale gradient information, the proposed algorithm embraces the nuances of distributed learning, and serves to be the first distributed online algorithm that can handle adversarial delays. A modified saddle point algorithm is also proposed that explicitly forces the agents to agree as per the constraint function resulting in zero network discrepancy, while incurring a slightly higher regret. To showcase the efficacy of the proposed asynchronous algorithm, a spatially correlated random field estimation problem is formulated and solved. Additionally, an application of vision based target localization with moving cameras demonstrates the benefits of this approach in practice.**

I. Introduction

The recent surge in multimedia usage coupled with the increase in the available cellular rates has led to the scarcity of certain resources such as battery charge and processor availability [2], [3]. Wireless devices, though capable of transmitting at high data rates, are increasingly focusing on optimal resource allocation. Multi-agent systems interested in carrying out coordinated learning tasks are required to not only utilize their power budget judiciously, but also refrain from communicating unnecessarily [4], [5]. Most multi-agent systems are also heterogeneous, and have disparate sleep schedules that do not allow continuous data exchange at high rates. The modern approach towards handling networks with such reticent and heterogeneous agents is to explicitly design distributed algorithms that can tolerate errors and delays while allowing the agents to 'fall behind' or 'catch up' intermittently.

This work considers the problem of distributed online learning with constraints. Keeping the vagaries of distributed operation in mind, the goal is to design asynchronous and flexible optimization algorithms that can learn from sequential data. We adopt the perspective of online convex optimization [6], where at each time slot, the learner selects an action (defining a parametric statistical model, for instance), and then observes a cost corresponding to that action. In the face of a non-stationary and possibly adversarial environment, we adopt a benchmark called regret, which is the difference between the cumulative costs incurred by the algorithm against that incurred by a clairvoyant that has access to all the cost functions up to a fixed time horizon T. An online algorithm is said to be no-regret if the regret grows sublinearly with T.

We consider a distributed setting wherein each agent $i \in \mathcal{V}$ in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ must act based on a locally observed sequentially available cost $f_{i,t}$, but seek to minimize the global network cost $f_t = \sum_i f_{i,t}$. The traditional approach is to assign a local action variable $\mathbf{x}_i$ for agent i but impose consensus constraints of the form $\mathbf{x}_i = \mathbf{x}_j$ for each $(i,j) \in \mathcal{E}$ [7], [8]. It is then possible to minimzize the network cost in a distributed fashion using frameworks such as alternating directions method of multipliers (ADMM) and distributed gradient descent (DGD) [8], [9]. Such formulations avoid the computational bottleneck at individual agents and leverage the parallel processing capabilities of distributed systems [10].

Complications arise when local variables $\mathbf{x}_i$ are associated with disctict priors and the edge relationships are non-linear. In such settings, forcing consensus across the edges may degrade the local predictive performance [11], [12]. For instance, within the context of distributed parameter estimation, if the observations across the network are independent but not identically distributed, consensus may yield a sub-optimal solution. Proximity instead of consensus constraints may better model such edge relationships where agent actions are required to be close but not exactly equal. Attempts to extend multi-agent optimization techniques to heterogeneously correlated problems have been considered in [13], [14] for special loss functions and correlation models. The generic problem was recently solved within the deterministic framework in [15].

State-of-the-art online learning algorithms capable of handling non-linear constraints in a distributed manner include the proximal point algorithms [16], [17]. The idea of handling non-linear constraints using proximal projection is viable only when the proximal operation is easy to perform. On the other hand, the present work allows a much broader class of convex constraint functions. Further, different from existing works that insist on synchronous operation, we put forth the first asynchronous distributed online algorithm that can handle non-linear constraints in adversarial settings. Asynchrony is

A. S. Bedi and K. Rajawat are with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: amritbd@iitk.ac.in; ketan@iitk.ac.in). A. Koppel is with U.S. Army Research Laboratory, Adelphi, MD, USA. (e-mail: alec.e.koppel.civ@mail.mil) This work was presented in part [1] at 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, USA and was among the finalists for the Best Student Paper Contest.

incorporated by allowing the agents to use stale information in the form of delayed gradients [18]. These delays are generally modeled as being random, e.g., arising from a local Poisson clock at each agent [19], or deterministic and bounded [20], as considered here. It is remarked that the dual asynchronous algorithm in [21] can also be used to solve the problem of interest here, but at a higher per-iteration cost. Specifically, the primal update in dual-ascent like methods entails solving the primal subproblem, as opposed to the simpler primal update step in saddle point methods. Asynchronous algorithms for unconstrained problems, such as those popular in the machine learning community [18], [22], are not directly applicable here due to the presence of constraints. An asynchronous algorithm for the constrained stochastic optimization problem is presented in [23] for non-adversarial settings. The algorithm proposed in [23] assumed independent loss functions at each time t, hence not applicable to tracking problems which are of interest in this paper. Additionally, the stationarity assumption in [23] allows the synchronous and asynchronous updates to be related to each other through the use of Lipschitz continuity of the objective function and its gradient.The same cannot be used in the present case as the objective function also depends on the time index t. Therefore, an entirely different approach is required where the asynchrony is handled right from the start of the analysis. In contrast, asynchronous alternating directions method of multipliers (ADMM) methods can handle convex constraints via a projection step, but have never been applied to online settings where the objective function is adversarial [24]–[27].

The major contributions of this work are stated as follows.

- In this paper, the algorithm proposed in [15], [28], [29] is extended to asynchronous settings for heterogeneous networks for for online multi-agent optimization with nonlinear network proximity constraints. The proposed algorithms advocates the use of delayed information in the form of stale gradient for primal and dual updates .
- The main technical contribution of the paper is to provide regret bounds both in terms of the global primal cost and constraint violation, which establish that the proposed approach belongs to the family of no-regret algorithms for this more challenging asynchronous setting.
- The proposed algorithm is then applied to the problem of estimating a spatially correlated random field in an asynchronous sensor network. In addition to that, efficacy of the proposed algorithm is demonstrated for an application of vision based target localization with moving cameras.

Different from [1], the detailed analysis for the Theorem 1 is presented in this work. Additionally, a modified saddle point algorithm is discussed that achieves zero network discrepancy with regret of $\mathcal{O}(T^{3/4})$ as summarized in Theorem 2. Further, we have detailed an application of vision based target tracking in this work to establish the usefulness of the proposed algorithm in practice. The paper is organized as follows. Problem formulation with an example is presented in Sec. II. Then, Sec.III describes the proposed asynchronous saddle point algorithm. Technical assumptions and regret bounds are provided in Sec.IV followed by the applications and numerical tests in Sec.V. In the end, Sec. VI concludes the paper.

Notations: All the scalars are denoted by regular font and vector in boldface. Capital boldface letters represents the matrix. All the vectors are column vectors unless otherwise stated. In notation $\mathbf{x} := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$, T denotes the transpose operation. Alternatively, in $\sum_{t=1}^{T}$, T represents the number of time slots over which summation is taken.

II. Problem Formulation

Consider a symmetric, connected, and directed network of N agents. Let $G = (\mathcal{V}, \mathcal{E})$ denote the graph of the network with $|\mathcal{V}| = N$ and $|\mathcal{E}| = M$ edges. Each agent i in the network seeks to develop a strategy for selecting its local sequence of action or decision variables $\mathbf{x}_{i,t} \in \mathcal{X}$ based on sequentially revealed convex cost functions $f_{i,u} : \mathcal{X} \to \mathbb{R}$ for $u \leq t$. Departing from the classical online learning framework, the setting considered here does not require the agents to operate on a common time scale. That is, agent i observes its local cost $f_{i,t-\tau_i(t)}(\cdot)$ with a delay of $0 \leq \tau_i(t) \leq \tau$. The quality of an individual learning rule is measured by the *delayed* local regret

$$\mathbf{Reg}_T^i = \sum_{t=1}^{T} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_{i,t-\tau_i(t)}(\mathbf{x}). \quad (1)$$

Stacking the decision variables $\mathbf{x} := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$ and all individual regrets, we obtain the *network delayed regret*

$$\mathbf{Reg}_T := \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\tilde{\mathbf{x}}_T^\star) \quad (2)$$

where $\tilde{\mathbf{x}}_T^* := \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x})$. Let us define $F_{t-\tau(t)}(\mathbf{x}) := \sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x})$ as a compact notation for the time varying objective function. Observe that an agent i may minimize its contribution to $\mathbf{Reg}_T$ in (2) independently of the other agents based solely upon its local sequence of costs $f_{i,t}$, i.e., it is possibly to solve (2) through N parallel local learning rules.

Numerous works on online multi-agent optimization operate on the assumption that in addition to making $\mathbf{Reg}_T$ small, they seek decision variables which coincide. This hypothesis may be implemented via consensus constraints, i.e.,

$$\mathbf{x}_{i,t} = \mathbf{x}_{j,t} \text{ for all } (i,j) \in \mathcal{E} \quad (3)$$

or by modifying the definition of regret to incorporate cross-agent dependencies, as in [9]. In applications where complex correlation structures or latent dependences among the agents' variables are present, the assumption of common estimate (considered in consensus) is clearly violated. In general, parameters of nearby agents may be close but not necessarily all equal, as is the situation in, e.g., the estimation of a smooth random field that is not uniform. We model this situation by introducing a convex local proximity function with real-valued range of the form $h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and a tolerance $\gamma_{ij} \geq 0$ to couple the decisions of agent i to that of its neighbors $j \in n_i$. The agents must cooperate to ensure that their individual delayed regrets $\mathbf{Reg}_T^i$ as well as their pairwise constraint violations $\left[\sum_{t=1}^{T} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)}) - \gamma_{ij}\right]_+$

grow sublinearly over time. Aggregating the pairwise constraint violations, define the *network discrepancy*

$$\mathrm{ND}_T := \sum_{(i,j)\in\mathcal{E}} \Big[\sum_{t=1}^{T} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_a tij(t)}) - \gamma_{ij}\Big]_+ \quad (4)$$

where $[\cdot]_+$ denotes projection onto the non-negatives, and the comparator in (2) is revised to be the constrained optimizer

$$\mathbf{x}_T^* = \operatorname*{argmin}_{\mathbf{x}} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x}_i) \quad (5)$$
$$\text{s. t. } h_{ij}(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma_{ij} \text{ for all } (i,j) \in \mathcal{E}.$$

The goal of online learning is asymptotic no-regret, i.e., $\lim_{T\to\infty} \mathbf{Reg}_T/T = 0$ as well as asymptotic constraint satisfaction, i.e., $\lim_{T\to\infty} \mathrm{ND}_T/T = 0$. Observe here that in (2) and (4), the delayed costs and constraint functions are both evaluated at the old decision vector $\mathbf{x}_{t-\boldsymbol{\tau}(t)}$ rather than the currently available one $\mathbf{x}_t$. The reason for this disparity in the time index is that in order to establish stability (see Section IV), the delayed primal costs must be evaluated at delayed actions, a stipulation that has appeared in past convergence analysis of asynchronous stochastic methods [30]. Before continuing, we discuss a representative example.

