# Distributed Online Optimization in Dynamic Environments Using Mirror Descent

Shahin Shahrampour and Ali Jadbabaie, *Fellow, IEEE*

***Abstract*—This work addresses decentralized online optimization in nonstationary environments. A network of agents aim to track the minimizer of a global, time-varying, and convex function. The minimizer follows a known linear dynamics corrupted by unknown unstructured noise. At each time, the global function (which could be a tracking error) can be cast as a sum of a finite number of local functions, each of which is assigned to one agent in the network. Moreover, the local functions become available to agents sequentially, and agents do not have prior knowledge of the future cost functions. Therefore, agents must communicate with each other to build an online approximation of the global function. We propose a decentralized variation of the celebrated mirror descent algorithm, according to which agents perform a consensus step to follow the global function and take into account the dynamics of the global minimizer. In order to measure the performance of the proposed online algorithm, we compare it to its offline counterpart, where the global functions are available *a priori*. The gap between the two losses is defined as dynamic regret. We establish a regret bound that scales inversely in the spectral gap of the network and represents the deviation of minimizer sequence with respect to the given dynamics. We show that our framework subsumes a number of results in distributed optimization.**

***Index Terms*—Autonomous systems, distributed optimization, distributed tracking, online learning, time-varying optimization.**

## I. Introduction

DISTRIBUTED convex optimization has received a great deal of attention from many researchers across diverse areas in science and engineering. Classical problems such as decentralized tracking, estimation, and detection are in essence optimization problems [1]–[5]. Early studies on parallel and distributed computation date back to three decades ago (see, e.g., seminal works of [6] and [7]). In any decentralized scheme, the objective is to perform a *global* task assigned to a number of agents in a network. Each individual agent has limited resources or partial information about the task. As a result, agents engage in *local* interactions to complement their insufficient knowledge and accomplish the global task. The use of decentralized techniques has increased rapidly, since they impose a low computational burden on agents and are robust to node failures as opposed to centralized algorithms, which heavily rely on a single information processing unit.

In distributed optimization, the main task is often minimization of a global convex function, written as the sum of local convex functions, where each agent holds a private copy of one specific local function. Then, based on a communication protocol, agents exchange local gradients to minimize the global cost function.

Decentralized optimization is a mature discipline in addressing problems dealing with *time-invariant* cost functions [8]–[11]. However, in many real-world applications, cost functions vary over time. Consider, for instance, the problem of tracking a moving target, where the goal is to follow the position, velocity, and acceleration of the target. One should tackle the problem by minimizing a loss function defined with respect to these parameters; however, since they are *time variant*, the cost function becomes *dynamic*.

When the problem framework is dynamic in nature, there are two key challenges one needs to consider:

1) Agents often observe the local cost functions in an *online* or a *sequential* fashion, i.e., the local functions are revealed only after agents make their instantaneous decisions at each round, and agents are unaware of future cost functions. In the last ten years, this problem (in the centralized domain) has been the main focus of the online optimization field in the machine learning community [12].
2) Any online algorithm should mimic the performance of its offline counterpart, and the gap between the two is called *regret*. The most stringent benchmark is an offline problem that aims to specify the minimizer of the global cost function at all times, which brings forward the notion of *dynamic* regret [13]. It is well known that this benchmark makes the problem intractable in the worst case. However, as studied in the centralized online optimization, the hardness of the problem can be characterized via a complexity measure that captures the variation in the minimizer sequence [13]–[15].

Though the first challenge has been recently addressed in the distributed setting (see, e.g., [16]–[18]), in this paper, we

Manuscript received January 19, 2017; revised May 1, 2017; accepted June 21, 2017. Date of publication August 23, 2017; date of current version February 26, 2018. This work was supported by Office of Naval Research Basic Research Challenge Program under Grant N000141210997 and DARPA's FunLoL Program under Grant W911NF1610578. Recommended by Associate Editor P. Bolzern. *(Corresponding author: Shahin Shahrampour.)*

S. Shahrampour is with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: shahin@seas.harvard.edu).

A. Jadbabaie is with the Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. (e-mail: jadbabai@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2017.2743462

aim to consider both directions simultaneously, focusing on the notion of dynamic regret for distributed online optimization. We consider an online optimization problem where the global cost is realized sequentially and the objective is to track the minimizer of the function with purely local information. The minimizer follows a linear dynamics whose state matrix is common knowledge, but the time-varying minimizer sequence can deviate from this path due to an *unstructured* noise.

For this setup, we propose a decentralized version of the well-known mirror descent algorithm[1] developed by Nemirovksi and Yudin [19]. Using the notion of Bregman divergence in lieu of Euclidean distance for projection, mirror descent has been shown to be a powerful tool for large-scale optimization. Our algorithm consists of three interleaved updates.

1) Each agent follows the local gradient while staying close to the previous estimates in the local neighborhood.
2) Agents take into account the dynamics of the minimizer sequence.
3) Agents average their estimates in their local neighborhood in a consensus step.

Motivated by centralized online optimization, we use the notion of *dynamic* regret to characterize the difference between our online decentralized algorithm and its offline centralized version. We establish a regret bound that scales inversely in the spectral gap of the network, and more notably, it represents the deviation of the minimizer sequence with respect to the given dynamics. We further consider stochastic optimization, where agents observe only noisy versions of their local gradients, and we prove that, in this case, our regret bound holds true in the expectation sense.

Our results subsume two important classes of decentralized optimization in the literature: 1) decentralized optimization of *time-invariant* objectives; and 2) decentralized optimization of a finite sum of objectives over *a common* variable (e.g., risk minimization). This generalization is due to allowing dynamics in both objective function and optimization variable.

We finally show that our algorithm is applicable to decentralized tracking of dynamic parameters. In particular, we show that the problem can be posed as the minimization of the quadratic loss using Euclidean distance as the Bregman divergence. We then empirically verify that the tracking quality depends on how well the parameter follows its given dynamics.

### A. Related Literature

This work is related to two distinct bodies of literature: 1) decentralized optimization and 2) online optimization in dynamic environments. This work lies at the interface of the two and provides a framework for decentralized *online* optimization in *nonstationary* environments. Below, we provide an overview of the related works to both scenarios.

***1) Decentralized Optimization:*** There is a host of results in the literature of decentralized optimization for time-invariant functions. The seminal work of [8] studies distributed subgradient methods over time-varying networks and provides convergence analysis. The effect of stochastic gradients is then considered in [9]. Of particular relevance to this work is [22], where decentralized mirror descent has been developed for when agents receive the gradients with a delay. Moreover, Rabbat [23] proposes a decentralized mirror descent for stochastic composite optimization problems and provides guarantees for strongly convex regularizers. In [24], Raginsky and Bouvrie investigate a distributed stochastic mirror descent algorithm in the continuous-time domain. On the other hand, Duchi *et al.* [10] study dual averaging for distributed optimization and provide a comprehensive analysis on the impact of network parameters on the problem. The extension of dual averaging to online distributed optimization is considered in [16]. Mateos-Núnez and Cortés [17] consider online optimization using subgradient descent of local functions, where the graph structure is time-varying. In [25], a decentralized variant of Nesterov's primal-dual algorithm is proposed for online optimization. Finally, in [18], distributed online optimization is studied for strongly convex objective functions over time-varying networks.

