Asynchronous Saddle Point Method: Interference Management Through Pricing

Amrit Singh Bedi, Alec Koppel, and Ketan Rajawat

***Abstract*— This paper considers a wireless network in which each node is charged with minimizing the global objective function, which is an average of sum of the statistical average loss function of each node (agent) in the network. Since agents are not assumed to observe data from identical distributions, the hypothesis that all agents seek a common action is violated, and thus the hypothesis upon which consensus constraints are formulated is violated. Thus, we consider nonlinear network proximity constraints which incentivize nearby nodes to make decisions which are close to one another but not necessarily coincide. Moreover, agents are not assumed to receive their sequentially arriving observations on a common time index, and thus seek to learn in an asynchronous manner. An asynchronous stochastic variant of the Arrow-Hurwicz saddle point method is proposed to solve this problem which operates by alternating primal stochastic descent steps and Lagrange multiplier updates which penalize the discrepancies between agents. Our main result establishes that the proposed method yields convergence in expectation both in terms of the primal sub-optimality and constraint violation to radii of sizes $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(T^{3/4})$, respectively. Empirical evaluation on an asynchronously operating wireless network that manages user channel interference through an adaptive communications pricing mechanism demonstrates that our theoretical results translate well to practice.**

I. Introduction

In emerging technologies such as wireless communications and networks consisting of interconnected consumer devices [2], 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 [3], 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 [4]. In this work, we seek to address this class of problems through extensions of online decentralized convex optimization [5] to the case where the agents of the network may be of multiple different classes, and operate on different time-scales [6].

To address the fact that we seek iterative tools for streaming data, we consider stochastic optimization problems [7], [8]. In this setting, the objective function $\mathbb{E}[f(\mathbf{x},\boldsymbol{\theta})]$ is an expectation over a set of functions parameterized by a random variable $\boldsymbol{\theta}$. The objective function encodes, for example, the quality of a statistical parameter estimate. Through a sequence of realizations of a random variable $\boldsymbol{\theta}_t$, we seek to find parameters that are good with respect to the average objective. The classical method to address this problem is stochastic gradient descent (SGD), which involves descending along the negative of the stochastic gradient in lieu of the true gradient to circumvent the computation of an infinite complexity expectation [9], [10]. SGD forms the foundation of tools considered in this paper for asynchronous multi-agent settings.

In the literature [11]–[13], consensus constraints are enforced in order to estimate a common decision variable while leveraging parallel processing architectures to achieve computational speedup [14]. On the other hand, in situations where distinct priors on information received at distinct subsets of agents are available, then requiring the network to reach a common decision may degrade local predictive accuracy [15]. Contrariwise, when unique priors on information available at distinct group of agents are available as in sensor [15] or robotic [16] networks, enforcing consensus degrades the statistical accuracy of each agent's estimate [15]. Specifically, if the observations at each node are independent but *not* identically distributed, consensus may yield a sub-optimal solution. Motivated by heterogeneous networked settings [16], [17] where each node observes a unique local data stream, we focus on the setting of multi-agent stochastic optimization with nonlinear *proximity* constraints which incentivize nearby nodes to select estimates which are similar but not necessarily equal.

In the setting of nonlinear constraints, penalty methods such as distributed gradient descent do not apply [11], and dual or proximal methods require a nonlinear minimization in an inner-loop of the algorithm [18]. Therefore, we adopt a method which hinges on Lagrange duality that avoids costly argmin computations in the algorithm inner-loop, namely primal-dual method [19], also referred to as saddle point method. Alternative attempts to extend multi-agent optimization techniques to heterogeneously correlated problems have been considered in [20] for special loss functions and correlation models, but the generic problem was online recently solved in [15] with a stochastic variant of the saddle point method.

However, 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, as well as a difference in the sparsity of each agent's data stream. Therefore, we attempt to extend multi-agent stochastic optimization with nonlinear

A. S. Bedi and K. Rajawat are with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, India 208016. {amrit, ketan}@iitk.ac.in

A. Koppel is with U.S. Army Research Laboratory, Adelphi, MD, USA. {alec.e.koppel.civ@mail.mil}

A technical report related to this work is available as a preprint at [1].

constraints to asynchronous settings [21]. Asynchrony in online optimization has taken on different forms, such as, for instance, maintaining a local Poisson clock for each agent [22] or a distribution-free generic bounded delay [6], the approach considered here.

In this paper, we extend the primal-dual method of [15], [19] for multi-agent stochastic optimization problems with nonlinear network proximity constraints to asynchronous settings. The proposed algorithm allows the gradient to be delayed for the primal and dual updates of the saddle point method. This feature of allowing delays makes it possible for each node to skip gradient calculation or defer it in order to save some energy, which is an important characteristic of energy harvesting networks. The main technical contribution of this paper is to provide mean convergent results for both the global primal cost and constraint violation [see [23] for proofs], establishing that the Lyapunov stability results of [15] translate successfully to asynchronous computing architectures increasingly important in intelligent communication systems and wireless communication networks with mobile users. Empirical evaluation on an asynchronously operating cellular network that manages cross-tier interference through an adaptive pricing mechanism demonstrates that our theoretical results translates well to practice.

