Beyond Consensus and Synchrony in Decentralized Online Optimization using Saddle Point Method

Amrit Singh Bedi*, Alec Koppel†, and Ketan Rajawat*

***Abstract*—We consider online learning problems in multi-agent systems comprised of distinct subsets of agents operating without a common time-scale. Each individual in the network is charged with minimizing the global regret, which is a sum of the instantaneous sub-optimality of each agent's actions with respect to a fixed global clairvoyant actor with access to all costs across the network for all time up to a time-horizon T. Since agents are not assumed to be of the same type, the hypothesis that all agents seek a common action is violated, and thus we instead introduce a notion of network discrepancy as a measure of how well agents coordinate their behavior while retaining distinct local behavior. Moreover, agents are not assumed to receive the sequentially arriving costs on a common time index, and thus seek to learn in an asynchronous manner. A variant of the Arrow-Hurwicz saddle point algorithm is proposed to control the growth of global regret and network discrepancy. This algorithm uses Lagrange multipliers to penalize the discrepancies between agents and leads to an implementation that relies on local operations and exchange of variables between neighbors. Decisions made with this method lead to regret whose order is $\mathcal{O}(\sqrt{T})$ and network discrepancy $\mathcal{O}(T^{3/4})$. Empirical evaluation is conducted on an asynchronously operating sensor network estimating a spatially correlated random field.**

I. INTRODUCTION

In emerging technologies such as wireless communications and networks consisting of interconnected consumer devices [1], increased sensing capabilities are leading to new theoretical challenges to classical parameter estimation. These challenges include the fact that data is persistently arriving in a sequential fashion [2], that it is physically decentralized across an interconnected network, and that the nodes of the network may correspond to disparate classes of objects (such as users and a base station) with different time-scale requirements [3]. In this work, we seek to address this class of problems through extensions of online decentralized convex optimization [4] to the case where the agents of the network may be of multiple different classes, and operate on different time-scales [5].

This work adopts the perspective of online convex optimization [6]: at each time slot, a learner selects an action (which defines a parametric statistical model, for instance), and then observes a convex cost that represents the most recent sensory information, for instance. In the face of a non-stationary target, online learning methods adopt a benchmark called *regret*, which is the time-accumulation of costs incurred by its adaptively selected sequence as compared to that of a clairvoyant actor which has access to all costs in advance up to a fixed time horizon T and acts optimally with respect to them. An online algorithm is said to be no-regret if its regret growth in T is sublinear. Distributed online learning refers to a situation in which each agent $i \in \mathcal{V}$ in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is facing a repeated decision task based on a locally observed sequentially available convex costs $f_{i,t}$, but its actual goal is to minimize a global cost, i.e. $f_t = \sum_i f_{i,t}$. Techniques that are good for distributed convex optimization, where one minimizes a global cost subject to consensus constraints $\mathbf{x}_i = \mathbf{x}_j$ for all node pairs $(i,j) \in \mathcal{E}$ [7], [8], have in most cases translated into the distributed online domain without major hurdles, as in [8], [9].

In the aforementioned works, consensus constraints are enforced in order to estimate a common decision variables with the aim to leverage parallel processing architectures to alleviate computational bottlenecks [10]. On the contrary, when distinct priors on information available at distinct group of agents are available, enforcing consensus may degrade local predictive accuracy [11]. Specifically, if the observations at each node are independent but *not* identically distributed, consensus may yield a sub-optimal solution. Motivated by situations in which the network may be heterogeneous [12], [13], and each node is faced with a sequential decision task, we focus on the setting of online multi-agent optimization with nonlinear *proximity* constraints which incentivize nearby nodes to select actions which are similar but not necessarily equal. Attempts to extend multi-agent optimization techniques to heterogeneously correlated problems have been considered in [14], [15] for special loss functions and correlation models, but the generic problem was online recently solved in [11].

Insisting on all agents to operate on a common clock creates a bottleneck for implementation in practical settings because typically nodes may be equipped with different computational capacity due to power and energy design specifications. Therefore, we attempt to extend online multi-agent optimization with nonlinear constraints to asynchronous settings [16]. Asynchrony in online optimization has taken on different forms, such as, for instance, maintaining a local Poisson clock for each agent [17] or a distribution-free generic bounded delay [18], the approach considered here.