***2) Online Optimization in Dynamic Environments:*** In online optimization, the benchmark can be defined abstractly in terms of a time-varying sequence, a particular case of which is the minimizer sequence of a time-varying cost function. Several versions of the problem have been studied in the literature of machine learning in the centralized case. In [13], Zinkevich develops the online gradient descent algorithm and considers its extension to time-varying sequences. The authors of [14] generalize this idea to study time-varying sequences following a given dynamics. Besbes *et al.* [26] restrict their attention to the minimizer sequence and introduce a complexity measure for the problem in terms of the variations in the cost function. For the same problem, the authors of [15] develop an adaptive algorithm whose regret bound is expressed in terms of the variations of both cost function and minimizer sequence, while in [27], an improved rate is derived for strongly convex objectives. Moreover, online dynamic optimization with linear objectives is discussed in [28]. Fazlyab *et al.* [29] consider interior point methods and provide continuous-time analysis for the problem, whereas Simonetto *et al.* [30] develop a second-order method working based on prediction and correction. Moreover, Yang *et al.* [31] provide optimal bounds for when the minimizer belongs to the feasible set. Finally, *high-probability* regret bounds in dynamic and decentralized setting are studied in [32].

### B. Organization

This paper is organized as follows. The notation, problem formulation, assumptions, and algorithm are described in Section II. In Section III, we provide our theoretical results characterizing the behavior of the dynamic regret. Section IV is dedicated to an application of our method in decentralized tracking of dynamic parameters. Section V concludes, and the proofs are given in the Appendix.

## II. Problem Formulation and Algorithm

*Notation:* We use the following notation in the exposition of our results:

[1]Algorithms relying on gradient descent minimize Euclidean distance in the projection step. Mirror descent generalizes the projection step using the concept of Bregman divergence [19], [20]. Euclidean distance is a special Bregman divergence that reduces mirror descent to gradient descent. Kullback–Leibler divergence is another well-known type of Bregman divergence (see, e.g., [21] for more details on Bregman divergence).

| | |
|---|---|
| $[n]$ | The set $\{1, 2, \ldots, n\}$ for any integer $n$ |
| $x^\top$ | Transpose of the vector $x$ |
| $x(k)$ | The $k$th element of vector $x$ |
| $I_n$ | Identity matrix of size $n$ |
| $\Delta_d$ | The $d$-dimensional probability simplex |
| $\langle \cdot, \cdot \rangle$ | Standard inner product operator |
| $\mathbb{E}[\cdot]$ | Expectation operator |
| $\\|\cdot\\|_p$ | $p$-norm operator |
| $\\|\cdot\\|_*$ | The dual norm of $\\|\cdot\\|$ |
| $\sigma_i(W)$ | The $i$th largest eigenvalue of $W$ in magnitude |

Throughout this paper, all the vectors are in column format.

### A. Decentralized Optimization in Dynamic Environments

In this work, we consider an optimization problem involving a *global* convex function. Let $\mathcal{X}$ be a convex compact set, and represent the global function by $f_t : \mathcal{X} \to \mathbb{R}$ at time $t$. The global function is time-varying, and the goal is to track the minimizer of $f_t(\cdot)$, denoted by $x_t^\star$, for a *finite* time $T$. We address a problem whose *offline* and *centralized* version can be posed as follows:

$$\underset{x_1,\ldots,x_T}{\text{minimize}} \quad \sum_{t=1}^{T} f_t(x_t)$$
$$\text{subject to} \quad x_t \in \mathcal{X},\ t \in [T]. \tag{1}$$

If all the functions $f_t(\cdot)$ for $t \in [T]$ are known in advance, since the summands in (1) are minimized over independent variables, the offline and centralized solution results in the value of $x_t^\star$ for all $t \in [T]$. However, the goal is to solve the problem above in an *online* and *decentralized* fashion. That is, the global function at each time $t$ is decomposed as the sum of $n$ *local* functions as

$$f_t(x) := \frac{1}{n}\sum_{i=1}^{n} f_{i,t}(x) \tag{2}$$

where $f_{i,t} : \mathcal{X} \to \mathbb{R}$ is a local convex function on $\mathcal{X}$ for all $i \in [n]$, and we have a network of $n$ agents facing the following two challenges in solving problem (1).

1) *Decentralized challenge:* Agent $j \in [n]$ receives private information only about $f_{j,t}(\cdot)$ and does not have access to the global function $f_t(\cdot)$. Therefore, agents must communicate with each other to approximate the global function, which is common to decentralized schemes.
2) *Online challenge:* The functions are revealed to agents sequentially along the time horizon, i.e., at any time instance $s$, agent $j$ has observed $f_{j,t}(\cdot)$ for $t < s$, whereas the agent does not know $f_{j,t}(\cdot)$ for $s \le t \le T$, which is common to online settings.

In this work, we address the online and decentralized form of problem (1) subject to an additional constraint. In particular, we assume that a stable nonexpansive mapping $A$ (as noted in Assumption 5) is *common* knowledge in the network, and

$$x_{t+1}^\star = Ax_t^\star + v_t \tag{3}$$

where $v_t$ is an unknown, unstructured noise unstructured noise. Assuming no structure on the noise can prove beneficial when $A$ is unknown, but an oracle can provide the network with a proper estimate of it. In such a case, if the network uses the estimate $\hat{A}$ in lieu of $A$, we have

$$x_{t+1}^\star = Ax_t^\star + v_t = \hat{A}x_t^\star + (A - \hat{A})x_t^\star + v_t = \hat{A}x_t^\star + v_t'$$

where $v_t' := (A - \hat{A})x_t^\star + v_t$ is small when the estimate $\hat{A}$ is close enough to $A$.

Note that our framework subsumes two important classes of decentralized optimization in the literature.

1) The existing methods often consider *time-invariant* objectives (see, e.g., [8], [10], and [22]). This is simply the special case where $f_t(x) = f(x)$ and $x_t = x$ in (1).
2) Online algorithms deal with *time-varying* functions, but often the network's objective is to minimize the temporal average of $\{f_t(x)\}_{t=1}^T$ over a fixed variable $x$ (see, e.g., [16] and [17]). This can be captured by our setup when $x_t = x$ in (1).

In both cases above, $A = I_d$ and $v_t = \mathbb{0}$ in (3). To exhibit the online nature of the problem, the latter class in above is usually reformulated in terms of a popular performance metric called *regret*. Since in that setup, $x_t = x$ for $t \in [T]$, denoting by $x^\star := \text{argmin}_{x\in\mathcal{X}} \sum_{t=1}^T f_t(x)$, the solution to problem (1) becomes $\sum_{t=1}^T f_t(x^\star)$. Then, the goal of online algorithm is to mimic its offline version by minimizing the regret defined as

$$\boldsymbol{Reg}_T^s := \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} f_t(x_{i,t}) - \sum_{t=1}^{T} f_t(x^\star) \tag{4}$$

where $x_{i,t}$ is the estimate of agent $i$ for $x^\star$ at time $t$. The superscript "s" reiterates the fact that the benchmark is minimum of the sum $\sum_{t=1}^T f_t(x)$ over a *static* or *fixed* comparator variable $x$ that resides in the set $\mathcal{X}$. In this setup, a successful algorithm incurs a sublinear regret, which asymptotically closes the gap between the online algorithm and the offline algorithm (when normalized by $T$).