II. Multi-Agent Optimization without Consensus

We consider agents i of a symmetric, connected, and directed network $\mathcal{G} = (V, \mathcal{E})$ with $|V| = N$ nodes and $|\mathcal{E}| = M$ edges. Each agent is associated with a (non-strongly) convex loss function $f^i : \mathcal{X} \times \Theta_i \to \mathbb{R}$ that is parameterized by a p-dimensional decision variable $\mathbf{x}^i \in \mathcal{X} \subset \mathbb{R}^p$ and a random vector $\boldsymbol{\theta}_i \in \Theta_i \subset \mathbb{R}^q$. The functions $f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)$ for different $\boldsymbol{\theta}^i$ encodes the merit of a particular linear statistical model $\mathbf{x}^i$, for instance, and the random vector $\boldsymbol{\theta}$ may be particularized to a random pair $\boldsymbol{\theta} = (\mathbf{z}, \mathbf{y})$. In this setting, the random pair corresponds to feature vectors $\mathbf{z}$ together with their binary labels $\mathbf{y} \in \{-1, 1\}$ or real values $\mathbf{y} \in \mathbb{R}$, for the respective problems of classification or regression. Here we address the case that the local random vector $\boldsymbol{\theta}^i$ represents data which is revealed to node i *sequentially* as realizations $\boldsymbol{\theta}^i_t$ at time t, and agents would like to process this information on the fly. Mathematically this is equivalent to the case where the total number of samples T revealed to agent i is not necessarily finite. A possible goal for agent i is the solution of the local expected risk minimization problem,

$$\mathbf{x}^{\mathrm{L}}(i) := \underset{\mathbf{x}^i \in \mathcal{X}}{\operatorname{argmin}} \, F^i(\mathbf{x}^i) := \underset{\mathbf{x}^i \in \mathbb{R}^p}{\operatorname{argmin}} \, \mathbb{E}_{\boldsymbol{\theta}^i}[f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)] \, . \quad (1)$$

where we define $F^i(\mathbf{x}^i) := \mathbb{E}_{\boldsymbol{\theta}^i}[f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)]$ as the local average function at node i. We also restrict $\mathcal{X}$ to be a compact convex subset of $\mathbb{R}^p$ associated with the p-dimensional parameter vector of agent i. By stacking the problem (1) across the entire network, we obtain the equivalent problem

$$\mathbf{x}^{\mathrm{L}} = \underset{\mathbf{x} \in \mathcal{X}^N}{\operatorname{argmin}} \, F(\mathbf{x}) := \underset{\mathbf{x} \in \mathcal{X}^N}{\operatorname{argmin}} \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\theta}^i}[f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)] \, . \quad (2)$$

where we define the stacked vector $\mathbf{x} = [\mathbf{x}^1, \dots, \mathbf{x}^N] \in \mathcal{X}^N \subset \mathbb{R}^{Np}$, and the global cost function $F(\mathbf{x}) := \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\theta}^i}[f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)]$. We define the global instantaneous cost similarly: $f(\mathbf{x}, \boldsymbol{\theta}) = \sum_i f^i(\mathbf{x}^i, \boldsymbol{\theta}^i)$.

Most distributed optimization works, for instance, consensus optimization, hypothesize that all agents seek to learn the common parameters $\mathbf{x}^i$ for all $i \in V$, i.e., $\mathbf{x}^i = \mathbf{x}^j$, for all $j \in n_i$. where n_i denotes the neighborhood of agent i. Making all agents variables equal only makes sense when agents observe information drawn from a common distribution, which is the case for industrial-scale machine learning, but is predominantly not the case for sensor [15] and robotic networks [16]. As noted in [15], generally, nearby nodes observe similar but not identical information, and thus to incentivize collaboration without enforcing consensus, we introduce a convex local proximity function with real-valued range of the form $h^{ij}(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}^i, \boldsymbol{\theta}^j)$ that depends on the observations of neighboring agents and a tolerance $\gamma_{ij} \geq 0$. These stochastic constraints then couple the decisions of agent i to those of its neighbors $j \in n_i$ as the solution of the constrained stochastic program

$$\begin{aligned} \mathbf{x}^* \in \; & \underset{\mathbf{x} \in \mathcal{X}^N}{\operatorname{argmin}} \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\theta}_i}[f^i(\mathbf{x}_i, \boldsymbol{\theta}_i)] \\ & \text{s.t.} \quad \mathbb{E}_{\boldsymbol{\theta}^i, \boldsymbol{\theta}^j}\left[h^{ij}(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\right] \leq \gamma_{ij}, \text{ for all } j \in n_i. \end{aligned} \quad (3)$$

Examples of the constraint in the above formulation include approximate consensus constraints $\left\|\mathbf{x}^i - \mathbf{x}^j\right\| \leq \gamma_{ij}$, quality of service $\textbf{SINR}(\mathbf{x}^i, \mathbf{x}^j) \geq \gamma_{ij}$ where SINR is the signal-to-interference-plus-noise function, relative entropy $D(\mathbf{x}^i \parallel \mathbf{x}^j) \leq \gamma_{ij}$, or budget $\gamma_{ij}^{\min} \leq x^i + x^j \leq \gamma_{ij}^{\max}$ constraints. In this work, we seek decentralized online solutions to the constrained problem (3) *without* the assumption that agents operate on a common time index, motivated by the fact that asynchronous computing settings are common in large distributed wireless networks. In the next section we turn to developing an algorithmic solution that meets these criteria.

III. Asynchronous Saddle Point Method

Methods based upon distributed gradient descent [24] are inapplicable to settings with nonlinear constraints. On the other hand, the dual methods proposed in [25], [26] require a nonlinear minimization computation at each algorithm iteration, and thus is impractically costly. Therefore, in this section we develop a computationally light weight method based on primal-dual method that may operate in decentralized online asynchronous settings with constant learning rates that are better suited to changing environments.