In this paper, we extend the algorithm proposed in [11], [19] to the asynchronous settings for online multi-agent optimization with nonlinear network proximity constraints. The proposed algorithm allows the gradient to be delayed for the primal and dual updates of the saddle point method. The main technical contribution of this 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 -see [20] for proofs. The proposed algorithm is then applied to the problem of estimating a spatially correlated random field in an asynchronous sensor network.

II. PROBLEM FORMULATION

Consider a symmetric, connected, and directed network of N nodes. 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

*Department of Elect. Engg., Indian Institute of Technology Kanpur, Kanpur, Uttar Pradesh, India. Email: {amritbd, ketan}@iitk.ac.in.

*Department of ESE, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Email: {akoppel}@seas.upenn.edu.

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

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*

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

where $\tilde{\mathbf{x}}_T^* := \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T \sum_{i=1}^N f_{i,t-\tau_i(t)}(\mathbf{x})$. Observe that an agent i may minimize its contribution to $\mathbf{Reg}_T$ in (1) independently of the other agents based solely upon its local sequence of costs $f_{i,t}$, i.e., it is possibly to solve (1) 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.,

$$\mathbf{x}_{i,t} = \mathbf{x}_{j,t} \text{ for all } (i,j) \in \mathcal{E} \quad (2)$$

or by modifying the definition of regret to incorporate cross-node dependencies, as in [9]. In applications where complex correlation structures or latent dependences among the nodes' variables are present, the assumption of common estimate (considered in consensus) is clearly violated. In general, parameters of nearby nodes 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*

$$\mathrm{ND}_T := \sum_{(i,j)\in\mathcal{E}} \left[\sum_{t=1}^T h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_a tij(t)}) - \gamma_{ij}\right]_+ \quad (3)$$

where $[\cdot]_+$ denotes projection onto the non-negatives, and the comparator in (1) is revised to be the constrained optimizer

$$\mathbf{x}_T^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} \sum_{t=1}^T \sum_{i=1}^N f_{i,t-\tau_i(t)}(\mathbf{x}) \quad (4)$$
$$\text{s. t. } h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)}) - \gamma_{ij}] \leq \text{ for all } (i,j) \in \mathcal{E}.$$

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 (1) and (3), 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 [21]. 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 nodes 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 nodes and time. Ignoring neighboring observations, the minimum mean square error local estimation problem at node i can then be written in the form of (1) 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 $t - \tau_i(t)$ (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 nodes in previous time slots. The quality of these estimates can be improved using the correlated information of adjacent nodes 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

$$\mathbf{Reg}_T = \sum_{t=1}^T \sum_{i=1}^N \|\mathbf{H}_i\mathbf{x}_{i,t-\tau_i(t)} - \boldsymbol{\theta}_{i,t-\tau_i(t)}\|^2 - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T \sum_{i=1}^N \|\mathbf{H}_i\mathbf{x} - \boldsymbol{\theta}_{i,t}\|^2 \quad (5)$$

$$\mathrm{ND}_T = \sum_{t=1}^T \sum_{(i,j)\in\mathcal{E}} (1/2)\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\|^2 \leq \gamma_{ij} \quad (6)$$

The constraint $(1/2)\|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\|^2 \leq \gamma_{ij}$ makes the estimate $\mathbf{x}_i^*$ of node i close to the estimates $\mathbf{x}_j^*$ of neighboring nodes $j \in n_i$ but not so close to the estimates $\mathbf{x}_k^*$ of nonadjacent nodes $k \notin n_i$. The problem formulation in (5) is a particular case of (1) with $f_{i,t}(\mathbf{x}_{i,t}) = \|\mathbf{H}_i\mathbf{x}_{i,t} - \boldsymbol{\theta}_i\|^2$ and (3) 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 (1) and network discrepancy (3), we define the *online augmented Lagrangian* of a networked learning rule at time t as

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^N \left[f_{i,t}(\mathbf{x}_i) + \frac{1}{2}\sum_{j\in n_i} \left(\lambda_{ij}\left(h_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \gamma_{ij}\right) - \frac{\delta\epsilon_t}{2}\lambda_{ij}^2\right)\right]. \quad (7)$$