The main focus of this paper, however, is to study the scenario where *functions* and *comparator variables* evolve simultaneously, i.e., the variables $\{x_t\}_{t=1}^T$ are not constrained to be fixed in (1). Let $x_t^\star := \text{argmin}_{x\in\mathcal{X}} f_t(x)$ be the minimizer of the global function at time $t$. Then, the solution to problem (1) is simply $\sum_{t=1}^T f_t(x_t^\star)$. Therefore, to capture the online nature of problem (1), we reformulate it using the notion of *dynamic* regret as

$$\boldsymbol{Reg}_T^d := \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} f_t(x_{i,t}) - \sum_{t=1}^{T} f_t(x_t^\star) \tag{5}$$

where $x_{i,t}$ is the estimate of agent $i$ for $x_t^\star$ at time $t$. Recalling that the objective is to track $x_t^\star$ in a distributed fashion, the dynamic regret serves as a proxy to measure the quality of performance through entire $T$ rounds. In other words, the dynamic regret formulates a finite-time version of the infinite-horizon problem. Intuitively, when agents' estimates are close to $x_t^\star$, the dynamic regret becomes small. Therefore, our objective can be translated to minimizing the dynamic regret measuring the gap between the online algorithm and its offline version. The superscript "d" indicates that the benchmark is the sum of minima $\sum_{t=1}^{T} f_t(x_t^\star)$ characterized by dynamic variables $\{x_t^\star\}_{t=1}^T$ that lie in the set $\mathcal{X}$.

It is well known that the more stringent benchmark in the dynamic setup makes the problem intractable in the worst case, i.e., achieving a sublinear regret could be impossible. In parallel to studies of centralized online optimization [14], [15], [26], the hardness of the problem should be characterized via a complexity measure that captures the variation in the minimizer sequence $\{x_t^\star\}_{t=1}^T$. More specifically, we want to prove a regret bound in terms of

$$C_T := \sum_{t=1}^{T} \left\|x_{t+1}^\star - Ax_t^\star\right\| = \sum_{t=1}^{T} \|v_t\| \tag{6}$$

which represents the deviation of the minimizer sequence with respect to the dynamics matrix $A$. We shall see that sublinearity of $C_T$ warrants sublinearity of the regret bound, posing a constraint on $\{v_t\}_{t=1}^T$ to achieve sublinear dynamic regret.

The problem setup (1) coupled with the dynamics given in (3) is reminiscent of distributed Kalman filtering [33]. However, there are fundamental differences: 1) The mismatch noise $v_t$ is neither Gaussian nor drawn from a known statistical distribution. It can be thought as an adversarial noise with *unknown* structure, which represents the deviation of the minimizer from the dynamics.[2] 2) Agents' observations are not necessarily linear; in fact, the observations are local gradients of $\{f_{i,t}(\cdot)\}_{t=1}^T$ and are nonlinear when the objective is nonquadratic. Furthermore, the focus here is on the *finite-time* analysis rather than asymptotic results.

We note that our framework also differs from distributed particle filtering [34], since agents receive only one observation per iteration, and again, the mismatch noise $v_t$ has no structure or distribution.

In order to minimize the dynamic regret in (5), we decentralize the mirror descent algorithm [19] and analyze it in a dynamic framework. The appealing feature of mirror descent is the generalization of the projection step by using Bregman divergence in lieu of Euclidean distance, which makes the algorithm applicable to a wide range of problems. Before defining Bregman divergence and elaborating on the algorithm, we start by stating a couple of standard assumptions in the context of decentralized optimization.

*Assumption 1:* For any $i \in [n]$, the function $f_{i,t}(\cdot)$ is Lipschitz continuous on $\mathcal{X}$ with a uniform constant $L$. That is,

$$|f_{i,t}(x) - f_{i,t}(y)| \leq L\,\|x-y\|$$

for any $x, y \in \mathcal{X}$. This implies that $f_t(\cdot)$ is also Lipschitz continuous on $\mathcal{X}$ with the constant $L$. It further implies that the gradient of $f_{i,t}(\cdot)$ denoted by $\nabla f_{i,t}(\cdot)$ is uniformly bounded on $\mathcal{X}$ by the constant $L$, i.e., we have $\|\nabla f_{i,t}(\cdot)\|_* \leq L$,[3] where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

[2]In online learning, the focus is not on the distribution of data. Instead, data are assumed to be generated arbitrarily, and their effect is observed through the loss functions [12].

[3]This relationship is standard; see, e.g., [12, Lemma 2.6] for more details.

Agents exchange information according to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [n]$ denotes the set of nodes (agents), and $\mathcal{E}$ is the set of edges (links between agents). Each agent $i$ assigns a positive weight $[W]_{ij}$ for the information received from agent $j \neq i$. Hence, the set of neighbors of agent $i$ is defined as $\mathcal{N}_i := \{j : [W]_{ij} > 0\}$.

*Assumption 2:* The network is connected, i.e., there exists a path from any agent $i \in [n]$ to any agent $j \in [n]$. Furthermore, the matrix $W$ is symmetric with positive diagonal, and it is doubly stochastic , i.e.,

$$\sum_{i=1}^{n}[W]_{ij} = \sum_{j=1}^{n}[W]_{ij} = 1.$$

### *B. Decentralized Online Mirror Descent*

The development of mirror descent relies on the Bregman divergence outlined in this section. Consider a convex set $\mathcal{X}$ in a Banach space $\mathcal{B}$, and let $\mathcal{R} : \mathcal{B} \to \mathbb{R}$ denote a 1-strongly convex function on $\mathcal{X}$ with respect to a norm $\|\cdot\|$. That is,

$$\mathcal{R}(x) \geq \mathcal{R}(y) + \langle \nabla\mathcal{R}(y), x-y\rangle + \frac{1}{2}\|x-y\|^2.$$

for any $x, y \in \mathcal{X}$. Then, the Bregman divergence $\mathcal{D}_\mathcal{R}(\cdot,\cdot)$ with respect to the function $\mathcal{R}(\cdot)$ is defined as follows:

$$\mathcal{D}_\mathcal{R}(x,y) := \mathcal{R}(x) - \mathcal{R}(y) - \langle x-y, \nabla\mathcal{R}(y)\rangle.$$

Combining the previous two formulas yields an important property of the Bregman divergence, and for any $x, y \in \mathcal{X}$, we get

$$\mathcal{D}_\mathcal{R}(x,y) \geq \frac{1}{2}\|x-y\|^2 \tag{7}$$

due to the strong convexity of $\mathcal{R}(\cdot)$. Two well-known examples are the Euclidean distance $\mathcal{D}_\mathcal{R}(x,y) = \|x-y\|^2$ generated from $\mathcal{R}(x) = \|x\|_2^2$, and the Kullback–Leibler (KL) divergence $\mathcal{D}_\mathcal{R}(x,y) = \sum_{i=1}^{d} x(i)(\log x(i) - \log y(i))$ between two points $x, y \in \Delta_d$ generated from $\mathcal{R}(x) = \sum_{i=1}^{d} x(i)\log x(i) - x(i)$. We now state some mild assumptions on the Bregman divergence as follows.

*Assumption 3:* Let $x$ and $\{y_i\}_{i=1}^n$ be vectors in $\mathbb{R}^d$. The Bregman divergence satisfies the separate convexity in the following sense:

$$\mathcal{D}_\mathcal{R}\left(x, \sum_{i=1}^{n}\alpha(i)y_i\right) \leq \sum_{i=1}^{n}\alpha(i)\mathcal{D}_\mathcal{R}(x, y_i)$$

where $\alpha \in \Delta_n$ is on the $n$-dimensional simplex.