To develop a decentralized, online and asynchronous algorithm, we begin by considering the approximate Lagrangian relaxation of (3) stated as

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \Bigg[\mathbb{E}\Bigg[& f^i(\mathbf{x}^i, \boldsymbol{\theta}^i) \\ & + \sum_{j \in n_i} \lambda^{ij} \left(h^{ij}\left(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}^i, \boldsymbol{\theta}^j\right) - \gamma_{ij} \right) - \frac{\delta \epsilon}{2} (\lambda^{ij})^2 \Bigg] \Bigg], \end{aligned} \quad (4)$$

where λ^{ij} is a non-negative Lagrange multiplier associated with the non-linear constraint in (3). Here, $\boldsymbol{\lambda}$ defines the collection of all dual variables λ^{ij} into a single vector $\boldsymbol{\lambda}$. Observe that (4) is not the standard Lagrangian of the (3) but instead an augmented Lagrangian due to the presence of the term $-(\delta\epsilon/2)(\lambda^{ij})^2$. This terms acts like a regularizer on the dual variable with associated parameters δ and ϵ that allow us to control the accumulation of constraint violation of the algorithm over time. The stochastic approximation of the augmented Lagrangian evaluated at observed realizations $\boldsymbol{\theta}_t^i$ of the random vectors $\boldsymbol{\theta}^i$ for each $i \in \mathcal{V}$:

$$\hat{\mathcal{L}}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \Big[f^i(\mathbf{x}^i, \boldsymbol{\theta}_t^i) + \sum_{j \in n_i} \lambda^{ij} \Big(h^{ij}\left(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}_t^i, \boldsymbol{\theta}_t^j\right) - \gamma_{ij} \Big) - \frac{\delta\epsilon}{2} (\lambda^{ij})^2 \Big]. \tag{5}$$

The stochastic saddle point method applied to the stochastic Lagrangian (5) takes the following form similar to [15] as

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \Big[\mathbf{x}_t - \epsilon \nabla_{\mathbf{x}} \hat{\mathcal{L}}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \Big], \tag{6}$$

$$\boldsymbol{\lambda}_{t+1} = \Big[\boldsymbol{\lambda}_t + \epsilon \nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \Big]_+, \tag{7}$$

where $\nabla_{\mathbf{x}} \hat{\mathcal{L}}(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ and $\nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}(\mathbf{x}_t, \boldsymbol{\lambda}_t)$, are the primal and dual stochastic gradients[1] of the augmented Lagrangian with respect to $\mathbf{x}$ and $\boldsymbol{\lambda}$, respectively. These are not the actual gradients of (4) rather are stochastic gradients calculated at the current realization of the random vectors $\boldsymbol{\theta}_t^i$ for all i. The component wise projection for a vector $\mathbf{x}$ on to the given compact set $\mathcal{X}$ is here denoted by $\mathcal{P}_{\mathcal{X}}(\mathbf{x})$. Similarity, $[\cdot]_+$ represents the component wise projection on to the positive orthant $\mathbb{R}_+^M$.

Observe that the implementation of the [15, Algorithm 1], which is defined by (6)-(7), the update of primal variable at node i requires the current gradient of its local objective function $\nabla_{\mathbf{x}^i} f^i(\mathbf{x}_t^i, \boldsymbol{\theta}_t^i)$ and current gradient from all the neighbors $j \in n_i$ of node i as $\nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}_t^i, \mathbf{x}_t^j, \boldsymbol{\theta}_t^i, \boldsymbol{\theta}_t^j\right)$. This availability of the gradients from the neighbors on a common time-scale is a strong assumption that insists upon perfect communications, similarity of computational capability of distinct nodes, and similar levels of sparsity among agents' data that are oftentimes violated in large heterogeneous systems.

In particular, to ameliorate the computational bottleneck associated with synchronized computation and communication rounds among the nodes, we consider an *asynchronous processing architecture* with delays introduced by each node. These delays take the form of random delays on the gradients which are used for the algorithm updates. We associate to each node i in the network a time-dependent delay $\tau_i(t)$ for its stochastic gradient. Since the gradient corresponding to node i are delayed by $\tau_i(t)$, it implies that the received gradient corresponds to $t - \tau_i(t)$ time slot which we denote as $[t]_i$. Rather than waiting for the current gradient at time t, agent i instead uses the delayed gradient from the neighboring nodes at time $[t]_j$ for its update at time $[t]_i$. This leads to the following asynchronous primal update for stochastic online saddle point algorithm at each node i

[1]Note that these may be subgradients if the objective/ constraint functions are non-differentiable. The proof is extendable to non-differentiable cases.

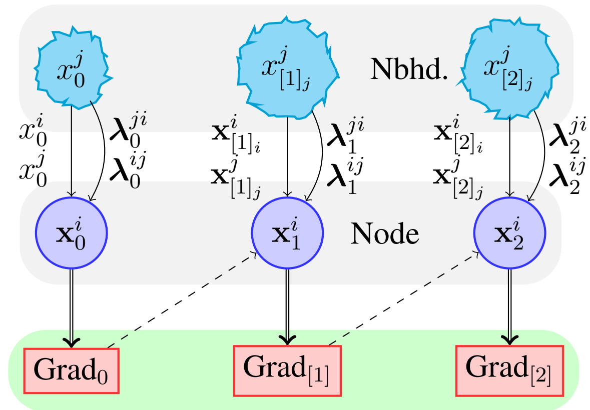


Fig. 1: Message passing in proposed algorithm. The red block represents the gradients required as mentioned in step 4 of the Algorithm 1 which depends upon the realizations $\boldsymbol{\theta}_{[t]_i}^i$ and $\boldsymbol{\theta}_{[t]_j}^j$ at time step t.