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 [22] to the Lagrangian stated in (7) proposed in [9] takes the following form

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_t - \epsilon_t \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\Big] \;, \tag{8}$$

$$\boldsymbol{\lambda}_{t+1} = \Big[\boldsymbol{\lambda}_t + \epsilon_t \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\Big]_+ \;, \tag{9}$$

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 (7) with respect to $\mathbf{x}$ and $\boldsymbol{\lambda}$, respectively. The notation $\mathcal{P}_{\mathcal{X}}(\mathbf{x})$ in (8) 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 (8) and (9) can be implemented in decentralized manner as proposed in [11, Prop. 1]. But for the implementation of primal update at node i [11, Prop. 1], each node 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 Nature. The constraint function $h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is assumed to be known at each node and only requires parameters $\mathbf{x}_j$ from the neighbor nodes to calculate the gradient $\nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and perform the update of (8). The bottleneck in the implementation of (8) and (9) 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 nodes are allowed to operate asynchronously. As in Section II, associate to each node 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 nodes within the time frame of time t. Let us denote the stacked delayed primal variable as $\mathbf{x}_{t-\boldsymbol{\tau}(t)} := \big[\mathbf{x}_{1,t-\tau_1(t)}, \mathbf{x}_{2,t-\tau_2(t)}, \cdots, \mathbf{x}_{N,t-\tau_N(t)}\big]$. For the asynchronous saddle point algorithm, the updates take the form

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_t - \epsilon_t \nabla_{\mathbf{x}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)\Big] \;, \tag{10}$$

$$\boldsymbol{\lambda}_{t+1} = \Big[\boldsymbol{\lambda}_t + \epsilon_t \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)\Big]_+ \;, \tag{11}$$

We define the k^{th} component of stacked primal gradient of the online Lagrangian in (7) evaluated at time $t - \boldsymbol{\tau}(t)$ as

$$\begin{aligned}[\nabla_{\mathbf{x}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)]_k &= \nabla_{\mathbf{x}_k} f_{k,t-\tau_k(t)}(\mathbf{x}_{k,t-\tau_k(t)}) \\ &+ \frac{1}{2}\sum_{j \in n_k} (\lambda_{kj,t} + \lambda_{jk,t}) \nabla_{\mathbf{x}_k} h_{kj}(\mathbf{x}_{k,t-\tau_k(t)}, \mathbf{x}_{j,t-\tau_j(t)}) .\end{aligned} \tag{12}$$

Further, the dual gradient of the online Lagrangian (7) at time $t - \tau(t)$ is given as

$$\begin{aligned}[\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t)]_k =& h_{kj}(\mathbf{x}_{k,t-\tau_k(t)}, \mathbf{x}_{j,t-\tau_j(t)}) - \gamma_{kj} \\ &- \delta \epsilon_t \lambda_{kj,t} \; .\end{aligned} \tag{13}$$

for all $j \in n_k$. Due to the fact that the online gradient computations in (12) and (13) associated to node k decouple from variables that depend on any other agent in the network except for those in n_k, we obtain that (10) and (11) yield a distributed algorithm, as summarized in the following proposition.

Algorithm 1 AOSP: Asynchronous Online Saddle Point

Require: initialization $\mathbf{x}_0$ and $\boldsymbol{\lambda}_0 = \mathbf{0}$, step-size ϵ_t, 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. (12).
  - 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 (14) local parameter $\mathbf{x}_{i,t}$

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

- **end loop**
- **loop in parallel** communication link $(i,j) \in \mathcal{E}$
  - Update dual variables at network link (i,j) [cf. (15)]

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

- **end loop**

end for

Proposition 1 *The primal and dual updates of* (10) *and* (11) *can be written in terms of N parallel primal with respect to local variables $\mathbf{x}_{i,t}$ as*

$$\begin{aligned}\mathbf{x}_{i,t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_{i,t} - \epsilon_t \Big(&\nabla_{\mathbf{x}_i} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) \\ &+ \frac{1}{2}\sum_{j \in n_i} (\lambda_{ij,t} + \lambda_{ji,t}) \nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\Big)\Big] \; .\end{aligned} \tag{14}$$

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

$$\lambda_{ij,t+1} = \Big[(1 - \epsilon_t^2 \delta)\lambda_{ij,t} + \epsilon_t \left(h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\right)\Big]_+ . \tag{15}$$

In contrast to the global time index required to implement (8) and (9), the implementation of (14) and (15) 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 $\mathbf{x}_{i,t}$, node 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 node 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. (14) - (15)]. Although each node 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.