The assumption is satisfied for commonly used Bregman divergences. As an example, the Euclidean distance satisfies the condition. The KL divergence also satisfies the constraint, and we refer the reader to [21, Th. 6.4] for the proof.

*Assumption 4:* The Bregman divergence satisfies a Lipschitz condition of the form

$$|\mathcal{D}_\mathcal{R}(x,z) - \mathcal{D}_\mathcal{R}(y,z)| \leq K\|x-y\|$$

for all $x, y, z \in \mathcal{X}$.

When the function $\mathcal{R}$ is Lipschitz on $\mathcal{X}$, the Lipschitz condition on the Bregman divergence is automatically satisfied. Again, for the Euclidean distance, the assumption evidently holds. In the particular case of the KL divergence, the condition can be achieved via mixing a uniform distribution to avoid the boundary. More specifically, consider $\mathcal{R}(x) = \sum_{i=1}^{d} x(i)\log x(i) - x(i)$, for which $|\nabla\mathcal{R}(x)| = |\sum_{i=1}^{d}\log x(i)| \leq d\log T$ as long as $x \in \{\mu : \sum_{i=1}^{d}\mu(i) = 1; \mu(i) \geq \frac{1}{T},\ \forall i \in [d]\}$. Therefore, in this case, the constant $K$ is of $\mathcal{O}(\log T)$ (see, e.g., [15] for more comments on the assumption).

We are now ready to propose a three-step algorithm to solve the optimization problem formulated in terms of dynamic regret in (5). We define the shorthand $\nabla_{i,t} := \nabla f_{i,t}(x_{i,t})$ to denote the local gradient evaluated at the corresponding estimate. Noticing the dynamic framework, we develop the decentralized online mirror descent via the following updates[4]:

$$\hat{x}_{i,t+1} = \text{argmin}_{x\in\mathcal{X}}\left\{\eta_t\langle x, \nabla_{i,t}\rangle + \mathcal{D}_\mathcal{R}(x, y_{i,t})\right\} \tag{8a}$$

$$x_{i,t+1} = A\hat{x}_{i,t+1}, \quad \text{and} \quad y_{i,t+1} = \sum_{j=1}^{n}[W]_{ij}x_{j,t+1} \tag{8b}$$

where $\{\eta_t\}_{t=0}^{T}$ is the global step-size sequence, and $A \in \mathbb{R}^{d\times d}$ is the given dynamics in (3), which is common knowledge. Recall that $x_{i,t} \in \mathbb{R}^d$ represents the estimate of agent $i$ for the global minimizer $x_t^\star$ at time $t$. The step-size sequence is nonincreasing and positive. Our proposed methodology can also be recognized as the decentralized variant of the dynamic mirror descent algorithm in [14] for the case of linear dynamics.

We remark that the set of updates (8a), (8b) can recover a few interesting frameworks in the existing literature. Optimizing a quadratic local objective using Euclidean distance as the Bregman divergence, one can recover the updates in [35] for distributed state estimation. On the other hand, when optimizing a linear objective over the probability simplex, one can use the KL divergence as the Bregman to recover the updates in [36] for distributed detection and social learning (with $A = I_d$). In the same context, replacing the arithmetic averaging by geometric averaging in (8b), one can recover faster learning updates as in [4], [5], and [36].

For a general optimization problem, the update (8a) allows the algorithm to follow the private gradient while staying close to the previous estimates in the local neighborhood. This proximity is in the sense of minimizing the Bregman divergence. The first update in (8b) takes into account the dynamics of the minimizer sequence, and the second update in (8b) is the *consensus* term averaging the estimates in the local neighborhood.

*Assumption 5:* The mapping $A$ is assumed to be *nonexpansive*. That is, the condition

$$\mathcal{D}_\mathcal{R}\big(Ax, Ay\big) \leq \mathcal{D}_\mathcal{R}\big(x, y\big)$$

holds for all $x, y \in \mathcal{X}$, and $\|A\| \leq 1$.

The assumption postulates a natural constraint on the mapping $A$: it does not allow the effect of a poor prediction (at one step) to be amplified as the algorithm moves forward.

[4]The algorithm is initialized at $x_{i,0} = \mathbb{0}$ to avoid clutter in the analysis. In general, any initialization inside the feasible set could work for the algorithm.

## III. Theoretical Results

In this section, we state our theoretical results and their consequences. The proofs are presented later in the Appendix. Our main result (see Theorem 3) proves a bound on the dynamic regret which captures the deviation of the minimizer trajectory from the matrix $A$ (tracking error) as well as the decentralization cost (network error). After stating the theorem, we show that our result recovers previous rates on decentralized optimization (static regret) once the tracking error is removed. Also, it recovers previous rates on centralized online optimization in the dynamic setting when the network error is factored out. This establishes that our generalization is valid, since our result recovers special cases.

### A. Preliminary Results

We start with a convergence result on the local estimates which presents an upper bound on the deviation of the local estimates at each iteration from their consensual value. A similar result has been proven in [22] for *time-invariant* functions *without* dynamics. The following lemma extends that of [22] to *online* setting and takes into account the *dynamics* matrix $A$ in (8b).

*Lemma 1 (Network error):* Given Assumptions 1, 2, and 5, the local estimates $\{x_{i,t}\}_{t=1}^{T}$ generated by the updates (8a), (8b) satisfy

$$\|x_{i,t+1} - \bar{x}_{t+1}\| \leq L\sqrt{n}\sum_{\tau=0}^{t}\eta_\tau\sigma_2^{t-\tau}(W)$$

for any $i \in [n]$, where $\bar{x}_t := \frac{1}{n}\sum_{i=1}^{n} x_{i,t}$.

It turns out that the error bound depends on $\sigma_2(W)$, the second largest eigenvalue of $W$ in magnitude, as well as the step-size sequence $\{\eta_t\}_{t=1}^{T}$. It is well known that smaller $\sigma_2(W)$ results in the closeness of estimates to their average by speeding up the mixing rate (see, e.g., results of [10]). On the other hand, a usual diminishing step size, which asymptotically goes to zero, can guarantee the asymptotic closeness of the estimates; however, such a step-size sequence is most suitable for *static* rather than *dynamic* environments, as we note that the estimates should be close to the global minimizer as well. We will discuss the choice of step size carefully when we state our main result. Before that, we need to state another lemma as follows.

*Lemma 2 (Tracking error):* Given Assumptions 2–5, it holds that

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\mathcal{D}_\mathcal{R}(x_t^\star, y_{i,t}) - \frac{1}{\eta_t}\mathcal{D}_\mathcal{R}(x_t^\star, \hat{x}_{i,t+1})\right)$$
$$\leq \frac{2R^2}{\eta_{T+1}} + \sum_{t=1}^{T}\frac{K}{\eta_{t+1}}\left\|x_{t+1}^\star - Ax_t^\star\right\|$$

where $R^2 := \sup_{x,y\in\mathcal{X}}\mathcal{D}_\mathcal{R}(x, y)$.