$$\mathbf{x}_{t+1}^i = \mathcal{P}_{\mathcal{X}} \Big[\mathbf{x}_t^i - \epsilon \Big(\nabla_{\mathbf{x}^i} f^i(\mathbf{x}_{[t]_i}^i, \boldsymbol{\theta}_{[t]_i}^i) + \sum_{j \in n_i} \left(\lambda_t^{ij} + \lambda_t^{ji} \right) \nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}_{[t]_i}^i, \mathbf{x}_{[t]_j}^j, \boldsymbol{\theta}_{[t]_i}^i, \boldsymbol{\theta}_{[t]_j}^j \right) \Big) \Big]. \tag{8}$$

Likewise, the dual update for each edge $(i, j) \in \mathcal{E}$ is

$$\lambda_{t+1}^{ij} = \Big[(1 - \epsilon^2\delta)\lambda_t^{ij} + \epsilon \left(h^{ij}\Big(\mathbf{x}_{[t]_i}^i, \mathbf{x}_{[t]_j}^j, \boldsymbol{\theta}_{[t]_i}^i, \boldsymbol{\theta}_{[t]_j}^j \Big) \right) \Big]_+. \tag{9}$$

Note that to perform the asynchronous primal updates at node i in (8), delayed primal gradients $\nabla_{\mathbf{x}^i} f^i(\mathbf{x}_{[t]_i}^i, \boldsymbol{\theta}_{[t]_i}^i)$ and $\nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}_{[t]_i}^i, \mathbf{x}_{[t]_j}^j, \boldsymbol{\theta}_{[t]_i}^i, \boldsymbol{\theta}_{[t]_j}^j \right)$ are utilized. Similarity, the dual delayed gradient is utilized for the update in (9). For the consistency in the algorithm implementation, it is assumed that at each node i, only the recent received copy of the gradient is kept and used for the update. Equivalently, this condition can be mentioned as $[t]_i \geq [t-1]_i$ which implies that $\tau_i(t) \leq \tau_i(t-1)+1$. For brevity, we will use the notation $[\mathbf{t}]$ as a collective notation for all the delayed time instances as $[\mathbf{t}] := [[t]_1; \cdots; [t]_N]$. The asynchronous algorithm is summarized in Algorithm 1. The practical implementation of the proposed asynchronous algorithm is explanied with the help of diagram in Fig.1. As described in figure, each node receives delayed parameters, gradients and carries out the updates accordingly. The convergence guarantees for the proposed algorithm are shown to hold as $t - [t]_i \leq \tau$ is finite (cf. **A6**) for all i and t. The convergence results presented in [15] can be obtained as a special case with $\tau_i(t) = 0$ from the results developed in this paper. This shows the generalization of the existing results in literature.

IV. Convergence Results

In this section, we establish convergence in expectation of the proposed asynchronous technique in (8)-(9) to a primal-dual optimal pair of the problem formulated in (3)

Algorithm 1 ASSP: Asynchronous Stochastic 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 $i \in \mathcal{V}$

    Send dual vars. $\boldsymbol{\lambda}_{ij,t}$ to nbhd. $j \in n_i$

    Observe delayed gradients $\nabla_{\mathbf{x}^i} f^i(\mathbf{x}^i_{[t]_i}, \boldsymbol{\theta}^i_{[t]_i})$, $\nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}^i_{[t]_i}, \mathbf{x}^j_{[t]_j}, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}\right)$ and constraint function $h^{ij}\left(\mathbf{x}^i_{[t]_i}, \mathbf{x}^j_{[t]_j}, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}\right)$.

    Update $\mathbf{x}^i_{t+1}$ using (8) local parameter $\mathbf{x}^i_t$

$$\mathbf{x}^i_{t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}^i_t - \epsilon\Big(\nabla_{\mathbf{x}^i} f^i(\mathbf{x}^i_{[t]_i}, \boldsymbol{\theta}^i_{[t]_i}) + \sum_{j \in n_i}\Big(\lambda^{ij}_t + \lambda^{ji}_t\Big)\nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}^i_{[t]_i}, \mathbf{x}^j_{[t]_j}, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}\right)\Big)\Big]$$

    Update dual variables at each agent i [cf. (9)]

$$\lambda^{ij}_{t+1} = \Big[(1 - \epsilon^2\delta)\lambda^{ij}_t + \epsilon\left(h^{ij}\left(\mathbf{x}^i_{[t]_i}, \mathbf{x}^j_{[t]_j}, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}\right)\right)\Big]_+.$$

  end loop

end for

when constant step sizes are used. It is shown that the time-average primal objective function $F(\mathbf{x}_t)$ converges to the optimal value $F(\mathbf{x}^\star)$ at a rate of $\mathcal{O}(1/\sqrt{T})$. Similarly, the time-average aggregated delayed constraint violation over network vanishes with the order of $\mathcal{O}(T^{-1/4})$, both in expectation, where T is the final iteration index. To prove convergence, some assumptions related to the system model and parameters are required which we state as follows.

[**A1**] (*Network connectivity*) The network $\mathcal{G}$ is symmetric and connected with diameter D.

[**A2**] (*Existence of Optima*) The set of primal-dual optimal pairs $\mathcal{X}^* \times \boldsymbol{\Lambda}^*$ of the constrained problem (3) has non-empty intersection with the feasible domain $\mathcal{X}^N \times \mathbb{R}^M_+$.