Example (Estimation of a Correlated Random Field). In a Gauss-Markov random field, the value of the field at the location of sensor i, denoted by $\mathbf{x}_i$, is of interest. Consider a sequential estimation problem in which the agents of the sensor network acquire noisy linear transformations of the field's value at their respective positions. Denote $\boldsymbol{\theta}_{i,t} \in \mathbb{R}^q$ as the observation collected by sensor i at time t. Observations $\boldsymbol{\theta}_{i,t}$ are assumed to be noisy linear transformations $\boldsymbol{\theta}_{i,t} = \mathbf{H}_i\mathbf{x}_i + \mathbf{w}_{i,t}$ of a signal $\mathbf{x}_i \in \mathbb{R}^p$ contaminated with Gaussian noise $\mathbf{w}_{i,t} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ independently distributed across agents and time. Ignoring neighboring observations, the minimum mean square error local estimation problem at agent i can then be written in the form of (2) with $f_{i,t}(\mathbf{x}_{i,t}) = \|\mathbf{H}_i\mathbf{x}_{i,t} - \boldsymbol{\theta}_{i,t}\|^2$. In this paper, we are interested in cases where the signal $\boldsymbol{\theta}_{i,t-\tau_i(t)}$ is revealed at time t (where the delay comes from, for example, feature extraction or latency due to data acquisition) to sensor i which then proceeds to determine the causal signal estimate $\mathbf{x}_{i,t} \in \mathbb{R}^p$ as a function of past observations $\boldsymbol{\theta}_{i,u}$ for $u = 1, \ldots, t-\tau_i(t)$ and information received from neighboring agents in previous time slots. The quality of these estimates can be improved using the correlated information of adjacent agents but would be hurt by trying to make estimates uniformly equal across the network. In this setting, the network regret and discrepancy, respectively, take the form

$$\mathbf{Reg}_T = \sum_{t=1}^{T}\sum_{i=1}^{N} \|\mathbf{H}_i\mathbf{x}_{i,t-\tau_i(t)} - \boldsymbol{\theta}_{i,t-\tau_i(t)}\|^2 \quad (6)$$
$$- \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T}\sum_{i=1}^{N} \|\mathbf{H}_i\mathbf{x} - \boldsymbol{\theta}_{i,t-\tau_i(t)}\|^2$$

$$\mathrm{ND}_T = \sum_{t=1}^{T}\sum_{(i,j)\in\mathcal{E}} [(1/2)\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\|^2 - \gamma_{ij}]_+. \quad (7)$$

The constraint $(1/2)\|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\|^2 \leq \gamma_{ij}$ makes the estimate $\mathbf{x}_i^*$ of agent i close to the estimates $\mathbf{x}_j^*$ of neighboring agents $j \in n_i$ but not so close to the estimates $\mathbf{x}_k^*$ of nonadjacent agents $k \notin n_i$. The problem formulation in (6) is a particular case of (2) with $f_{i,t}(\mathbf{x}_{i,t}) = \|\mathbf{H}_i\mathbf{x}_{i,t} - \boldsymbol{\theta}_i\|^2$ and (4) with $h_{ij}(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) = (1/2)\|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\|^2$.

III. Algorithm Development

To develop algorithms that yield sublinear growth in regret (2) and network discrepancy (4), we define the *online augmented Lagrangian* of a networked learning rule at time t as

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N}\Bigg[f_{i,t}(\mathbf{x}_i) \quad (8)$$
$$+ \sum_{j\in n_i}\left(\lambda_{ij}\left(h_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \gamma_{ij}\right) - \frac{\delta\epsilon}{2}\lambda_{ij}^2\right)\Bigg].$$

where the stacked vector $\boldsymbol{\lambda} := [\lambda_1; \cdots; \lambda_M] \in \mathbb{R}^M$ is the dual variable associated with each edge $(i,j) \in \mathcal{E}$ constraint. The saddle point method applied [31] to the Lagrangian stated in (8) proposed in [9] takes the following form

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_t - \epsilon\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\Big], \quad (9)$$

$$\boldsymbol{\lambda}_{t+1} = \Big[\boldsymbol{\lambda}_t + \epsilon\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\Big]_+, \quad (10)$$

where $\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ and $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)$, are the primal and dual gradients of the augmented Lagrangian in (8) with respect to $\mathbf{x}$ and $\boldsymbol{\lambda}$, respectively. The notation $\mathcal{P}_{\mathcal{X}}(\mathbf{x})$ in (9) describes the component wise orthogonal projection of primal variables $\mathbf{x}_i$ onto the given convex compact set $\mathcal{X}$, and $[\cdot]_+$ denotes the projection onto the M-dimensional nonnegative orthant $\mathbb{R}_+^M$.

The updates given in (9) and (10) can be implemented in decentralized manner as proposed in [15, Prop. 1]. But for the implementation of primal update at agent i [15, Prop. 1], each agent will require the loss $f_{i,t}(\mathbf{x}_{i,t})$ and its gradient $\nabla_{\mathbf{x}_i} f_i(\mathbf{x}_{i,t})$ from the adversary. The constraint function $h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is assumed to be known at each agent and only requires parameters $\mathbf{x}_j$ from the neighbor agents to calculate the gradient $\nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and perform the update of (9). The bottleneck in the implementation of (9) and (10) is the assumption of availability of feedback $\nabla_{\mathbf{x}_i} f_i(\mathbf{x}_{i,t})$ within the same time slot in which the action $\mathbf{x}_{i,t}$ is performed. In practice, the requirement that $f_{i,t}(\mathbf{x}_{i,t})$ be available for the update of agent i is violated for the setting considered in this work where convex costs are revealed with random delays $\tau_i(t)$. This delay may be caused by latency in data acquisition or feature extraction. Therefore, to address the fact that convex costs are revealed

at delayed time slots, and that each agent's performance is quantified in terms of delayed regret and constraint violation, we develop a protocol in which agents are allowed to operate asynchronously. As in Section II, associate to each agent i a delay of $0 \leq \tau_i(t) \leq \tau$ which denotes the respective gradient availability of its local objective. The dual variable $\boldsymbol{\lambda}_t$ is still assumed to be passed among the agents within the time frame of time t. Let us denote the stacked delayed primal variable as $\mathbf{x}_{t-\boldsymbol{\tau}(t)} := \left[\mathbf{x}_{1,t-\tau_1(t)}, \mathbf{x}_{2,t-\tau_2(t)}, \cdots, \mathbf{x}_{N,t-\tau_N(t)}\right]$. For the asynchronous saddle point algorithm, the updates take the form

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left[\mathbf{x}_t - \epsilon \nabla_{\mathbf{x}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)\right], \tag{11}$$

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \epsilon \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)\right]_+, \tag{12}$$

We define the k^{th} component of stacked primal gradient of the online Lagrangian in (8) evaluated at time $t - \boldsymbol{\tau}(t)$ as

$$[\nabla_{\mathbf{x}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)]_k = \nabla_{\mathbf{x}_k} f_{k,t-\tau_k(t)}(\mathbf{x}_{k,t-\tau_k(t)}) + \frac{1}{2} \sum_{j \in n_k} (\lambda_{kj,t} + \lambda_{jk,t}) \nabla_{\mathbf{x}_k} h_{kj}(\mathbf{x}_{k,t-\tau_k(t)}, \mathbf{x}_{j,t-\tau_j(t)}). \tag{13}$$

Further, the dual gradient of the online Lagrangian (8) at time $t - \tau(t)$ is given as

$$[\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)]_k = h_{kj}(\mathbf{x}_{k,t-\tau_k(t)}, \mathbf{x}_{j,t-\tau_j(t)}) - \gamma_{kj} - \delta\epsilon\lambda_{kj,t}. \tag{14}$$

for all $j \in n_k$. Due to the fact that the online gradient computations in (13) and (14) associated to agent k decouple from variables that depend on any other agent in the network except for those in n_k, we obtain that (11) and (12) yield a distributed algorithm, as summarized in the following proposition.

Proposition 1 *The primal and dual updates of* (11) *and* (12) *can be written in terms of N parallel primal with respect to local variables $\mathbf{x}_{i,t}$ as*

$$\mathbf{x}_{i,t+1} = \mathcal{P}_{\mathcal{X}}\left[\mathbf{x}_{i,t} - \epsilon\Big(\nabla_{\mathbf{x}_i} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) + \frac{1}{2} \sum_{j \in n_i} (\lambda_{ij,t} + \lambda_{ji,t}) \nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\Big)\right]. \tag{15}$$

Similarly, updating $\boldsymbol{\lambda}_t$ separates into M updates of $\lambda_{ij,t}$

$$\lambda_{ij,t+1} = \Big[(1-\epsilon^2\delta)\lambda_{ij,t} + \epsilon\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\right)\Big]_+. \tag{16}$$

In contrast to the global time index required to implement (9) and (10), the implementation of (15) and (16) does not require waiting for the availability of local cost at time t before updating. As summarized in Algorithm 1, at time slot t, after performing the action $\mathbf{x}_{i,t}$, agent i receives delayed loss $f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)})$ and computes the delayed gradient $\nabla_{\mathbf{x}_i} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)})$ at time stamp $t - \tau_i(t)$. Then each agent transfers its action $\mathbf{x}_{i,t-\tau_i(t)}$ to its neighbors to calculate the proximity constraint $\nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_{i,t-\tau_i(\tau)}, \mathbf{x}_{j,t-\tau_{j,t}})$ and update its primal and dual variables [cf. (15) - (16)]. Although each agent has $\mathbf{x}_{i,t}$, it evaluates its cost and constraints at earlier times $\mathbf{x}_{i,t-\tau_i(\tau)}$ which is required for sublinear regret.