In the update (8a), each agent $i$ follows the local gradient $\nabla_{i,t}$, while staying close to $y_{i,t}$, the previous averaged estimate over the local neighborhood. The "closeness" is achieved by including the Bregman divergence in the minimization. Lemma 2 establishes a bound on the difference of these two quantities, when they are evaluated in the Bregman divergence with respect to $x_t^\star$. The relation of the left-hand side with the dynamic regret is not immediately observable, and it becomes clear in the analysis. However, the term $\left\|x_{t+1}^\star - Ax_t^\star\right\| = \|v_t\|$ in the bound highlights the impact of mismatch noise $v_t$ in the tracking quality. Lemmas 1 and 2 disclose the critical parameters involved in the regret bound. We shall discuss the consequences of these bounds in the subsequent section.

## B. Finite-Time Performance: Regret Bound

In this section, we state our main result on the *nonasymptotic* performance of the decentralized online mirror descent algorithm in dynamic environments. The succeeding theorem provides the regret bound in the general case, and it is followed by a corollary characterizing the regret rate for the optimized, fixed step size. In particular, the theorem uses the results in the previous section to present an upper bound on the dynamic regret, decomposed into tracking and network errors.

*Theorem 3:* Let all the Assumptions 1–5 hold. Then, using the local estimates $\{x_{i,t}\}_{t=1}^T$ generated by the updates (8a), (8b), the regret (5) can be bounded as

$$\boldsymbol{Reg}_T^d \leq \mathtt{E}_{\mathtt{Track}} + \mathtt{E}_{\mathtt{Net}}$$

where

$$\mathtt{E}_{\mathtt{Track}} := \frac{2R^2}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{K}{\eta_{t+1}} \left\|x_{t+1}^\star - Ax_t^\star\right\| + L^2 \sum_{t=1}^{T} \frac{\eta_t}{2}$$

and

$$\mathtt{E}_{\mathtt{Net}} := 4L^2\sqrt{n} \sum_{t=1}^{T}\sum_{\tau=0}^{t-1} \eta_\tau \sigma_2^{t-\tau-1}(W).$$

*Corollary 4:* Under the same conditions stated in Theorem 3, using the fixed step size $\eta = \sqrt{(1-\sigma_2(W))(C_T + 2R^2)/T}$ yields a regret bound of order

$$\boldsymbol{Reg}_T^d \leq \mathcal{O}\left(\sqrt{\frac{(1+C_T)T}{1-\sigma_2(W)}}\right)$$

where $C_T = \sum_{t=1}^{T} \left\|x_{t+1}^\star - Ax_t^\star\right\|$.

*Proof:* The proof of the corollary follows directly from substituting the step size into the bound of Theorem 3. ■

Theorem 3 proves an upper bound on the dynamic regret and decomposes that into two terms for a general step-size sequence $\{\eta_t\}_{t=1}^T$. In Corollary 4, we fix the step size[5] and observe the role of $C_T$ in controlling the regret bound. As we recall from (3), this quantity collects mismatch errors $\{v_t\}_{t=1}^T$ that are not necessarily Gaussian or of some statistical distribution. Also, we observe the impact of $1 - \sigma_2(W)$, defined as the *spectral gap* of the network, in the regret bound.

Before discussing the role of $C_T$ in general, let us consider a few special cases. In Section II, we discussed that our setup generalizes some of the previous works, and it is important to notice that our result recovers the corresponding rates when restricted to those special cases.

1) When the global function $f_t(x) = f(x)$ is time-invariant, the minimizer sequence $\{x_t^\star\}_{t=1}^T$ is fixed, i.e., the mapping $A = I_d$ and $v_t = \mathbb{0}$ in (3). In this case, in Theorem 3, the term involving $\left\|x_{t+1}^\star - Ax_t^\star\right\|$ in $\mathtt{E}_{\mathtt{Track}}$ is equal to zero, and we can use the step-size sequence $\eta = \sqrt{(1-\sigma_2(W))/T}$ to recover the result of comparable algorithms, such as [10], in which distributed dual averaging is proposed.
2) The same argument holds when the global function is time-varying, but the comparator variables are fixed [that is, $x_t = x$ in (1)]. In this case, the problem is reduced to minimizing the static regret (4), and we again have $A = I_d$ and $v_t = \mathbb{0}$ in (3). Since $\left\|x_{t+1}^\star - Ax_t^\star\right\| = 0$ again, our result recovers that of [16] on distributed online dual averaging.
3) When the graph is complete, $\sigma_2(W) = 0$ and the $\mathtt{E}_{\mathtt{Net}}$ term in Theorem 3 vanishes. We then recover the results of [14] on centralized online learning in dynamic environments (in the linear case).

In general, the order of $C_T$ depends completely upon the order of the mismatch errors $\{v_t\}_{t=1}^T$. When the mismatch errors die fast enough, $C_T$ becomes summable and of constant order. On the other hand, when mismatch errors $\{v_t\}_{t=1}^T$ are large and nonvanishing, the minimizer sequence $\{x_t^\star\}_{t=1}^T$ fluctuates drastically, and $C_T$ could become linear in time. The bound in the corollary is then not useful in the sense of keeping the dynamic regret sublinear. Such behavior is natural since even in the centralized online optimization, the algorithm receives only a *single* gradient to predict the next step[6]. As discussed in Section II, in this worst case, the problem is generally intractable. However, our goal was to consider $C_T$ as a complexity measure of the problem environment and express the regret bound with respect to this quantity. It is proved in the centralized case that if the algorithm is allowed to query *multiple* gradients per time, the error is reduced to some extent [37], but pursuing this direction for distributed case is beyond the scope of this paper.

## C. Optimization With Stochastic Gradients

In decentralized tracking, learning, and estimation, agents observations are usually noisy. In this section, we demonstrate that the result of Theorem 3 does not rely on exact gradients, and it holds true in the expectation sense when agents follow stochastic gradients. Mathematically speaking, let $\mathcal{F}_t$ be the $\sigma$-field containing all information prior to the outset of round $t+1$. Let also $\boldsymbol{\nabla}_{i,t}$ represent the stochastic gradient observed by agent $i$ after calculating the estimate $\mathbf{x}_{i,t}$. Then, we define a stochastic oracle that provides noisy gradients respecting the following conditions:

$$\mathbb{E}\left[\boldsymbol{\nabla}_{i,t}\middle|\mathcal{F}_{t-1}\right] = \nabla_{i,t} \qquad \mathbb{E}\left[\left\|\boldsymbol{\nabla}_{i,t}\right\|_*^2\middle|\mathcal{F}_{t-1}\right] \leq G^2 \quad (9)$$

[5]Note that this choice of step size is not adaptive, i.e., the agents need to know the value of $C_T$ to tune the step size. If agents know an upper bound on $C_T$ and use that in lieu of $C_T$ for tuning the step size, the upper bound would replace $C_T$ in the regret bound, giving rise to a looser bound.

[6]Even in a more structured problem setting such as Kalman filtering, when we know the exact value of a state at a time step, we cannot exactly predict the next state, and we incur a minimum mean-squared error of the size of noise variance.

where these conditions guarantee that the stochastic gradient is unbiased and has bounded second moment. The new updates take the following form:

$$\hat{\mathbf{x}}_{i,t+1} = \text{argmin}_{x\in\mathcal{X}}\{\eta_t\langle x, \boldsymbol{\nabla}_{i,t}\rangle + \mathcal{D}_{\mathcal{R}}(x, \mathbf{y}_{i,t})\} \tag{10a}$$

$$\mathbf{x}_{i,t} = A\hat{\mathbf{x}}_{i,t}, \quad \text{and} \quad \mathbf{y}_{i,t} = \sum_{j=1}^{n}[W]_{ij}\mathbf{x}_{j,t} \tag{10b}$$

where the only distinction between (10a) and (8a) is using the stochastic gradient in (10a). A commonly used model to generate stochastic gradients satisfying (9) is an additive zero-mean noise with bounded variance. We now discuss the impact of stochastic gradients in the following theorem.