[**A3**](*Stochastic Gradient Variance*) The instantaneous objective and constraints for all i and t satisfy $\mathbb{E}\left\|\nabla_{\mathbf{x}^i} f^i(\mathbf{x}^i, \boldsymbol{\theta}^i_t)\right\|^2 \leq \sigma^2_f$ and $\mathbb{E}\left\|\nabla_{\mathbf{x}^i} h^{ij}\left(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}\right)\right\|^2 \leq \sigma^2_h$ which states that the second moment of the norm of objective and constraint function gradients are bounded above.

[**A4**] (*Constraint Function Variance*) For the instantaneous constrain function for all pairs $(i, j) \in \mathcal{E}$ and t over the compact set $\mathcal{X}$, it holds that

$$\max_{(\mathbf{x}^i, \mathbf{x}^j) \in \mathcal{X}} \mathbb{E}\left[\left(h^{ij}\left(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}^i_t, \boldsymbol{\theta}^j_t\right)^2\right)\right] \leq \sigma^2_{\boldsymbol{\lambda}} \tag{10}$$

which implies that the maximum value the constraint function can take is bounded by some finite scalar $\sigma^2_{\boldsymbol{\lambda}}$ in expectation.

[**A5**] (Lipschitz continuity) The expected objective function defined in (2) satisfies $\|F(\mathbf{x}) - F(\mathbf{y})\| \leq L_f \|\mathbf{x} - \mathbf{y}\|$. for any $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{Np}$.

[**A6**] (Bounded Delay) The delay $\tau_i(t)$ associated with each node i is upper bounded: $\tau_i(t) \leq \tau$ for some $\tau < \infty$.

A1 ensures that the graph is connected and the rate at which information diffuses across the network is finite. This condition is standard in distributed algorithms [27]. Moreover, **A2** is a Slater's condition and appeared in [28]. **A3** assumes an upper bound on the mean norm of the primal and dual stochastic gradients, which is crucial to developing the gradient bounds for the Lagrangian used in the proof. **A4** yields an upper bound on the maximum possible value of the constraint function in expectation similar to that of [29], and is guaranteed to hold when $\mathcal{X}$ is compact and h^{ij} is Lipschitz. **A5** is related to the Lipschitz continuity of the primal objective function. **A6** ensures that the delay is always bounded by τ, which holds in most wireless communications problems and autonomous multi-agent networks [30].

Next, in Theorem 1, we present the main convergence results [see [23] for proofs] in terms of primal objective optimality gap and aggregated network constraint violation.

Theorem 1 *Under the Assumptions* ***A1*** *-* ***A6*** *, denote* $(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ *as the sequence of primal-dual variables generated by Algorithm 1 [cf.* (8)-(9)*]. When the algorithm is run for* T *total iterations with constant step size* $\epsilon = 1/\sqrt{T}$*, the average time aggregation of the sub-optimality sequence* $\mathbb{E}\left[F(\mathbf{x}_t) - F(\mathbf{x}^*)\right]$*, with* $\mathbf{x}^*$ *defined by* (3)*, grows sublinearly with* T *as*

$$\sum_{t=1}^{T} \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)] \leq \mathcal{O}(\sqrt{T}). \tag{11}$$

Likewise, the delayed time aggregation of the average constrain violation also grows sublinearly in T *as*

$$\sum_{(i,j) \in \mathcal{E}} \mathbb{E}\left[\Big[\sum_{t=1}^{T}\Big(h^{ij}(\mathbf{x}^i_{[t]_i}, \mathbf{x}^j_{[t]_j}, \boldsymbol{\theta}^i_{[t]_i}, \boldsymbol{\theta}^j_{[t]_j}) - \gamma_{ij}\Big)\Big]_+\right] \leq \mathcal{O}(T^{3/4}). \tag{12}$$

Theorem 1 presents the behavior of the Algorithm 1 when run for T total iterations with a constant step size. Specifically, the average aggregated objective function error sequence is upper bounded by a constant time $\sqrt{T}$ sequence. This establishes that the expected value of the objective function $\mathbb{E}\left[F(\mathbf{x}_t)\right]$ will become closer to the optimal $F(\mathbf{x}^\star)$ for larger T. Similar behavior is shown by the average delayed aggregated network constraint violation term. The key innovation establishing this result is the bound on the directional error caused by asynchrony. We achieve this by bounding it in terms of the gradient norms and dual variable $\boldsymbol{\lambda}$ and exploiting the fact that the delay is at-worst bounded.

These results are similar to those for the unconstrained convex optimization problems with sub-gradient descent approach and constant step size. For most of the algorithms in this context [31, Section 2.2, eqn. 2.19], or [32, Section 4], convergence to the neighborhood of size $\mathcal{O}(\epsilon T)$ is well known. For such algorithms, the primal sub-optimality is shown of the order $\mathcal{O}(\epsilon T)$ is shown and the radius of suboptimality is optimally controlled by selecting $\epsilon = 1/\sqrt{T}$ [33]. The bound $\mathcal{O}(T^{3/4})$ on the constraint violation aggregation is comparable to existing results for synchronized multi-agent online learning [29] and stochastic approximation [15].

V. Interference management through pricing

The rising number of cellular users has fueled the increase in infrastructure spending by cellular operators towards better serving densely populated areas. In order to circumvent the near-absolute limits on spectrum availability, the current and future generations rely heavily on frequency reuse via small cells and associated interference management techniques [34], [35]. This work builds upon the pricing-based interference management framework proposed in [35]. We consider heterogeneous networks with multiple autonomous small cell users. Under heavy load situations, the macro base station (MBS) may assign the same operating frequency to multiple but geographically disparate small cell base stations (SCBS) and macro cell users (MU). The base station regulates the resulting cross-tier interference (from SCBS to MU) by penalizing the received interference power at the MUs. Consequently, the SCBSs coordinate among themselves and employ power control to limit their interference at the MUs. This section considers the pricing problem from the perspective of the BS that seeks to maximize its revenue.