It is remarked that the asynchronous algorithm updates in (15) and (16) automatically incorporates the issue of missing updates in the network. For instance, to update the dual variable in (16) at time instance t, if agent i and j are unable to communicate with each other then older values of primal variables can be used. Further extending the idea to a large network, this asynchronous implementation allows the agents to communication only once after certain time slots. This feature is important if the agents are located far away in the network or they seek to minimize the energy consumption.

Algorithm 1 AOSP: Asynchronous Online Saddle Point

Require: initialization $\mathbf{x}_0$ and $\boldsymbol{\lambda}_0 = \mathbf{0}$, step-size ϵ, regularizer δ

for $t = 1, 2, \ldots, T$ **do**
  loop in parallel agent $i \in V$
    Observe delayed cost $f_{i,t-\tau_i(t)}(\mathbf{x})$, compute grad. (13).
    Send primal, dual vars. $\mathbf{x}_{i,t-\tau_i(t)}, \boldsymbol{\lambda}_{ij,t}$ to nbhd. $j \in n_i$
    Receive variables $\mathbf{x}_{j,t-\tau_j(t)}, \boldsymbol{\lambda}_{ji,t}$ from neighbors $j \in n_i$
    Update action $\mathbf{x}_{i,t+1}$ using (15) local parameter $\mathbf{x}_{i,t}$

$$\mathbf{x}_{i,t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_{i,t} - \epsilon\Big(\nabla_{\mathbf{x}_i} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) + \frac{1}{2} \sum_{j \in n_i} (\lambda_{ij,t} + \lambda_{ji,t}) \nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\Big)\Big].$$

  end loop
  loop in parallel communication link $(i,j) \in \mathcal{E}$
    Update dual variables at network link (i,j) [cf. (16)]

$$\lambda_{ij,t+1} = \Big[(1-\epsilon^2\delta)\lambda_{ij,t} + \epsilon\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\right)\Big]_+$$

  end loop
end for

IV. Regret Bounds

In this section, we establish that the proposed asynchronous online saddle point method (Algorithm 1) achieves regret bounds [cf. (2)] and bounds on the time-aggregation of constraint violation [cf. (4)] which are sublinear in the final time T, and thus solve distributed asynchronous online learning problem with convex inequality constraints. To obtain these results, technical conditions are required which we state next.

Assumption 1 *The objective $f_{i,t}(\mathbf{x})$ for each agent i is Lipschitz continuous with constant L_f and satisfies*

$$\|f_{i,t}(\mathbf{x}) - f_{i,t}(\tilde{\mathbf{x}})\| \leq L_f \|\mathbf{x} - \tilde{\mathbf{x}}\|. \tag{17}$$

for all t. Likewise, the constraint function $h_{ij}(\mathbf{x})$ for each edge (i,j) satisfies

$$\|h_{ij}(\mathbf{x}) - h_{ij}(\tilde{\mathbf{x}})\| \leq L_h \|\mathbf{x} - \tilde{\mathbf{x}}\|. \tag{18}$$

Assumption 2 *The set $\mathcal{X}$ onto which the primal variables are projected contains the constrained optimizer as defined in* (5)*.*

Assumption 3 *For the constraint function, it holds that*

$$D := \max_i \max_{\mathbf{x} \in \mathcal{X}} h_{ij}(\mathbf{x}, \mathbf{x}_j) \leq L_g R \quad for\ all \quad j \in n_i. \tag{19}$$

Assumption 4 (Delay model) *The delay introduced at each agent i is upper bounded by a finite number τ as $\tau_i(t) \leq \tau$.*

Assumption 1 is called Lipschitz continuity and associated with the smoothness of the objective, constraint function and bounds the maximum changes in function values between distinct points over the domain. Assumption 2 ensures that optimal primal variable $\mathbf{x}_T^* \in \mathcal{X}$ for all T as defined in (5). This assumption holds directly if Slater's condition is satisfied. In Assumption 3, we assume a bound on the maximum possible value of the constraint function similar to that of [32]. Assumption 4 describes the type of delay model used in this paper. This assumptions makes sure that the delays are finite and bonded above by a constant τ. In order to proceed with the main analysis of this paper, upper bounds on the squared norm quantities $\|\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda})\|^2$ and $\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda})\|^2$ are required. The bounds developed in [15, eq. 19, 20] holds directly, since they are established for arbitrary $\mathbf{x}$ and $\boldsymbol{\lambda}$:

$$[\|\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda})\|^2] \leq 2(N+M^2)L^2(1+\|\boldsymbol{\lambda}\|^2) \tag{20}$$

with $L := \max(L_f, L_h)$, and

$$\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda})\|^2 \leq 2ML_g^2R^2 + 2\delta^2\epsilon^2\|\boldsymbol{\lambda}\|^2. \tag{21}$$

The idea of representing gradient bounds in terms of dual variable and then using the dual regularization term $(\delta\epsilon/2)\|\boldsymbol{\lambda}\|^2$ term to control the growth of delayed network discrepancy is extended to the asynchronous case as well [15].

Remark 1 One may interpret online convex optimization as a repeated adversarial game, as is common in game theory [33]. In this setting, at time t, an adversary is assumed to select a loss function f_t informing the merit of an action $\mathbf{x}_t$ taken by the learner. In the asynchronously delayed setting of this work, for each taken action $\mathbf{x}_{i,t}$ is succeeded by the revealing of a loss function $f_{i,t-\tau_i(t)}$ as feedback. Observe, however, that we do not evaluate the loss $f_{i,t-\tau_i(t)}$ at the most recent action but instead at the delayed action $\mathbf{x}_{i,t-\tau_i(t)}$. Doing so allows us to address the possible instabilities caused by asynchrony associated with delay $\tau_i(t)$. We can interpret these delays as being selected by an adversary, as in [34]. In principle, a whole family of costs at times $[t-\tau_i(t), t]$ could be revealed to the agent, but in this work we evaluate the most recent gradient.

For the establishment of sub-linear regret bound and delayed network discrepancy, the following lemma established.

Lemma 1 *Under Assumptions 1 - 4, for the sequence of* $(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ *generated by the saddle point algorithm in* (11) *and* (12) *with step-size* ϵ*, it holds that*

$$\begin{aligned}
&\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}) - \mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x},\boldsymbol{\lambda}_t) \\
&\leq \frac{1}{2\epsilon}\big(\|\mathbf{x}_t-\mathbf{x}\|^2 - \|\mathbf{x}_{t+1}-\mathbf{x}\|^2 + \|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2 - \|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\big) \\
&+\frac{\epsilon}{2}\Big(\big\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\big\|^2 + \|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\|^2\Big) \\
&+ \nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}_t).
\end{aligned} \tag{22}$$

Variants of this lemma appear in past works on primal-dual methods for constrained online learning [15], [32], [35], and make use of the convex-concave structure of the Lagrangian in (8). The main difference here is the primal variable $\mathbf{x}_{t-\boldsymbol{\tau}(t)}$ used for Lagrangian and its gradient. Due to the asynchronous nature of the proposed algorithm, we get an extra term involving inner product of primal gradient and difference of $\mathbf{x}_{t-\boldsymbol{\tau}(t)}$ and $\mathbf{x}_t$. Clearly, for synchronous case with $\boldsymbol{\tau}(t) = \mathbf{0}$, the result in (22) is identical to [15, Lemma 1]. The asynchronous extension (Lemma 1) allows us to establish the sublinear regret bound and sublinear delayed network discrepancy stated next.

Theorem 1 *The network regret* $\mathbf{Reg}_T$ *[cf.* (2)*] for the set of actions* $(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ *generated by Algorithm 1 with constant step-size* $\epsilon = T^{-1/2}$ *grows sublinearly in final time* T*:*

$$\mathbf{Reg}_T \leq \mathcal{O}(\sqrt{T}). \tag{23}$$

Moreover, the delayed network discrepancy (constraint violation) of the algorithm grows sublinearly in final time T *as*

$$ND_T \leq \mathcal{O}(T^{3/4}). \tag{24}$$

Theorem 1 establishes that the sequence of actions chosen by each agent in the network according to Algorithm 1 yield both a sublinear growth in the delayed regret and network discrepancy (constraint violation). These results are comparable to existing results for the setting in which the network operates on a synchronized clock [15], but do not require global coordination among agents in order to achieve convergence. These bounds are similar to mean convergence behavior of primal-dual stochastic methods to an error-neighborhood of a radius depending on the final iteration index.

Remark 2 It can be seen that the step size ϵ in Theorem 1 depends on the time horizon T. In real-time or streaming settings where T is not known in advance, it is possible to use the so-called doubling trick [36, Sec. 2.3.1]. The idea is to divide the time horizon into epochs such that the k-th epoch is of duration 2^k. Restarting the algorithm at each epoch with the corresponding value of ϵ still yields the same regret and network discrepancy as that in Theorem 1, up to a constant factor.

A. Regret bounds with zero network discrepancy

For certain applications such as cellular resource allocation where constraints translate to physical limitations such as maximum transmit power, constraint violation may not be an option. Likewise, if the agents in a system are required to be within the communication range, violating the constraints may be catastrophic. In such cases, it may be desirable to enforce zero network discrepancy even at the cost of higher regret. This section puts forth a modified algorithm that achieves zero network discrepancy while still incurring sublinear regret. Intuitively, this is accomplished by tightening the constraints in (5) to be of the form $h_{ij}(\mathbf{x}_i,\mathbf{x}_j) + \kappa <= \gamma_{ij}$ where $\kappa > 0$ is a small constant that is carefully selected to yield zero ND_T while keeping the regret bounded by $\mathcal{O}(T^{3/4})$. A similar approach was first proposed in [32] within the synchronous and centralized setting.