*Theorem 5:* Let all the Assumptions 1–5 hold as stated in Theorem 3. Furthermore, let the local estimates $\{\mathbf{x}_{i,t}\}_{t=1}^{T}$ be generated by updates (10a), (10b), where the stochastic gradients satisfy the condition (9). Then

$$\mathbb{E}\left[\boldsymbol{Reg}_T^d\right] \leq \frac{2R^2}{\eta_{T+1}} + \sum_{t=1}^{T}\frac{K}{\eta_{t+1}}\left\|x_{t+1}^\star - Ax_t^\star\right\| + G^2\sum_{t=1}^{T}\frac{\eta_t}{2}$$
$$+ 2G(L+G)\sqrt{n}\sum_{t=1}^{T}\sum_{\tau=0}^{t-1}\eta_\tau\sigma_2^{t-\tau-1}(W).$$

The theorem indicates that when using stochastic gradients, the result of Theorem 3 holds true in the expectation sense. Thus, the algorithm can be used in dynamic environments where agents observations are noisy.

## IV. Numerical Experiment: State Estimation and Tracking Dynamic Parameters

In this section, we present an application of our method in distributed tracking in the presence of unstructured noise. Let us consider a slowly maneuvering target in the 2-D plane and assume that each position component of the target evolves independently according to a nearly constant velocity model [38]. The state of the target at each time consists of four components: horizontal position, vertical position, horizontal velocity, and vertical velocity. Therefore, representing the state at time $t$ by $x_t^\star \in \mathbb{R}^4$, the state-space model takes the form

$$x_{t+1}^\star = Ax_t^\star + v_t$$

where $v_t \in \mathbb{R}^4$ is the system noise, and using $\otimes$ for Kronecker product, $A$ is described as

$$A = I_2 \otimes \begin{bmatrix} 1 & \epsilon \\ 0 & 1 \end{bmatrix}$$

with $\epsilon$ being the sampling interval[7]. The goal is to track $x_t^\star$ using a network of agents. For our experiment, we generate this noise "only once" according to a zero-mean Gaussian distribution with covariance matrix $\Sigma$ as follows:

$$\Sigma = \sigma_v^2 I_2 \otimes \begin{bmatrix} \epsilon^3/3 & \epsilon^2/2 \\ \epsilon^2/2 & \epsilon \end{bmatrix}.$$

[7]The sampling interval of $\epsilon$ (seconds) is equivalent to the sampling rate of $1/\epsilon$ (Hz).

We let the sampling interval be $\epsilon = 0.1$ s, which is equivalent to frequency 10 Hz. The constant $\sigma_v^2$ is changed in different scenarios, so we describe the choice of this parameter later. Though generated randomly, the noise is fixed with each run of our experiment later.

We consider a sensor network of $n = 25$ agents located on a $5 \times 5$ grid. Agents aim to track the moving target $x_t^\star$ collaboratively. At time $t$, agent $i$ observes $\mathbf{z}_{i,t}$, a noisy version of one coordinate of $x_t^\star$, as follows:

$$\mathbf{z}_{i,t} = \mathbb{e}_{k_i}^\top x_t^\star + \mathbf{w}_{i,t} \tag{11}$$

where $\mathbf{w}_{i,t} \in \mathbb{R}$ denotes the observation noise, and $\mathbb{e}_k$ is the $k$th unit vector in the standard basis of $\mathbb{R}^4$ for $k \in \{1,2,3,4\}$. We divide agents into four groups, and for each group, we choose one specific $k_i$ from the set $\{1,2,3,4\}$. We generate $\mathbf{w}_{i,t}$ independently from a uniform distribution on $[-1,1]$. Though not locally observable to each agent, it is straightforward to see that the target $x_t^\star$ is *globally* identifiable from the standpoint of the whole network (see, e.g., [35]).

We use the *local* square loss

$$f_{i,t}(x) = \mathbb{E}\left[\left(\mathbf{z}_{i,t} - \mathbb{e}_{k_i}^\top x\right)^2\right]$$

for each agent $i$, resulting in the *network* loss

$$f_t(x) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(\mathbf{z}_{i,t} - \mathbb{e}_{k_i}^\top x\right)^2\right].$$

Now, using Euclidean distance as the Bregman divergence in updates (10a), (10b), we can derive the following update:

$$\mathbf{x}_{i,t} = \sum_{j=1}^{n}[W]_{ij}A\mathbf{x}_{j,t-1} + \eta_t A\mathbb{e}_{k_i}\left(\mathbf{z}_{i,t-1} - \mathbb{e}_{k_i}^\top\mathbf{x}_{i,t-1}\right).$$

We fix the step size to $\eta_t = \eta = 0.5$, since using diminishing step size is not useful in tracking unless we have diminishing system noise [39], [40].

It is proved that in decentralized tracking, the dynamic regret can be presented in terms of the tracking error $\mathbf{x}_{i,t} - x_t^\star$ of all agents (see [41, Lemma 2]). More specifically, the dynamic regret roughly averages the tracking error over network and time (when normalized by $T$). Exploiting this connection and combining that with the result of Theorem 5, we observe that once the parameter does not deviate too much from the linear dynamics, i.e., when $\sum_{t=1}^{T}\|v_t\|$ is small, the bound on the dynamic regret (or equivalently the collective tracking error) becomes small and vice versa.

We demonstrate this intuitive idea by tuning $\sigma_v^2$. Larger values for $\sigma_v^2$ are more likely to cause deviations from the linear dynamics; therefore, we expect large dynamic regret (worse performance) when $\sigma_v^2$ is large. In Fig. 1, we plot the dynamic regret for $\sigma_v^2 \in \{0.25, 0.5, 0.75, 1\}$. For each specific value of $\sigma_v^2$, we run the experiment 50 times and average out the dynamic regret over all runs. As we expected, the performance improves once $\sigma_v^2$ tends to smaller values.

Let us now focus on the case that $\sigma_v^2 = 0.5$. For one run of this case, we provide a snapshot of the target trajectory (in red) in Fig. 2 and plot the estimator trajectory (in blue) for agents

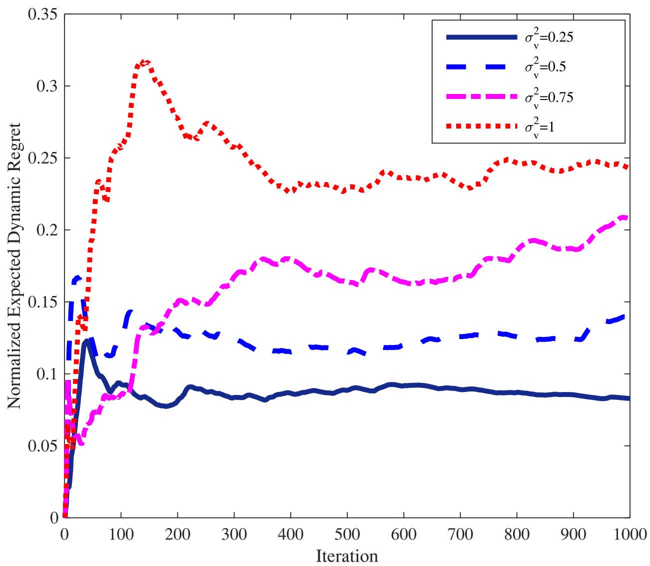


Fig. 1. Plot of dynamic regret versus iterations. Naturally, when $\sigma_v^2$ is smaller, the innovation noise added to the dynamics is smaller with high probability, and the network incurs a lower dynamic regret. In this plot, the dynamic regret is normalized by iterations, so the $y$-axis is $\mathbb{E}\left[\boldsymbol{Reg}_T^d\right]/T$.