A. Problem formulation

Consider the network depicted in Fig. 2, consisting of a MBS serving M MU users and N SCBSs [35]. Each MU is assigned a unique subchannel, indexed by $i \in \{1, \ldots, M\}$. At times of high traffic, the BS also allows the SCBSs to use these M subchannels, so that the n-th SCBS may serve $K_n \leq M$ SCUs. In other words, at each time slot, a particular subchannel is used by MU i and a non-empty set of SCBSs $\mathcal{N}_i \subset \{1, \ldots, N\}$. Denoting the channel gains between the n-th SCBS and m-th MU by g_{ni}, it follows that the total interference at the MU m is given by $I_i := \sum_{n \in \mathcal{N}_i} g_{ni} p_n^i$ where p_n^i is the transmit power of SCBS n while using subchannel assigned to MU m. The BS regulates this cross-tier interference by imposing a penalty x_n^i on the SCBSs $n \in \mathcal{N}_i$. The total revenue generated by the BS is therefore given by

$$\sum_{i=1}^{M} \sum_{n \in \mathcal{N}_i} x_n^i g_{ni} p_n^i \tag{13}$$

which the BS seeks to maximize. The BS also adheres to the constraint that the total penalty imposed on each SCBS is within certain limit, i.e.,

$$C_{\min} \leq \sum_{i: n \in \mathcal{N}_i} x_n^i \leq C_{\max}. \tag{14}$$

The limit on the maximum and minimum penalties can also be viewed as a means for BS to be fair to all small cell operators. The power allocation at the SCBSs is governed by their local transmission costs, denoted by c per unit transmit power, and the interference prices levied by the BS. As in [35], each SCBS solves a penalized rate minimization subproblem, resulting in the power allocation

$$p_n^i = \left((W/(c\mu_n + \nu_n x_n^i)) - (1/h_n^i) \right)_+ \tag{15}$$

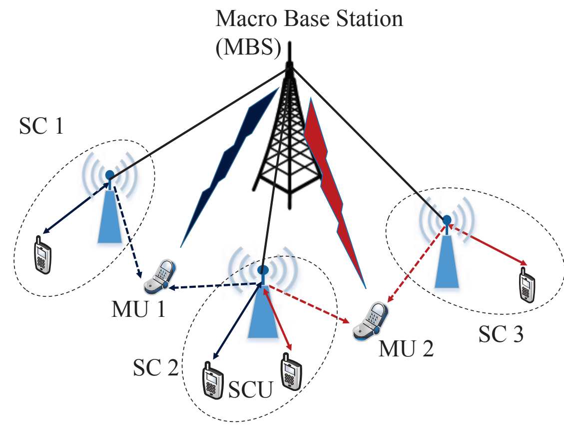


Fig. 2: Heterogeneous cellular network with one MBS, two MUs, and three SCBSs with each serving one, two, and one SCU, respectively.

for all $n \in \mathcal{N}_i$ and $1 \leq m \leq M$. Here, μ_n and ν_n represent SCBS-specific parameters used to trade-off the achieved sum rate against the transmission costs and W is the bandwidth per subcarrier. Finally, the channel gains g_{ni} and h_n^i are not known in advance, so the BS seeks to solve the following stochastic optimization problem:

$$\max_{\{x_n^i\}} \sum_{i=1}^{M} \sum_{n \in \mathcal{N}_i} \mathbb{E}\left[x_n^i g_{ni} p_n^i(x_n^i, h_n^i)\right] \tag{16a}$$

$$\text{s. t.} \sum_{n \in \mathcal{N}_i} \mathbb{E}\left[g_{ni} p_n^i(x_n^i, h_n^i)\right] \leq \gamma_i \quad 1 \leq i \leq M \tag{16b}$$

$$C_{\min} \leq \sum_{i: n \in \mathcal{N}_i} x_n^i \leq C_{\max} \quad 1 \leq n \leq N \tag{16c}$$

Here, observe that the interference constraint is required to hold only on an average, while the limits on the interference penalties are imposed at every time slot.

B. Solution using stochastic saddle point algorithm

It can be seen that the stochastic optimization problem formulated in (16) is of the form required in (3) with $\mathcal{X}$ capturing the constraint in (16c). Since the random variables h_n^i and $g_{ni,t}$ have bounded moments, the assumptions in Section IV can be readily verified. Further, the saddle point method may be applied for solving (16). To do so, we use the preceding definition of the power function p_n^i defined as in (15), and associating dual variable λ^i with the i-th constraint in (16), the stochastic augmented Lagrangian is given by

$$\hat{\mathcal{L}}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{M} \sum_{n \in \mathcal{N}_i} x_n^i g_{ni,t} p_n^i(x_n^i, h_{n,t}^i) + \sum_{i=1}^{M} \lambda^i [\gamma_i - \sum_{n \in \mathcal{N}_i} g_{ni,t} p_n^i(x_n^i, h_{n,t}^i)] - \frac{\delta \epsilon}{2} \|\boldsymbol{\lambda}\|^2 \tag{17}$$

where $\mathbf{x}$ collects the variables $\{x_n^i\}_{i=1, n \in n_i}^M$ and $\boldsymbol{\lambda}$ collects the dual variables $\{\lambda^i\}_{i=1}^M$.