For the sake of brevity, let us redefine some notations for this subsection. We consider

$$h(\mathbf{x}) := \max_{(i,j)\in\mathcal{E}}(h_{ij}(\mathbf{x}_i,\mathbf{x}_j) - \gamma_{ij}), \tag{25}$$

and $h(\mathbf{x}) \leq 0$ is a sufficient condition to satisfy all the constrains. Next, we need a sequence of actions $\mathbf{x}_t$ such that $\text{ND}_T = \sum_{t=1}^{T} h(\mathbf{x}_{t-\tau(t)}) \leq 0$ in a long run. The idea is to add a constant κ to constraint in (25) and then solve the following optimization problem

$$\mathbf{x}_*^\kappa = \underset{\mathbf{x}}{\text{argmin}} \sum_{t=1}^{T} F_{t-\tau(t)}(\mathbf{x}) \quad \text{s. t. } h(\mathbf{x}) + \kappa \leq 0. \tag{26}$$

In order to proceed with the analysis, we make the following assumption.

Assumption 5 *The norm of the gradient of the modified constraint function $h(\mathbf{x})$ is lower bounded at the boundary as*

$$\min_{h(\mathbf{x})+\kappa=0} \|\nabla h(\mathbf{x})\| \geq \sigma. \tag{27}$$

Considering the Assumption 5 and the result in [32, Theorem 7], we present the following Corollary 1.

Corollary 1 *Let $\mathbf{x}_\star^0$ is the solution of* (26) *while setting $\kappa = 0$, and $F(\mathbf{x}) = \sum_{t=1}^{T} F_{t-\tau(t)}(\mathbf{x})$ we have*

$$|F(\mathbf{x}_\star^0) - F(\mathbf{x}_\star^\kappa)| \leq \frac{L_f}{\sigma} \kappa T. \tag{28}$$

Proof: The optimization problem in (26) can be equivalently written using the Lagrangian relaxation as

$$F(\mathbf{x}_\star^\kappa) = \min_{\mathbf{x}} \max_{\lambda \geq 0} \sum_{t=1}^{T} F_{t-\tau(t)}(\mathbf{x}) + \lambda(h(\mathbf{x}) + \kappa). \tag{29}$$

Let $\mathbf{x}_\star^\kappa$ and $\lambda_\star^\kappa$ be the optimal solutions to (29), which implies

$$\begin{aligned} F(\mathbf{x}_\star^\kappa) &= \min_{\mathbf{x}} \sum_{t=1}^{T} F_{t-\tau(t)}(\mathbf{x}) + \lambda_\star^\kappa(h(\mathbf{x}) + \kappa) && (30) \\ &\leq \sum_{t=1}^{T} F_{t-\tau(t)}(\mathbf{x}_\star^0) + \lambda_\star^\kappa(h(\mathbf{x}_\star^0) + \kappa) && (31) \\ &\leq F(\mathbf{x}_\star^0) + \kappa\lambda_\star^\kappa && (32) \end{aligned}$$

where the inequality in (31) follows from the optimality of $\mathbf{x}_\star^\kappa$, and second inequality of (32) follows from the optimality of $\mathbf{x}_\star^0$ which implies that $h(\mathbf{x}_\star^0) \leq 0$. In order to get the result in Corollary 1, we need to upper bound the dual optimal $\lambda_\star^\kappa$. From the optimality condition in (29), we can write

$$-\nabla F(\mathbf{x}_\star^\kappa) = \lambda_\star^\kappa \nabla h(\mathbf{x}_\star^\kappa). \tag{33}$$

From Assumption 1 and Assumption 5, it holds that

$$\lambda_\star^\kappa \leq \frac{\|\nabla F(\mathbf{x}_\star^\kappa)\|}{\|\nabla h(\mathbf{x}_\star^\kappa)\|} \leq \frac{L_f T}{\sigma}. \tag{34}$$

■

Under the Assumption 5 and Corollary 1, we present the main result of this subsection in the form of Theorem 2 with proof provided in Appendix C.

Theorem 2 *With constant κ added to the constraint, for the set of actions $(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ generated by proposed algorithm, it holds that*

$$\mathbf{Reg}_T \leq \mathcal{O}(T^{3/4}) \tag{35}$$

with $\text{ND}_T \to 0$.

Theorem 2 establishes that it is possible to achieve zero network discrepancy while incurring a higher regret as compared to that in Theorem 1. As discussed earlier, such a result is useful for cases where constraint violations might be catastrophic.

V. Applications and Numerical tests

A. Correlated Random Field Estimation

We return to the problem of random spatially correlated random field estimation as detailed in Sec. II. Consider the planar field $\mathcal{A} \subseteq \mathbb{R}^2$ covered with N number of wireless sensors deployed uniformly in a grid form over the field. The spatial field between the two sensors i and j is correlate through the relation $\rho(\mathbf{x}_i, \mathbf{x}_j) := \exp^{-2.2\|l_i - l_j\|}$, where $l_i \in \mathcal{A}$ and $l_j \in \mathcal{A}$ are the location of the sensors i and j, respectively. Due to this spatial correlation, agents which are close to each other will have the same field while the distant agents are less important. Each sensor in the network collects $\boldsymbol{\theta}_{i,t}$ which is the noisy linear transformation of the original field $\mathbf{x}_i$ at time instant t. From this sequential reception of data, the task is to find an estimate $\mathbf{x}_{i,t}$ which will minimize the regret defined in (2) and satisfy the neighborhood constraints of (4) in long run. The updates for Algorithm 1 take the form

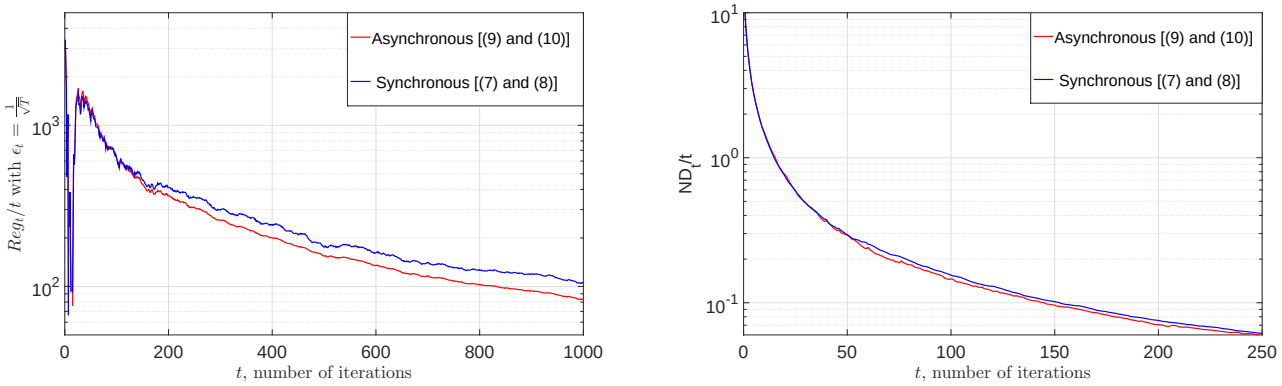
$$\begin{aligned} \mathbf{x}_{i,t+1} &= \mathcal{P}_{\mathcal{X}} \Big[\mathbf{x}_{i,t} - \epsilon_t \Big(2(\mathbf{H}_i \mathbf{x}_{i,t-\tau_i(t)} - \boldsymbol{\theta}_{i,t-\tau_i,t-\tau_i(t)}) \\ &\quad + \frac{1}{2} \sum_{j \in n_i} (\lambda_{ij,t} + \lambda_{ji,t}) (\big\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\big\|^2 - \gamma_{ij}) \Big) \Big], \\ \lambda_{ij,t+1} &= \Big[(1 - \epsilon_t^2 \delta) \lambda_{ij,t} + \frac{\epsilon_t}{2} [\big\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\big\|^2 - \gamma_{ij}] \Big]_+. \end{aligned} \tag{36}$$

For the simulations purpose, we consider a wireless sensor network of $N = 20$ agents for the scalar field estimation ($p = q = 1$) where agents are placed at constant distance over the planar region of $\mathcal{A} := \{(x, y) : 0 \leq x \leq 200, 0 \leq y \leq 200\}$. The spatial correlation factor γ_{ij} is considered to be $\gamma_{ij} = \rho(\mathbf{x}_i, \mathbf{x}_j)$. The other parameter values are $\sigma^2 = 50$, $\epsilon_t = \epsilon = 10^{-2.2}$, and $\delta = 10^3$. Observe that the matrix $\mathbf{H}$ will be scalar in this case an considered to be equal to 1 for all agents. For the asynchrony, the maximum delay is $\tau = 10$.

In Fig. 1a and Fig. 1b, respectively, we plot the evolution of $\mathbf{Reg}_t/t$ and ND_t/t for both synchronous ([15]) and asynchronous cases. The optimal $\mathbf{x}_T^*$ is calculated by solving the centralized problem in cvx. Observe that the average regret and constraint violation goes to zero as $T \to \infty$.

B. Target Localization with Moving Cameras

Target localization in the three-dimensional space has been a research problem of interest for a long time. It finds applications in radar systems [37], mobile positioning system



(a) Regret performance

(b) Network Discrepancy

Fig. 1: Algorithm 1 applied to random spatially correlated field estimation. Observe that stability behavior in both the asynchronous and synchronous implementations are comparable. Thus, we may solve decentralized online learning problems without a synchronized clock.

in wireless networks [38], search and rescue missions [39], and sensor networks [40]. One of the popular methods for target localization is using Angle of Arrival (AOA) measurements. To get AOA measurements, commonly directional antennas are deployed on each sensor in the network [41], [42]. But for vision based target localization which is problem of interest here, moving cameras are deployed at each sensor for the measurement. For instance, think of a scenario in which cameras mounted on UAVs to hover around the target for the localization purpose as described in Fig. 2. A similar problem is considered in [43] with the assumption of fixed cameras. But in realistic scenarios, the assumption of fixed cameras is very restrictive because the target location could be arbitrary and sometimes it is not even possible to install cameras around the location of the target. Therefore, it becomes necessary to create a temporary network by sending camera mounted on UAVs to get the location of the target.