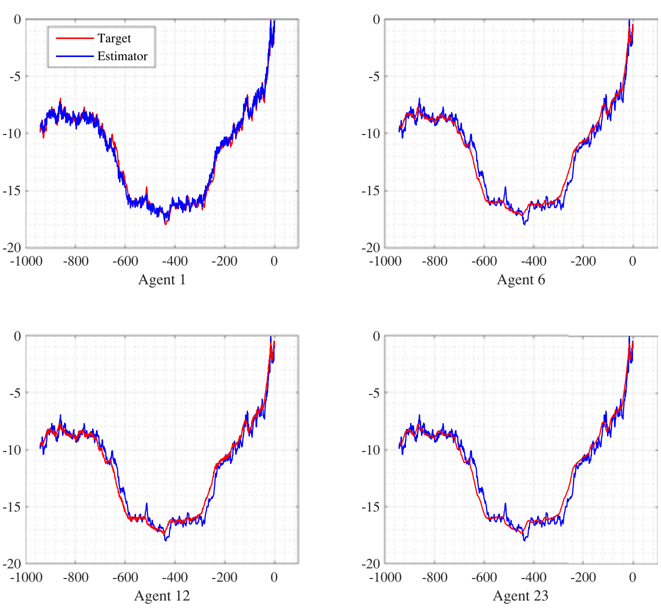


Fig. 2. Trajectory of $x_t^\star$ over $T = 1000$ iterations is shown in red. We also depict the trajectory of the estimator $\mathbf{x}_{i,t}$ (shown in blue) for $i \in \{1, 6, 12, 23\}$ and observe that it closely follows $x_t^\star$ in every case.

$i \in \{1, 6, 12, 23\}$. While the dynamic regret can be controlled in the expectation sense (see Theorem 5), Fig. 2 suggests that agents' estimators closely follow the trajectory of the moving target with high probability.

## V. Conclusion

This work generalizes a number of results in the literature by addressing *decentralized online* optimization in *dynamic* environments. We considered tracking the minimizer of a global time-varying convex function via a network of agents. The minimizer of the global function follows a linear dynamics whose state matrix is known to agents, but an unknown unstructured noise causes deviation from the linear trajectory. The global function can be written as a sum of local functions at each time step, and each agent can only observe its associated local function. However, these local functions appear sequentially, and agents do not have prior knowledge of the future cost functions. Our proposed algorithm for this setup can be cast as a decentralized version of the mirror descent algorithm. However, the algorithm possesses two additional steps to take into consideration agents' interactions and dynamics of the minimizer. We used a notion of network dynamic regret to measure the performance of our algorithm versus its offline counterpart. We established that the regret bound scales inversely in the spectral gap of the network and captures the deviation of minimizer sequence with respect to the given dynamics. We showed that our generalization is valid and convincing in the sense that the results recover those of distributed optimization in both online and offline settings.

These results open a few new directions for future research. We conjecture that our theoretical results can be strengthened once agents receive multiple gradients per time step. Also, the result of Corollary 4 assumes that the step size is tuned in advance. This would require the knowledge of $C_T$ or an upper bound on the quantity. For the centralized setting, one can potentially avoid the issue using doubling tricks, which requires online accumulation of the mismatch noise $v_t$. However, it is more natural to consider that this noise is not fully observable in the decentralized setting. Therefore, an adaptive solution to step-size tuning remains open for the future investigation.

## Appendix

The following lemma is standard in the analysis of mirror descent. We state the lemma here and use it in our analysis later.

*Lemma 6 (see [20]):* Let $\mathcal{X}$ be a convex set in a Banach space $\mathcal{B}$, $\mathcal{R} : \mathcal{B} \to \mathbb{R}$ denote a 1-strongly convex function on $\mathcal{X}$ with respect to a norm $\|\cdot\|$, and $\mathcal{D}_\mathcal{R}(\cdot,\cdot)$ represent the Bregman divergence with respect to $\mathcal{R}$, respectively. Then, any update of the form

$$\hat{x} = \text{argmin}_{x\in\mathcal{X}}\left\{\langle a, x\rangle + \mathcal{D}_\mathcal{R}(x, c)\right\}$$

satisfies the following inequality:

$$\langle \hat{x} - d, a\rangle \le \mathcal{D}_\mathcal{R}(d, c) - \mathcal{D}_\mathcal{R}(d, \hat{x}) - \mathcal{D}_\mathcal{R}(\hat{x}, c)$$

for any $d \in \mathcal{X}$.

### A. Proof of Lemma 1

*A. Proof of Lemma 1:* Applying Lemma 6 to the update (8a), we get

$$\eta_t\langle \hat{x}_{i,t+1} - y_{i,t}, \nabla_{i,t}\rangle \le -\mathcal{D}_\mathcal{R}(y_{i,t}, \hat{x}_{i,t+1}) - \mathcal{D}_\mathcal{R}(\hat{x}_{i,t+1}, y_{i,t}).$$

As pointed out in (7), in view of the strong convexity of $\mathcal{R}$, the Bregman divergence satisfies $\mathcal{D}_\mathcal{R}(x, y) \ge \frac{1}{2}\|x - y\|^2$ for any $x, y \in \mathcal{X}$. Therefore, we can simplify the equation above as follows:

$$\begin{aligned}\eta_t\langle y_{i,t} - \hat{x}_{i,t+1}, \nabla_{i,t}\rangle &\ge \mathcal{D}_\mathcal{R}(y_{i,t}, \hat{x}_{i,t+1}) + \mathcal{D}_\mathcal{R}(\hat{x}_{i,t+1}, y_{i,t}) \\ &\ge \|y_{i,t} - \hat{x}_{i,t+1}\|^2 .\end{aligned} \tag{12}$$

On the other hand, for any primal-dual norm pair, it holds that

$$\begin{aligned}\langle y_{i,t} - \hat{x}_{i,t+1}, \nabla_{i,t}\rangle &\le \|y_{i,t} - \hat{x}_{i,t+1}\| \, \|\nabla_{i,t}\|_* \\ &\le L\,\|y_{i,t} - \hat{x}_{i,t+1}\|\end{aligned}$$

using Assumption 1 in the last line. Combining above with (12), we obtain

$$\|y_{i,t} - \hat{x}_{i,t+1}\| \le L\eta_t. \tag{13}$$

Letting $e_{i,t} := \hat{x}_{i,t+1} - y_{i,t}$, we can now rewrite update (8b) as $\hat{x}_{i,t+1} = \sum_{j=1}^{n}[W]_{ij}x_{j,t} + e_{i,t}$, which implies

$$x_{i,t+1} = A\hat{x}_{i,t+1} = \sum_{j=1}^{n}[W]_{ij}Ax_{j,t} + Ae_{i,t}. \tag{14}$$