IV. Regret Bounds

In this section, we establish that the proposed asynchronous online saddle point method (Algorithm 1) achieves regret bounds [cf. (1)] and bounds on the time-aggregation of constraint violation [cf. (3)] 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 node i is Lipschitz continuous with constant L_f and satisfies*

$$\|f_{i,t}(\mathbf{x})-f_{i,t}(\tilde{\mathbf{x}})\|\leq L_f\|\mathbf{x}-\tilde{\mathbf{x}}\| \ . \tag{16}$$

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

$$\|h_{ij}(\mathbf{x})-h_{ij}(\tilde{\mathbf{x}})\| \leq L_h\|\mathbf{x}-\tilde{\mathbf{x}}\|. \tag{17}$$

Assumption 2 *The set $\mathcal{X}$ onto which the primal variables are projected contains the constrained optimizer as defined in* (4).

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

$$D := \max_i \max_{\mathbf{x}\in\mathcal{X}} h_{ij}(\mathbf{x},\mathbf{x}_j) \leq L_g R \quad for\ all \quad j\in n_i. \tag{18}$$

Assumption 4 (Delay model) *The delay introduced at each node 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 (4). 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 [23]. 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 [11, eq. 19, 20] holds directly, since they are established for arbitrary $\mathbf{x}$ and $\boldsymbol{\lambda}$:

$$[\|\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda})\|^2]\leq(N+M)L^2(1+\|\boldsymbol{\lambda}\|^2) \tag{19}$$

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

$$\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda})\|^2\leq 2ML_g^2R^2+2\delta^2\epsilon^2\|\boldsymbol{\lambda}\|^2 \tag{20}$$

The idea of representing gradient bounds in terms of dual variable and then using 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 [11].

Remark 1 One may interpret online convex optimization as a repeated adversarial game, as is common in game theory [24]. 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 [25]. 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* (8) *and* (9) *with step-size ϵ, it holds that*

$$\begin{aligned}&\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda})-\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x},\boldsymbol{\lambda}_t)\\&\leq\frac{1}{2\epsilon_t}\left(\|\mathbf{x}_t-\mathbf{x}\|^2-\|\mathbf{x}_{t+1}-\mathbf{x}\|^2+\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}\|^2\right)\\&+\frac{\epsilon_t}{2}\Big(\|\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\|^2+\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)\|^2\Big)\\&+\nabla_{\mathbf{x}}\mathcal{L}_{t-\boldsymbol{\tau}(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)},\boldsymbol{\lambda}_t)^T(\mathbf{x}_{t-\boldsymbol{\tau}(t)}-\mathbf{x}_t)\end{aligned} \tag{21}$$

Variants of this lemma apppear in past works on primal-dual methods for constrained online learning [11], [23], [26], and make use of the convex-concave structure of the Lagrangian in (7). 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 (21) is identical to [11, 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.* (1)*] 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:*

$$\mathbf{Reg}_T\leq\mathcal{O}(\sqrt{T}). \tag{22}$$

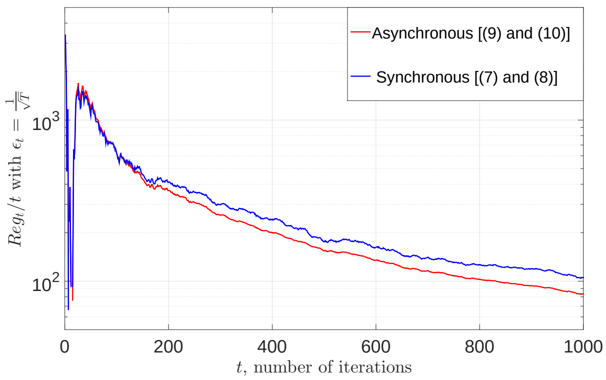
Moreover, the delayed network discrepancy (constraint violation) of the algorithm grows sublinearly in final time T as