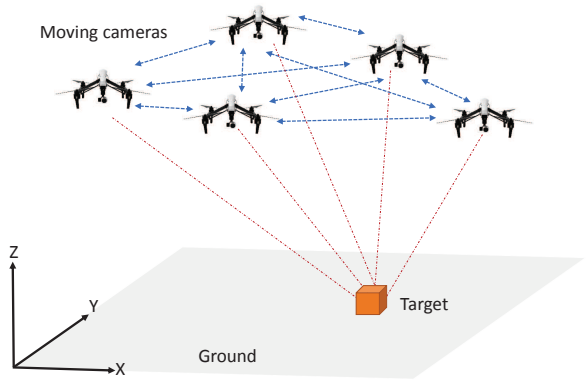


Fig. 2: Vision based target localization with moving cameras

To formulate the problem, consider a connected, undirected network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of $N = |\mathcal{V}|$ agents and $\mathcal{E} = |\mathcal{E}|$ edges. Each agent is equipped with a camera and agent is moving in nature such as UAVs. Each camera i cooperate with the other cameras in its neighborhood n_i to estimate the target location. To capture the three-dimensional location of the target, we need at least two cameras since width information is lost while taking picture with a camera. It is assumed that the time-varying location $\mathbf{c}_i(t) \in \mathbb{R}^3$ in global coordinates and orientation of the cameras is sequentially revealed at each time instant t. Each camera i estimates the unit directional vector $\mathbf{d}_i(t)$ from the image it captures of the target and then convert it into global coordinate unit directional vector $\mathbf{s}_i(t)$ using known rotation matrix $\mathbf{R}_i$ as described in [43]. With the camera coordinates $\mathbf{c}_i(t)$ and directional vector $\mathbf{s}_i(t)$, we obtain a line in 3-dimensional space representing sight of the target for each camera i as follows

$$\ell_i(t) : \frac{x - \mathbf{c}_i^x(t)}{\mathbf{s}_i^x(t)} = \frac{y - \mathbf{c}_i^y(t)}{\mathbf{s}_i^y(t)} = \frac{z - \mathbf{c}_i^z(t)}{\mathbf{s}_i^z(t)}, \tag{37}$$

where the superscripts x, y, z represents the x, y, z coordinates in 3-dimensional space.

There are N such lines at each time instant t generated by each camera. Since the target might be away from the observed line, it is required to measure the perpendicular distance of the target from each line. Let us assume that the perpendicular drawn from target to each line will intersect at a point $\mathbf{o}_i(t)$ for camera i. Since this point is on the line $\ell_i(t)$, it holds that

$$\frac{\mathbf{o}_i^x(t) - \mathbf{c}_i^x(t)}{\mathbf{s}_i^x(t)} = \frac{\mathbf{o}_i^y(t) - \mathbf{c}_i^y(t)}{\mathbf{s}_i^y(t)} = \frac{\mathbf{o}_i^z(t) - \mathbf{c}_i^z(t)}{\mathbf{s}_i^z(t)} = a_i(t). \tag{38}$$

Alternatively, since the line from target to $\ell_i(t)$ is a perpendicular, we can write

$$a_i(t) = \frac{\mathbf{s}_i(t)^T(\mathbf{z} - \mathbf{c}_i(t))}{\mathbf{s}_i(t)^T\mathbf{s}_i(t)} \tag{39}$$

which further results in

$$\mathbf{o}_i(t) = a_i(t)\mathbf{s}_i(t) + \mathbf{c}_i(t). \tag{40}$$

Using (39) and (40), the objective function of minimizing the perpendicular distance between line $\ell_i(t)$ and target location $\mathbf{z}$ can be written as

$$f_{i,t}(\mathbf{z}) := \|\mathbf{o}_i(t) - \mathbf{z}\|^2. \tag{41}$$

Next, from practical perspective, it is motivated that location estimate of each camera need not be exactly equal to the true

location of the target $\mathbf{z}$. The location estimates of each camera could be within $\sqrt{\gamma}$ ball around the target location. Hence, each camera need to estimate the location $\mathbf{z}_i$ such that

$$(1/2)\left\|\mathbf{z}^i-\mathbf{z}^j\right\|^2\leq\gamma \tag{42}$$

for all $(i,j)\in\mathcal{E}$ and where γ is a predefined constant. From (41) and (42), the online augmented Lagrangian can be written as

$$\mathcal{L}_t(\{\mathbf{z}_i\}_i,\boldsymbol{\lambda})=\sum_{i=1}^{N}\Bigg[\left\|\mathbf{o}_i(t)-\mathbf{z}\right\|^2 + \frac{1}{2}\sum_{j\in n_i}\left(\lambda_{ij}\left((1/2)\left\|\mathbf{z}^i-\mathbf{z}^j\right\|^2-\gamma\right)-\frac{\delta\epsilon}{2}\lambda_{ij}^2\right)\Bigg]. \tag{43}$$

The goal here is to solve the problem formulated in (43) in a decentralized and asynchronous manner. Since the directional vectors $\mathbf{s}_i(t)$ for each camera i are sequentially available, we apply the proposed asynchronous saddle point algorithm to solve (43). Further, since each camera is moving and captures a new image at each time instant t, it may take some processing time to calculate the directional vector for the corresponding time instance $\mathbf{s}_i(t)$. Therefore, an asynchronous algorithm is required which can utilize the previously calculated directional vectors (calculated at $t-\tau_i(t)$) at current time instant t. We associate a dual variable λ_{ij} to each constraint and corresponding primal and dual updates are written as

$$\mathbf{z}_{i,t+1}=\mathbf{z}_{i,t}-\epsilon\Big[\nabla f_{i,t-\tau_{i,t}}(\mathbf{z}_{i,t-\tau_{i,t}}) + \sum_{j\in n_i}\lambda_{i,j}(t)(\mathbf{z}_{i,t-\tau_{i,t}}-\mathbf{z}_{j,t-\tau_{j,t}})\Big] \tag{44}$$

$$\lambda_{ij,t+1}=\Big[(1-\epsilon^2\delta)\lambda_{ij,t}+\epsilon(\left\|\mathbf{z}_{i,t-\tau_{i,t}}-\mathbf{z}_{j,t-\tau_{j,t}}\right\|^2-2\gamma)\Big]_+ \tag{45}$$

For the simulation purposes, we consider a network of $N=5$ cameras hovering around a target. The cameras are free to move in any direction as long as they can see the target to be localized. Fig. 3a shows the regret performance of centralized synchronous stochastic gradient descent and decentralized asynchronous saddle point methods. It is clear that both the algorithms have the similar regret rate but the later exhibits the decentralized implementation which is of more practical importance. Fig. 3b shows the network constraint violation for a particular pair of agents $(i,j)\in\mathcal{E}$ which clearly shows that it goes to zero as $T\to\infty$. Next, Fig. 4 demonstrates the working of the vision based target localization with moving cameras. It has been observed that the arbitrary motion of the cameras does not affect the estimates of the target location at each agent (camera). In this figure, red square shows the target position and all other triangles shows the different camera positions at different time instances. It is clear that the estimate of target location at each camera comes closer to the actual positions with in $40-50$ iterations of the proposed algorithm represented by solid lines of different color.

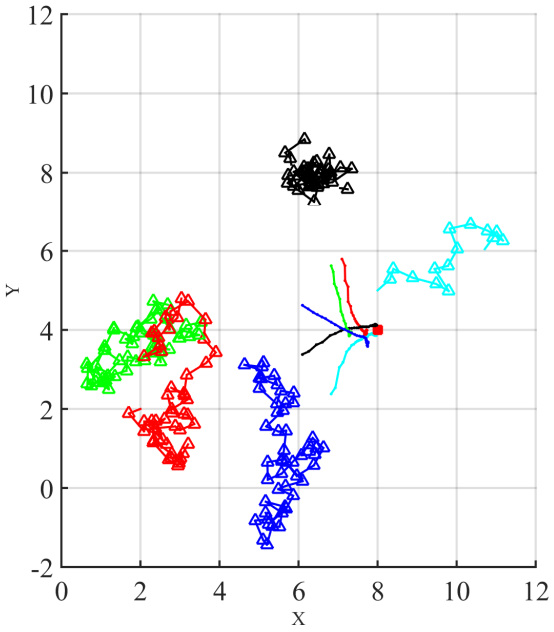


Fig. 4: Top view of vision based target localization using five cameras with target on ground. The marked lines represents the different camera movements and solid lines represents the local estimation of the target at each camera.

VI. Conclusion

We consider a multi-agent system comprising of agents with heterogeneous computational capabilities. Each agent has a local objective that is revealed to it in an online fashion, but seeks to cooperate with other agents in order to minimize the global objective. A distributed and asynchronous online saddle point algorithm is proposed that allows the agent actions to be related via a non-linear constraint. The proposed algorithm readily handles adversarial bounded delays in the gradient information, while incurring a regret of at most $\mathcal{O}(\sqrt{T})$. Due to the online and distributed nature of the problem, the agents also incur a network discrepancy of $\mathcal{O}(T^{3/4})$. A modified saddle point algorithm is also proposed that eliminates the network discrepancy but incurs a regret of $\mathcal{O}(T^{3/4})$. The proposed algorithm is tested on the problem of correlated random field estimation and that of vision-based target localization with moving cameras.