The asynchronous saddle point method for pricing-based interference management in wireless systems then takes the form of Algorithm 2 with the modified projection defined as

$$P_{\mathcal{X}_n}(\mathbf{u}) := \min_{y} \|\mathbf{y} - \mathbf{u}\| \quad \text{s. t.} \quad C_{\min} \le \langle \mathbf{1}, \mathbf{y} \rangle \le C_{\max}. \tag{18}$$

In order to get the primal and dual updates of step 5 and 6, note that the subgradient of the Lagrangian in (17) with respect to primal variables is given by

$$\partial_{x_n^i} \mathcal{L}_t(x_n^i, \lambda^i) := g_{ni,t} \left[\frac{W(c\mu_n + \nu_n \lambda^i)}{(c\mu_n + \nu_n x_n^i)^2} - \frac{1}{h_{n,t}^i} \right] \cdot \mathbf{1}(x_n^i)$$

and the gradient of the Lagrangian with respect to the dual variable $\boldsymbol{\lambda}^i$ is given by

$$\nabla_{\lambda^i} \mathcal{L}_t(x_n^i, \lambda^i) = \gamma_i - \sum_{n \in \mathcal{N}_i} g_{ni,t} \left(\frac{W}{c\mu_n + \nu_n x_{n,t}^i} - \frac{1}{h_{n,t}^i} \right)_+ - \delta\epsilon\lambda^i.$$

Observe here that the implementation in Algorithm 2 allows the primal updates to be carried out in a decentralized manner. On the other hand, the base station carries out the dual updates. Consequently both, the SCBSs and the BSs may utilize old price iterates $x_{n,[t]}^i$.

Algorithm 2 Online interference management through pricing

Require: initialization $\mathbf{x}_0$ and $\boldsymbol{\lambda}_0 = \mathbf{0}$, step-size ϵ, regularizer δ
1: **for** $t = 1, 2, \ldots, T$ **do**
2: **loop in parallel** for all MU and SCBS user
3: Send dual vars. λ_t^m to nbhd.
4: Observe the delayed primal and dual (sub)-gradients
5: Update the price $x_{n,t+1}^i$ at SCBS n as

$$x_{n,t+1}^i = \mathcal{P}_{\mathcal{X}_n}\left[x_{n,t}^i + \epsilon \left(g_{ni,[t]} \left[\frac{W(c\mu_n + \nu_n \lambda_t^i)}{(c\mu_n + \nu_n x_{n,[t]}^i)^2} - \frac{1}{h_{n,[t]}^i} \right] \cdot \mathbf{1}(x_{n,[t]}^i) \right) \right]$$

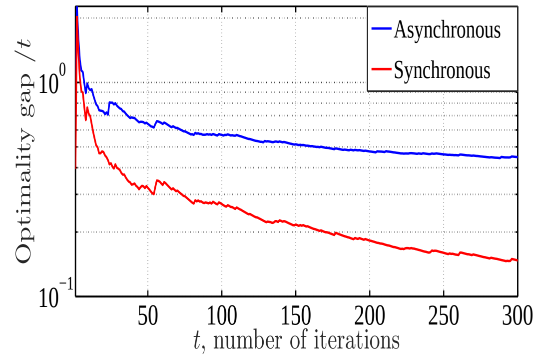
6: Update dual variables at each MU m [cf. (9)]

$$\lambda_{t+1}^i = (1 + \delta\epsilon^2)\lambda_t^i - \epsilon \left(\gamma_i - \sum_{n \in \mathcal{N}_i} g_{ni,[t]} \left(\frac{W}{c\mu_n + \nu_n x_{n,[t]}^i} - \frac{1}{h_{n,[t]}^i} \right)_+ \right).$$

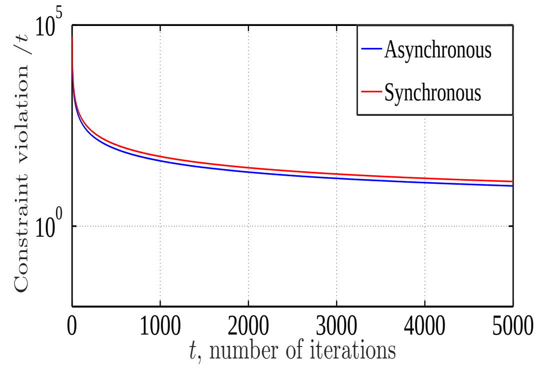
7: **end loop**
8: **end for**

For the simulation purposes, we considered a cellular network with $m = 2$ MBSs and $n = 3$ SCBSs with index $\{m1, m2\}$ and $\{s1, s2, s3\}$. The scenario considered is similar to as shown in Fig. 2, means that {s1, s2} are in the neighborhood of MU $m1$ and $\{s2, s3\}$ constitutes the neighborhood of MU $m2$. The random channel gain g_{ni} and h_n^i are assumed to be exponentially distributed with mean $\mu = 3$. The minimum and maximum values $C_{\min} = 2$ and $C_{\max} = 20$. The other parameter values are $W = 1MHz$, $\nu = 0.6, \gamma_i = 0.5, \delta = 10^{-5}$, $c = 0.1$, $\mu_n = \nu_n = 1$, and $\epsilon = 0.01$. The maximum delay parameter is $\tau = 10$.

Fig. 3a shows the difference of running average of primal objective from its optimal value. It is important to note that the difference goes to zero as $t \to \infty$. The result for both synchronous and asynchronous algorithm algorithms are plotted. The optimal value to plot Fig. 3a is obtained by running the synchronous algorithm for long duration of time and utilizing the converged value as the optimal one. We observe that running the saddle point method without synchrony breaks the bottleneck associated with heterogeneous computing capabilities of different nodes, although it attains slightly slower learning than its synchronized counterpart.