$$ND_T\leq\mathcal{O}(T^{3/4}). \tag{23}$$

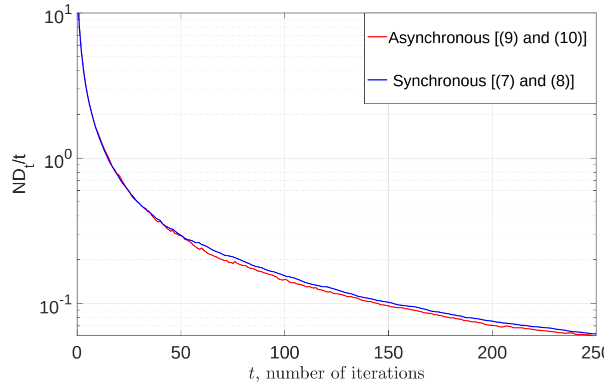
Theorem 1 establishes that the sequence of actions chosen by each agent in the network according to Algorithm 1 yield both a sublinear growth in the delayed regret and network discrepancy (constraint violation). These results are comparable to existing results for the setting in which the network operates on a synchronized clock [11], 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.

V. Numerical tests

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, nodes which are close to each other will have the same field while the distant nodes 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 (1) and satisfy the neighborhood constraints of (3) in long run. The updates for Algorithm 1 take the form



(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.

$$\mathbf{x}_{i,t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_{i,t} - \epsilon_t\Big(2(\mathbf{H}_i\mathbf{x}_{i,t-\tau_i(t)} - \boldsymbol{\theta}_{i,t-\tau_i,t-\tau_i(t)}) + \frac{1}{2}\sum_{j\in n_i}(\lambda_{ij,t}+\lambda_{ji,t})(\big\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\big\|^2 - \gamma_{ij})\Big)\Big],$$
$$\lambda_{ij,t+1} = \Big[(1-\epsilon_t^2\delta)\lambda_{ij,t} + \frac{\epsilon_t}{2}[\big\|\mathbf{x}_{i,t-\tau_i(t)} - \mathbf{x}_{j,t-\tau_j(t)}\big\|^2 - \gamma_{ij}]\Big]_+. \tag{24}$$

For the simulations purpose, we consider a wireless sensor network of $N = 20$ nodes for the scalar field estimation ($p = q = 1$) where nodes 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 nodes. 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 ([11]) 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 $T \to \infty$.

References

[1] 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.

[2] A. Koppel, J. Fink, G. Warnell, E. Stump, and A. Ribeiro, "Online learning for characterizing unknown environments in ground robotic vehicle models," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 626–633.

[3] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug 2006.

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

[5] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.

[6] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. on Machine Learning*, 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 *American Control Conference (ACC), 2012*. IEEE, 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 *Advances in neural information processing systems*, 2008, pp. 161–168.

[11] A. Koppel, B. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," *IEEE Transactions on Signal Processing*, 2017.

[12] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4l: Decentralized dynamic discriminative dictionary learning," *IEEE Trans. Signal and Info. Process. over Networks*, vol. (submitted), June 2017, available at http://www.seas.upenn.edu/ aribeiro/wiki.

[13] A. S. Bedi and K. Rajawat, "Asynchronous incremental stochastic dual descent algorithm for network resource allocation," *arXiv preprint arXiv:1702.08290*, 2017.

[14] 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

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

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

[17] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 772–790, 2011.

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

[19] 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.

[20] A. S. Bedi, A. Koppel, and K. Rajawat, "Beyond consensus and synchrony in decentralized online optimization using saddle point method," *IIT-Kanpur/UPenn Technical Report*, 2017. [Online]. Available: https://fling.seas.upenn.edu/~akoppel/assets/papers/asyn_saddlept_report.pdf

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

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

[23] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[24] 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.

[25] K. Quanrud and D. Khashabi, "Online learning with adversarial delays," in *Advances in Neural Information Processing Systems*, 2015, pp. 1270–1278.

[26] R. Jenatton, J. Huang, and C. Archambeau, "Adaptive algorithms for online convex optimization with long-term constraints," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 402–411.