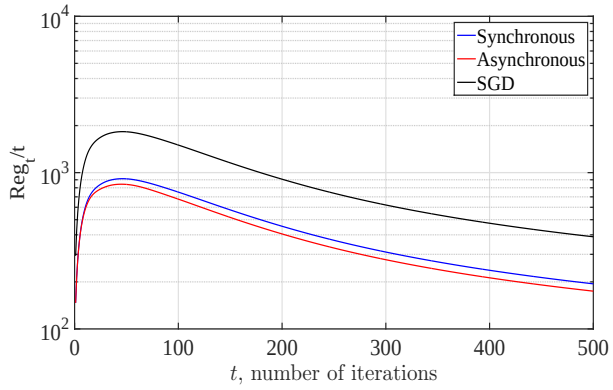
Appendix A
Proof of Lemma 1

The proof is split into two parts, corresponding to establishing an upper bound on $\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x},\boldsymbol{\lambda}_t)$ and a lower bound on $\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda})$.

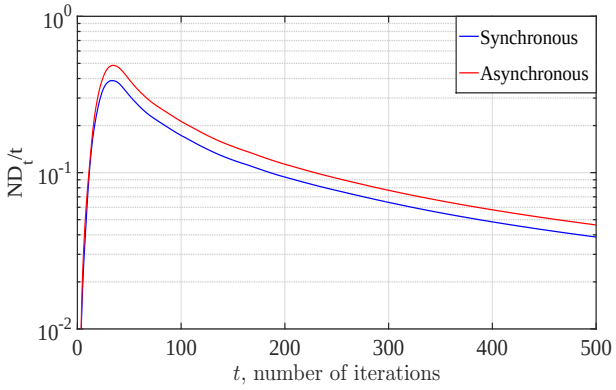
Bound on $\mathcal{L}_t(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda}_t)$: We begin with the observation that for any $\mathbf{x}\in\mathcal{X}$, it holds from (11) that

$$\begin{aligned}\|\mathbf{x}_{t+1}-\mathbf{x}\|^2&=\|\mathcal{P}_{\mathcal{X}}[\mathbf{x}_t-\epsilon\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)]-\mathbf{x}\|^2\\&\leq\|\mathbf{x}_t-\epsilon\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathbf{x}\|^2 \qquad (46)\\&=\|\mathbf{x}_t-\mathbf{x}\|^2-2\epsilon\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_t-\mathbf{x})\\&\quad+\epsilon^2\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\|^2, \qquad (47)\end{aligned}$$

where (46) follows from the fact that the projection operator $\mathcal{P}_{\mathcal{X}}[\cdot]$ is non-expansive. Next, adding



(a) Regret performance



(b) Network Discrepancy

Fig. 3: Algorithm 1 applied to vision based target localization problem. Observe that stability behavior in both the asynchronous and synchronous implementations are comparable. Thus, we may solve decentralized online learning problems without a synchronized clock.

$\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T\mathbf{x}_{t-\boldsymbol{\tau}(t)}$ on the both sides of (47) and rearranging, we obtain

$$\begin{aligned}&\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x})\\&\quad\leq\frac{1}{2\epsilon}\left(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2\right)\\&\quad+\frac{\epsilon}{2}\left\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\right\|^2\\&\qquad+\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}_t).\end{aligned}\tag{48}$$

Since the Lagrangian is a convex function of $\mathbf{x}$, the first order condition for convexity implies that

$$\begin{aligned}&\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x},\boldsymbol{\lambda}_t)\\&\qquad\leq\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}),\end{aligned}\tag{49}$$

which, together with (48) yields

$$\begin{aligned}&\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x},\boldsymbol{\lambda}_t)\\&\quad\leq\frac{1}{2\epsilon}\left(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2\right)\\&\qquad+\frac{\epsilon}{2}\left\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\right\|^2\\&\qquad\quad+\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}_t)\end{aligned}\tag{50}$$

which is the desired bound. ■

Bound on $\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda})$*:* The derivation of the bound proceeds along similar lines as before. In this case, for any $\boldsymbol{\lambda}\geq 0$ we have that

$$\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2=\|[\boldsymbol{\lambda}_t+\epsilon\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)]_+-\boldsymbol{\lambda}\|^2.\tag{51}$$

Using the non-expansiveness of the projection operator $[\cdot]_+$, concavity of the Lagrangian function with respect to $\boldsymbol{\lambda}$, and reordering the terms as before, we obtain

$$\begin{aligned}&\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda})\\&\quad\geq\frac{1}{2\epsilon}\left(\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2\right)\\&\qquad-\frac{\epsilon}{2}\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\|^2,\end{aligned}\tag{52}$$

which is the desired bound. ■

The required result in (22) follows simply by combining the bounds in (50) and (52).

Appendix B
Proof of Theorem 1

The theorem is divided into three main steps. First, an upper bound in developed for the lagrangian which consists of sum of regret term and constraint violation term. After obtained the upper bound in terms of step size ϵ, we separately derive the sublinear *delayed regret* and sublinear *network discrepancy*. Substituting the expression for the augmented Lagrangian (cf. (8)) and the bounds in (20)-(21) in (22), and collecting the common terms yields

$$\begin{aligned}&\sum_{i=1}^{N}[f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)})-f_{i,t-\tau_i(t)}(\mathbf{x}_i)]+\frac{\delta\epsilon}{2}(\|\boldsymbol{\lambda}_t\|^2-\|\boldsymbol{\lambda}\|^2)\\&+\sum_{(i,j)\in\mathcal{E}}\Big[\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\\&\qquad-\lambda_{ij,t}\left(h_{ij}(\mathbf{x}_i,\mathbf{x}_j)-\gamma_{ij}\right)\Big]\\&\leq\Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\Big)\\&\qquad+\frac{\epsilon}{2}\Big(K_1+K_2\|\boldsymbol{\lambda}_t\|^2\Big)\Big]\\&\qquad+\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\tau(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\tau(t)}-\mathbf{x}_t),\end{aligned}\tag{53}$$

where $K_1:=2(N+M^2)L^2+2ML_g^2R^2$ and $K_2:=2(N+M^2)L^2+2\delta^2\epsilon^2$. Recall that since $\mathbf{x}$ is feasible, it holds that $h_{ij}(\mathbf{x}_i,\mathbf{x}_j)-\gamma_{ij}\leq 0$ for all $(i,j)\in\mathcal{E}$. Therefore, the last term on the left hand side of (53) can be dropped. Recalling the definition of $F_t(\mathbf{x})$ from (2), (53) yields

$$\begin{aligned}&F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})+\frac{\delta\epsilon}{2}(\|\boldsymbol{\lambda}_t\|^2-\|\boldsymbol{\lambda}\|^2)\\&+\sum_{(i,j)\in\mathcal{E}}\left[\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]\\&\leq\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\Big)\\&\qquad+\frac{\epsilon}{2}\Big(K_1+K_2\|\boldsymbol{\lambda}_t\|^2\Big)\Big]+I,\end{aligned}\tag{54}$$

where

$$I:=\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\tau(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\tau(t)}-\mathbf{x}_t).\tag{55}$$

Let us develop upper bound for I first as follows.

$$I \leq \left\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\tau(t)},\boldsymbol{\lambda}_t)\right\| \left\|(\mathbf{x}_{t-\tau(t)}-\mathbf{x}_t)\right\|. \quad (56)$$

The Inequality in (56) follows from Cauchy- Schwartz inequality. Next, considering the term $\left\|\mathbf{x}_t-\mathbf{x}_{t-\tau(t)}\right\|$, note that

$$\begin{aligned}\left\|\mathbf{x}_t-\mathbf{x}_{t-\tau(t)}\right\| &\leq \sum_{k=t-\tau}^{t-1}\|\mathbf{x}_{k+1}-\mathbf{x}_k\| \\ &\leq \epsilon\sum_{k=t-\tau}^{t-1}\left\|\nabla_{\mathbf{x}}\mathcal{L}_{k-\tau(k)}(\mathbf{x}_{k-\tau(k)},\boldsymbol{\lambda}_l)\right\|. \quad (57)\end{aligned}$$

For brevity, let us denote $B_t := \left\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\right\|$ and substituting upper bound obtained in (57) into (56), we get

$$\begin{aligned}I &\leq \epsilon\sum_{k=t-\tau}^{t-1}[B_tB_k] \leq \frac{\epsilon}{2}\sum_{k=t-\tau}^{t-1}[B_t^2+B_k^2] \quad (58)\\ &= \frac{\epsilon}{2}\Big[\tau B_t^2+\sum_{k=t-\tau}^{t-1}B_k^2\Big]. \quad (59)\end{aligned}$$

In the above derivation, the second inequality in (58) follows from the application of $ab \leq \frac{a^2+b^2}{2}$. Next, applying the gradient norm square upper bound of (20), we obtain

$$\begin{aligned}I &\leq \frac{\epsilon}{2}\Big[\tau\Big((N+M)L^2(1+\|\boldsymbol{\lambda}_t\|^2)\Big) \\ &\qquad+\sum_{k=t-\tau}^{t-1}\Big((N+M)L^2(1+\|\boldsymbol{\lambda}_k\|^2)\Big)\Big] \quad (60)\\ &= \frac{\epsilon}{2}\Big[2\tau(N+M)L^2+\tau(N+M)L^2\|\boldsymbol{\lambda}_t\|^2 \\ &\qquad+(N+M)L^2\sum_{k=t-\tau}^{t-1}\|\boldsymbol{\lambda}_k\|^2\Big]. \quad (61)\end{aligned}$$

By multiplying the last term of (61) by τ, we get the following inequality

$$\begin{aligned}I &\leq \frac{\epsilon}{2}\Big[2\tau(N+M)L^2+\tau(N+M)L^2\|\boldsymbol{\lambda}_t\|^2 \\ &\qquad+\tau(N+M)L^2\sum_{k=t-\tau}^{t-1}\|\boldsymbol{\lambda}_k\|^2\Big] \quad (62)\\ &\leq \frac{\epsilon}{2}\tau K_1\Big[2+\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2\Big], \quad (63)\end{aligned}$$

where (63) holds from $(N+M)L^2 < K_1$. Substituting the bound obtained in (63) into (54), we get

$$\begin{aligned}&\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})+\frac{\delta\epsilon}{2}(\|\boldsymbol{\lambda}_t\|^2-\|\boldsymbol{\lambda}\|^2)\Big] \quad (64)\\ &\quad+\sum_{(i,j)\in\mathcal{E}}\left[\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]\\ &\leq\Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\Big)\\ &\quad+\frac{\epsilon}{2}\Big(K_1+K_2\|\boldsymbol{\lambda}_t\|^2\Big)\Big]+\frac{\epsilon}{2}\tau K_1\Big[2+\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2\Big] \quad (65)\\ &=\Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\Big)\Big]\\ &\quad+\frac{\epsilon}{2}K+\frac{\epsilon}{2}K_2\|\boldsymbol{\lambda}_t\|^2+\epsilon\tau K_1\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2. \quad (66)\end{aligned}$$