(a) Average objective sub-optimality vs. iteration t



(b) Average constraint violation vs. iteration t

Fig. 3: Algorithm 2 applied to a 5G cellular network with two MBS user and three SCBSs. The y axis of first figure is $\frac{1}{t}\sum_{u=1}^{t}\mathbb{E}[F(\mathbf{x}_u) - F(\mathbf{x}^*)]$ and of second figure is $(1/t)\mathbb{E}\Big[\Big[\sum_{u=1}^{t}\Big(h^i\left(\{\mathbf{x}^j, \boldsymbol{\theta}_t^j\}_{j\in n_i'}\right)\Big)\Big]_+\Big]$. Observe that both the asynchronous and synchronous implementations attain convergence but the asynchronous method settles to a higher level of sub-optimality. Thus, we may solve decentralized online learning problems without a synchronized clock.

Fig. 3b shows the behavior of constraint violation term derived in (12) for a randomly chosen MBS. In the sample path of the empirical average constraint violation, the trend of sublinear growth of objective sub-optimality from Fig. 3a is further substantiated by convergence in expectation of the constraint violation as the iteration index t increases. We observe that the performance reduction of asynchronous operations relative to synchronous ones is smaller with respect to constraint violation as compared with primal-suboptimality, corroborating the rate analysis of Theorem 1.

To further demonstrate the advantage of asynchronous implementation over the synchronous implementation, we have compared the total revenue of the scheme in Fig. 4. It is emphasized that the synchronous implementation is akin

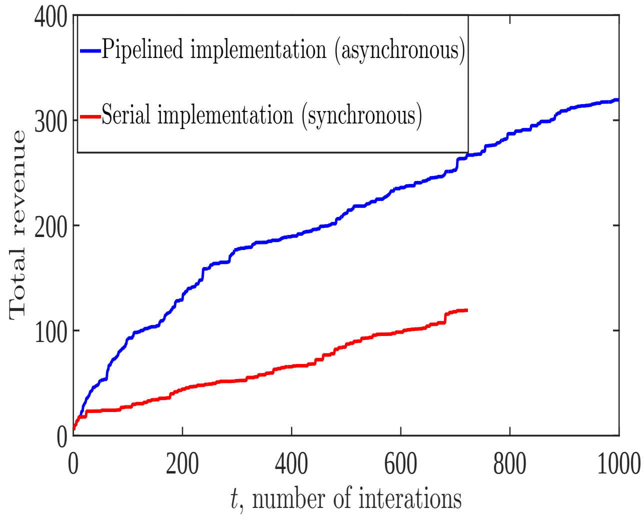


Fig. 4: Average constraint violation vs. iteration t

to serial implementation while asynchronous implementation allows pipelined execution of the updates. We consider a scenario in which each node introduces a processing delay of 7 clock cycles (equal to iteration in this case) in the network. For the synchronous algorithm, each node must wait for 7 clock cycles to get the required information to perform the primal and dual updates. For instance, over 5000 time slots, synchronous updates may only be carried out 724 times as the nodes are waiting for updates at other times. On the other hand, a pipelined implementation is possible with the asynchronous algorithm since at each time t, the node may simply use information from $t-7$. Consequently, the asychronous updates occur at every time slot, resulting in a higher revenue even when using delayed information, as apparent from Fig. 4.

References

[1] A. S. Bedi, A. Koppel, and K. Rajawat, "Asynchronous decentralized stochastic optimization in heterogeneous networks," *arXiv preprint arXiv:1707.05816*, 2017.

[2] 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.

[3] A. Koppel, J. Fink, G. Warnell, E. Stump, and A. Ribeiro, "Online learning for characterizing unknown environments in ground robotic vehicle models," in *Proc. of IEEE IROS*, 2016, pp. 626–633.

[4] 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.

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

[6] 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.

[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

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

[9] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

[10] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, Ed. Cambridge, UK: Cambridge University Press, 1998.

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

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

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

[14] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

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

[16] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4l: Decentralized dynamic discriminative dictionary learning," *IEEE Trans. Signal Inf. Process. over Netw.*, 2017.

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

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, 2011.

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

[20] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

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

[22] 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.

[23] A. S. Bedi, A. Koppel, and K. Rajawat, "Technical report for asynchronous saddle point method: Interference management through pricing," 2018. [Online]. Available: https://tinyurl.com/yb6t5var

[24] D. Jakovetic, J. M. F. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *CoRR*, vol. abs/1112.2972, Apr. 2011.

[25] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *6th IEEE SPAWC*, Jun. 5-8 2005, pp. 1088–1092.

[26] F. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 450–466, Feb. 2013.

[27] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J Optimiz. Theory App.*, vol. 147, no. 3, pp. 516–545, Sep. 2010.

[28] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[29] 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.

[30] A. Nedić, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," *Studies in Computational Mathematics*, vol. 8, no. C, pp. 381–407, 2001.

[31] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optimiz.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[32] F. R. Bach, "Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression." *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 595–627, 2014.

[33] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. of the ICML*, vol. 20, no. 2, Washington DC, USA, Aug. 21-24 2003, pp. 928–936.

[34] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, 2005.

[35] S. Bu, F. R. Yu, and H. Yanikomeroglu, "Interference-aware energy-efficient resource allocation for ofdma-based heterogeneous networks with incomplete channel state information," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1036–1050, 2015.