Last equality of (66) is obtained by combining the various constant terms and written as $K := K_1(1+2\tau)$. Assume that the delay is zero for $1 \leq t \leq \tau$ which is required to make the difference $t-\tau$ always positive. Take the summation on both sides of (66) from $t=1$ to T and telescopic sum yields

$$\begin{aligned}&\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})+\frac{\delta\epsilon}{2}(\|\boldsymbol{\lambda}_t\|^2-\|\boldsymbol{\lambda}\|^2)\\ &\quad+\sum_{(i,j)\in\mathcal{E}}\left[\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]\Big]\\ &\leq\sum_{t=1}^{T}\Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_1-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_1-\boldsymbol{\lambda}\|^2\Big)+\frac{\epsilon}{2}K\Big]\\ &\quad+\frac{1}{2}K_2\sum_{t=1}^{T}\epsilon\|\boldsymbol{\lambda}_t\|^2+\epsilon\tau K_1\sum_{t=1}^{T}\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2. \quad (67)\end{aligned}$$

The terms related to $-\|\mathbf{x}_{T+1}-\mathbf{x}\|$ and $-\|\boldsymbol{\lambda}_{T+1}-\boldsymbol{\lambda}\|$ are removed because they are upper bounded by 0. Observe that the last sum in (67) expression can be further expressed as

$$\begin{aligned}\sum_{t=1}^{T}\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2 &= \|\boldsymbol{\lambda}_1\|^2+(\|\boldsymbol{\lambda}_1\|^2+\|\boldsymbol{\lambda}_2\|^2)\\ &\quad+(\|\boldsymbol{\lambda}_1\|^2+\|\boldsymbol{\lambda}_2\|^2+\|\boldsymbol{\lambda}_3\|^2)+\cdots\\ &\quad+(\|\boldsymbol{\lambda}_{T-\tau}\|^2+\|\boldsymbol{\lambda}_{T-\tau+1}\|^2+\cdots+\|\boldsymbol{\lambda}_T\|^2). \quad (68)\end{aligned}$$

The representation in (68) would result in the following bound

$$\sum_{t=1}^{T}\sum_{k=t-\tau}^{t}\|\boldsymbol{\lambda}_k\|^2 \leq (\tau+1)\sum_{t=1}^{T}\|\boldsymbol{\lambda}_t\|^2. \quad (69)$$

Utilizing the upper bound obtained in (69) into (67), we get

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})+\frac{\delta\epsilon}{2}(\|\boldsymbol{\lambda}_t\|^2-\|\boldsymbol{\lambda}\|^2) + \sum_{(i,j)\in\mathcal{E}}\big[\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\big]\Big] \leq \frac{1}{2\epsilon}\Big(\|\mathbf{x}_1-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_1-\boldsymbol{\lambda}\|^2\Big)+\frac{\epsilon KT}{2} + \frac{\epsilon}{2}K_2\sum_{t=1}^{T}\|\boldsymbol{\lambda}_t\|^2+\epsilon\tau(\tau+1)\sum_{t=1}^{T}\|\boldsymbol{\lambda}_i\|^2. \tag{70}$$

Rearranging the like terms, we get

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})-\frac{\delta\epsilon}{2}\|\boldsymbol{\lambda}\|^2\Big] \tag{71}$$

$$+\sum_{(i,j)\in\mathcal{E}}\lambda_{ij}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right] \leq \frac{1}{2\epsilon}\Big(\|\mathbf{x}_1-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_1-\boldsymbol{\lambda}\|^2\Big)+\frac{\epsilon KT}{2} +\frac{\epsilon}{2}\sum_{t=1}^{T}\Big[K_2+2\tau(\tau+1)-\delta\Big]\|\boldsymbol{\lambda}_t\|^2. \tag{72}$$

Now for fixed T, we will select δ such that $K_2+2\tau(\tau+1)\leq\delta$ in order to drop the last terms associated with $\boldsymbol{\lambda}_t$. Finally, we have

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})-\frac{\delta\epsilon}{2}\|\boldsymbol{\lambda}\|^2\Big] \tag{73}$$

$$+\sum_{(i,j)\in\mathcal{E}}\lambda_{ij}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right] \leq \frac{1}{2\epsilon}\Big(\|\mathbf{x}_1-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_1-\boldsymbol{\lambda}\|^2\Big)+\frac{\epsilon KT}{2}. \tag{74}$$

Now, using the condition that $\boldsymbol{\lambda}_1=0$ and collecting the like terms of $\boldsymbol{\lambda}$ to the left hand side of (74), we get

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})\Big]-\left(\frac{\delta\epsilon T}{2}+\frac{1}{2\epsilon}\right)\|\boldsymbol{\lambda}\|^2 + \sum_{(i,j)\in\mathcal{E}}\left[\sum_{t=1}^{T}\lambda_{ij}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right] \leq \frac{1}{2\epsilon}\|\mathbf{x}_1-\mathbf{x}\|^2+\frac{\epsilon TK}{2}. \tag{75}$$

Now, let $g(\boldsymbol{\lambda})$ is defined as

$$g(\boldsymbol{\lambda})=\sum_{(i,j)\in\mathcal{E}}\lambda_{ij}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right] -\left(\frac{\delta\epsilon T}{2}+\frac{1}{2\epsilon}\right)\lambda_{ij}^2. \tag{76}$$

Calculating the gradient of (76) and then solving for each dual variable, we get

$$\tilde{\lambda}_{ij}=\frac{1}{\delta\epsilon T+1/\epsilon}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]_+. \tag{77}$$

Substituting it back in (75), we get

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})\Big] + \sum_{(i,j)\in\mathcal{E}}\frac{\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]_+^2}{\delta\epsilon T+1/\epsilon} \tag{78}$$

$$\leq\frac{1}{2\epsilon}\|\mathbf{x}_1-\mathbf{x}\|^2+\frac{\epsilon TK}{2}. \tag{79}$$

Now select $\epsilon=1/\sqrt{T}$, we get

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x})\Big] + \sum_{(i,j)\in\mathcal{E}}\frac{\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]_+^2}{\sqrt{T}(\delta+1)} \leq\frac{\sqrt{T}}{2}\left[\|\mathbf{x}_1-\mathbf{x}\|^2+K\right]. \tag{80}$$

To obtain the regret result: The second term of the left hand side of (80) is always lower bounded by zero, therefore it holds that

$$\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_T^\star)\Big] \leq\frac{\sqrt{T}}{2}\left[\|\mathbf{x}_1-\mathbf{x}_T^\star\|^2+K\right]=\mathcal{O}(\sqrt{T}). \tag{81}$$

For constraint violation sub-linear growth: Note that from the Lipschitz continuity of the objective function, we have $|F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_T^\star)|\leq L_f\left\|\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}_T^\star\right\|$. An immediate consequence of this inequality is that $F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)})-F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_T^\star)\geq-2L_fR$, using this in (80), we get

$$\sum_{(i,j)\in\mathcal{E}}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]_+^2 \leq\sqrt{T}(\delta+1)\left(\mathcal{O}(\sqrt{T})+2TL_fR\right) \leq(\delta+1)\sqrt{T}\left(\mathcal{O}(\sqrt{T})+2TL_fR\right). \tag{82}$$

Considering the lower bound for left hand side of (82), we get

$$\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)},\mathbf{x}_{j,t-\tau_j(t)})-\gamma_{ij}\right)\right]_+^2 \leq(\delta+1)\left(\mathcal{O}(T)+2T^{3/2}L_fR\right). \tag{83}$$

Take the square root on both sides and then sum over all (i,j), we get

$$\sum_{(i,j)\in\mathcal{E}}\left[\sum_{t=1}^{T}\left(h_{ij}(\mathbf{x}_{i,t-\tau},\mathbf{x}_{j,t-\tau})-\gamma_{ij}\right)\right]_+ \leq\sqrt{(\delta+1)}\left(\mathcal{O}(\sqrt{T})+2T^{3/4}L_fR\right). \tag{84}$$

Hence proved.

It is remarked that for the step size choice of $\epsilon = \frac{1}{\sqrt{T}}$, the selection of parameter δ is such that

$$\frac{2\delta^2}{T} + (N+M)L^2 + 2\tau(\tau+1) \leq \delta. \tag{85}$$

For a sufficiently large T, we can always choose $\delta \geq (N+M)L^2 + 2\tau(\tau+1)$.

APPENDIX C
PROOF OF THEOREM 2

The online augmented Lagrangian with the update constraint is written as

$$\mathcal{L}_t(\mathbf{x}, \lambda) = F_t(\mathbf{x}) + \lambda(h(\mathbf{x}) + \kappa). \tag{86}$$

Substituting this into the result of Lemma 1, we obtain

$$\begin{aligned}
&[F_{t-\tau(t)}(\mathbf{x}_{t-\tau(t)}) - F_{t-\tau(t)}(\mathbf{x})] + \frac{\delta\epsilon}{2}(\lambda_t^2 - \lambda^2) \\
&+ \Big[\lambda\left(h(\mathbf{x}_{t-\tau(t)}) + \kappa\right) - \lambda_t\left(h(\mathbf{x}) + \kappa\right)\Big] \\
&\leq \Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + (\lambda_t - \lambda)^2 - (\lambda_{t+1} - \lambda)^2\Big) \\
&\qquad + \frac{\epsilon}{2}\Big(K_1' + K_2'\lambda_t^2\Big)\Big] \\
&\qquad + \nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\tau(t)}, \lambda_t)^T(\mathbf{x}_{t-\tau(t)} - \mathbf{x}_t),
\end{aligned} \tag{87}$$

where $K_1' := 2(N+1)L^2 + 4L_g^2R^2 + 2\kappa^2$ and $K_2' := 2(N+1)L^2 + 2\delta^2\epsilon^2$. Now consider $\mathbf{x} = \mathbf{x}_\star^\kappa$, for which it holds that $h(\mathbf{x}_\star^\kappa) + \kappa \leq 0$. Therefore, the last term on the left hand side of (87) can be dropped. Applying the bound in (63) on the last term on right hand side of (87), we get

$$\begin{aligned}
&\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}) + \frac{\delta\epsilon}{2}(\lambda_t^2 - \lambda^2)\Big] \qquad (88)\\
&\quad + \left[\lambda\left(h(\mathbf{x}_{t-\tau(t)}) + \kappa\right)\right] \\
&\leq \Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + (\lambda_t - \lambda)^2 - (\lambda_{t+1} - \lambda)^2\Big) \\
&\quad + \frac{\epsilon}{2}\Big(K_1' + K_2'\lambda_t^2\Big)\Big] + \frac{\epsilon}{2}\tau K_1'\Big[2 + \sum_{k=t-\tau}^{t}\lambda_k^2\Big] \qquad (89)\\
&= \Big[\frac{1}{2\epsilon}\Big(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + (\lambda_t - \lambda)^2 - (\lambda_{t+1} - \lambda)^2\Big)\Big] \\
&\quad + \frac{\epsilon}{2}K' + \frac{\epsilon}{2}K_2'\lambda_t^2 + \epsilon\tau K_1'\sum_{k=t-\tau}^{t}\lambda_k^2. \qquad (90)
\end{aligned}$$

Last equality of (90) is obtained by combining the various constant terms and written as $K' := K_1'(1+2\tau)$. Assume that the delay is zero for $1 \leq t \leq \tau$ which is required to make the difference $t - \tau$ always positive. Taking the summation on both sides of (90) from $t = 1$ to T and applying the similar steps from (66) to (80), we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_\star^\kappa)\Big] \\
&\qquad + \frac{\Big[\sum_{t=1}^{T}h(\mathbf{x}_{t-\tau(t)}) + \kappa T\Big]_+^2}{\sqrt{T}(\delta+1)} \\
&\leq \frac{\sqrt{T}}{2}\left[\|\mathbf{x}_1 - \mathbf{x}_\star^\kappa\|^2 + K'\right].
\end{aligned} \tag{91}$$

To obtain the regret result: The second term of the left hand side of (80) is always lower bounded by zero, therefore it holds that

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_\star^\kappa)\Big] \\
&\leq \frac{\sqrt{T}}{2}\left[\|\mathbf{x}_1 - \mathbf{x}_\star^\kappa\|^2 + K'\right] = \mathcal{O}(\sqrt{T}).
\end{aligned} \tag{92}$$

Using triangle inequality, the result from (28), and setting $\kappa = cT^{-1/4}$ for some $c > 0$, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}^\star)\Big] \\
&\leq \mathcal{O}(\sqrt{T}) + \frac{cL_h}{\sigma}T^{3/4} := \mathcal{O}(T^{3/4}).
\end{aligned} \tag{93}$$

For constraint violation sub-linear growth: Note that from the Lipschitz continuity of the objective function, we have $|F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_T^\star)| \leq L_f\left\|\mathbf{x}_{t-\boldsymbol{\tau}(t)} - \mathbf{x}_T^\star\right\|$. An immediate consequence of this inequality is that $F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}) - F_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_T^\star) \geq -2L_fR$, using this in (80), we get

$$\begin{aligned}
&\left[\sum_{t=1}^{T}\left(h(\mathbf{x}_{t-\tau(t)}) + \kappa T\right)\right]_+^2 \\
&\leq (\delta+1)\sqrt{T}\left(\frac{\sqrt{T}}{2}\left[R^2 + K'\right] + 2TL_fR\right),
\end{aligned} \tag{94}$$

where we have used the fact that $\|\mathbf{x}_1 - \mathbf{x}_\star^\kappa\| \leq R$. Taking square root on both sides, we get

$$\begin{aligned}
&\sum_{t=1}^{T}h(\mathbf{x}_{t-\tau(t)}) \\
&\leq \sqrt{(\delta+1)\sqrt{T}\left(\frac{\sqrt{T}}{2}(R^2 + K') + 2TL_fR\right)} - \kappa T.
\end{aligned} \tag{95}$$

For sufficiently large T, we can assume that $2TL_fR >> \frac{\sqrt{T}}{2}(R^2 + K')$. This implies that

$$\sum_{t=1}^{T}h(\mathbf{x}_{t-\tau(t)}) \leq 2(\delta+1)L_fRT^{3/4} - \kappa T. \tag{96}$$

Next, considering $\kappa = cT^{-1/4}$ and $c = 2(\delta+1)L_fR$ ensures the zero constraint violation.

REFERENCES

[1] A. S. Bedi, A. Koppel, and K. Rajawat, "Beyond consensus and synchrony in decentralized online optimization using saddle point method," in *51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 293–297.

[2] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, 2006.

[3] L. Chen, S. Low, M. Chiang, and J. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," in *in Proc. IEEE INFOCOM*, April 2006, pp. 1–13.

[4] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, 2005.

[5] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource Allocation and Cross-layer Control in Wireless Networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, Apr. 2006.

[6] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. on Mach. Learn.*, vol. 20, no. 2, Washington DC, USA, Aug. 21-24 2003, pp. 928–936.

[7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, 2009.

[8] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. of IEEE American Control Conference (ACC)*, 2012, pp. 1067–1072.

[9] A. Koppel, F. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, p. 15, Oct 2015.

[10] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," in *Adv. Neural Inf. Process. Syst.*, 2008, pp. 161–168.

[11] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4l: Decentralized dynamic discriminative dictionary learning," *IEEE Trans. Signal and Info. Process. over Networks*, vol. 3, no. 4, pp. 728–743, 2017.

[12] A. S. Bedi and K. Rajawat, "Asynchronous incremental stochastic dual descent algorithm for network resource allocation," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2229–2244, May 2018.

[13] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1007379606734

[14] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, 2014.

[15] A. Koppel, B. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," *IEEE Trans. Signal Process.*, 2017.

[16] A. Simonetto, "Time-varying convex optimization via time-varying averaged operators," *arXiv preprint arXiv:1704.07338*, 2017.

[17] R. Dixit, A. S. Bedi, R. Tripathi, and K. Rajawat, "Online learning with inexact proximal online gradient descent algorithms," *arXiv preprint arXiv:1806.00202*, 2018.

[18] J. C. Duchi, S. Chaturapruek, and C. Ré, "Asynchronous stochastic convex optimization," *arXiv preprint arXiv:1508.00882*, 2015.

[19] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, 2011.

[20] A. H. Sayed and X. Zhao, "Asynchronous adaptive networks," *arXiv preprint arXiv:1511.09180*, 2015.

[21] A. Ribeiro, "Ergodic Stochastic Optimization Algorithms for Wireless Communication and Networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.

[22] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Adv. Neural Inf. Process. Syst.*, 2011, pp. 873–881.

[23] A. S. Bedi, A. Koppel, and K. Rajawat, "Asynchronous saddle point algorithm for stochastic optimization in heterogeneous networks," *arXiv preprint arXiv:1707.05816*, 2017.

[24] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization-part I : Algorithm and convergence analysis," *arXiv preprint arXiv:1509.02597*, 2015.

[25] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. of 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1701–1709.

[26] E. Wei and A. Ozdaglar, "On the o (1/k) convergence of asynchronous distributed alternating direction method of multipliers," *arXiv preprint arXiv:1307.8254*, 2013.

[27] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous optimization over heterogeneous networks via consensus admm," *IEEE Trans. Signal and Inf. Proces. over Netw.*, vol. 3, no. 1, pp. 114–129, March 2017.

[28] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, ser. Stanford Mathematical Studies in the Social Sciences. Stanford University Press, Stanford, Dec. 1958, vol. II.

[29] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5149–5164, 2015.

[30] A. Mokhtari, A. Koppel, and A. Ribeiro, "A class of parallel doubly stochastic algorithms for large-scale learning," *arXiv preprint arXiv:1606.04991*, 2016.

[31] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J Optimiz. Theory App.*, vol. 142, no. 1, pp. 205–228, Aug. 2009.

[32] 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.

[33] D. P. Foster and R. Vohra, "Regret in the on-line decision problem," *Games and Economic Behavior*, vol. 29, no. 1, pp. 7–35, 1999.

[34] K. Quanrud and D. Khashabi, "Online learning with adversarial delays," in *Adv. Neural Inf. Process. Syst.*, 2015, pp. 1270–1278.

[35] R. Jenatton, J. Huang, and C. Archambeau, "Adaptive algorithms for online convex optimization with long-term constraints," in *Proc. of 33rd Int. Conf. Mach. Learn.*, 2016, pp. 402–411.

[36] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[37] V. S. Chernyak, *Fundamentals of multisite radar systems: multistatic radars and multistatic radar systems*. CRC Press, 1998.

[38] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag*, vol. 22, no. 4, pp. 41–53, 2005.

[39] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *J FIELD ROBOT*, vol. 25, no. 1-2, pp. 89–110, 2008.

[40] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, 2005.

[41] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.

[42] H.-J. Shao, X.-P. Zhang, and Z. Wang, "Efficient closed-form algorithms for aoa based self-localization of sensor nodes using auxiliary variables," *IEEE Trans. Signal Process.*, vol. 62, no. 10, pp. 2580–2594, 2014.

[43] J. Diao, J. Guo, and C.-Y. Sun, "Vision-based target localization: A distributed convex optimization approach," in *36th IEEE Chinese Control Conference (CCC),*, 2017, pp. 8999–9004.