Noting that the algorithm is initialized at $\mathbb{0}$ and using Assumption 2 (doubly stochasticity of $W$), the above immediately yields

$$\bar{x}_{t+1} := \frac{1}{n}\sum_{i=1}^{n} x_{i,t+1} = A\bar{x}_t + A\bar{e}_t = \sum_{\tau=0}^{t} A^{t+1-\tau}\bar{e}_\tau \tag{15}$$

where $\bar{e}_t := \frac{1}{n}\sum_{i=1}^{n} e_{i,t}$. On the other hand, stacking the local vectors $x_{i,t}$ and $e_{i,t}$ in (14) in the following form:

$$x_t := [x_{1,t}^\top, x_{2,t}^\top, \ldots, x_{n,t}^\top]^\top \quad e_t := [e_{1,t}^\top, e_{2,t}^\top, \ldots, e_{n,t}^\top]^\top$$

and using $\otimes$ to denote the Kronecker product, we can write (14) in the matrix format as $x_{t+1} = (W \otimes A)x_t + (I_n \otimes A)e_t$, which implies

$$x_{i,t+1} = \sum_{\tau=0}^{t}\sum_{j=1}^{n}\left[W^{t-\tau}\right]_{ij} A^{t+1-\tau}e_{j,\tau}.$$

Combining above with (15), we derive

$$x_{i,t+1} - \bar{x}_{t+1} = \sum_{\tau=0}^{t}\sum_{j=1}^{n}\left(\left[W^{t-\tau}\right]_{ij} - \frac{1}{n}\right) A^{t+1-\tau}e_{j,\tau}$$

which entails

$$\|x_{i,t+1} - \bar{x}_{t+1}\| \le \sum_{\tau=0}^{t}\sum_{j=1}^{n}\left|\left[W^{t-\tau}\right]_{ij} - \frac{1}{n}\right| L\eta_\tau, \tag{16}$$

where we used $\|e_{i,\tau}\| \le L\eta_\tau$ obtained in (13) as well as the assumption $\|A\| \le 1$ (see Assumption 5). By standard properties of doubly stochastic matrices (see, e.g., [42]), the matrix $W$ satisfies

$$\sum_{j=1}^{n}\left|\left[W^{t}\right]_{ij} - \frac{1}{n}\right| \le \sqrt{n}\sigma_2^t(W).$$

Substituting above into (16) completes the proof. ■

***B. Proof of Lemma 2:*** We start by adding, subtracting, and regrouping several terms as follows:

$$\begin{aligned}\frac{1}{\eta_t}&\mathcal{D}_{\mathcal{R}}(x_t^\star, y_{i,t}) - \frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) = \\ &+\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star, y_{i,t}) - \frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(x_{t+1}^\star, y_{i,t+1}) \\ &+\frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(x_{t+1}^\star, y_{i,t+1}) - \frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, y_{i,t+1}) \\ &+\frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, y_{i,t+1}) - \frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &+\frac{1}{\eta_{t+1}}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) - \frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}).\end{aligned} \tag{17}$$

We now need to bound each of the four terms above. For the second term, we note that

$$\mathcal{D}_{\mathcal{R}}(x_{t+1}^\star, y_{i,t+1}) - \mathcal{D}_{\mathcal{R}}(Ax_t^\star, y_{i,t+1}) \le K\left\|x_{t+1}^\star - Ax_t^\star\right\| \tag{18}$$

by the Lipschitz condition on the Bregman divergence (see Assumption 4). Also, by the separate convexity of the Bregman divergence (see Assumption 3) as well as stochasticity of $W$ (see Assumption 2), we have

$$\begin{aligned}&\sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, y_{i,t+1}) - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &= \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}\left(Ax_t^\star, \sum_{j=1}^{n}[W]_{ij}x_{j,t+1}\right) - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &\le \sum_{i=1}^{n}\sum_{j=1}^{n}[W]_{ij}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, x_{j,t+1}) - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &= \sum_{j=1}^{n}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, x_{j,t+1})\sum_{i=1}^{n}[W]_{ij} - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &= \sum_{j=1}^{n}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, x_{j,t+1}) - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \\ &= \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(Ax_t^\star, A\hat{x}_{i,t+1}) - \sum_{i=1}^{n}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{x}_{i,t+1}) \le 0\end{aligned} \tag{19}$$

where the last inequality follows from the fact that $A$ is nonexpansive (see Assumption 5). When summing (17) over $t \in [T]$, the first term telescopes, while the second and third terms are handled with the bounds in (18) and (19), respectively. Recalling from the statement of the lemma that $R^2 = \sup_{x,y\in\mathcal{X}}\mathcal{D}_{\mathcal{R}}(x, y)$, we obtain

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,y_{i,t})-\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,\hat{x}_{i,t+1})\right)&\\ \leq\frac{R^2}{\eta_1}+\sum_{t=1}^{T}\frac{K}{\eta_{t+1}}\left\|x_{t+1}^\star-Ax_t^\star\right\|&\\ +R^2\sum_{t=1}^{T}\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}&\\ \leq\frac{2R^2}{\eta_{T+1}}+\sum_{t=1}^{T}\frac{K}{\eta_{t+1}}\left\|x_{t+1}^\star-Ax_t^\star\right\|&\end{aligned}$$

where we used the fact that the step size is positive and nonincreasing in the last line. ■

***C. Auxiliary Lemma:*** In the proof of Theorem 3, we make use of another technical lemma provided below.

*Lemma 7:* Let all the Assumptions 1–5 hold as stated in Theorem 3. Then, for the local estimates $\{x_{i,t}\}_{t=1}^T$ generated by the updates (8a), (8b), it holds that

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\Big(f_{i,t}(x_{i,t})-f_{i,t}(x_t^\star)\Big)&\\ \leq\frac{2R^2}{\eta_{T+1}}+\sum_{t=1}^{T}\frac{K}{\eta_{t+1}}\left\|x_{t+1}^\star-Ax_t^\star\right\|+L^2\sum_{t=1}^{T}\frac{\eta_t}{2}&\\ +2L^2\sqrt{n}\sum_{t=1}^{T}\sum_{\tau=0}^{t-1}\eta_\tau\sigma_2^{t-\tau-1}(W)&\end{aligned}$$

where $R^2:=\sup_{x,y\in\mathcal{X}}\mathcal{D}_{\mathcal{R}}(x,y)$.

*Proof:* In view of the convexity of $f_{i,t}(\cdot)$, we have

$$\begin{aligned}f_{i,t}(x_{i,t})-f_{i,t}(x_t^\star)&\leq\langle\nabla_{i,t},x_{i,t}-x_t^\star\rangle\\ &=\langle\nabla_{i,t},\hat{x}_{i,t+1}-x_t^\star\rangle+\langle\nabla_{i,t},x_{i,t}-y_{i,t}\rangle\\ &\quad+\langle\nabla_{i,t},y_{i,t}-\hat{x}_{i,t+1}\rangle \qquad (20)\end{aligned}$$

for any $i\in[n]$. We now need to bound each of the three terms on the right-hand side of (20). Starting with the last term and using boundedness of gradients (see Assumption 1), we have that

$$\begin{aligned}\langle\nabla_{i,t},y_{i,t}-\hat{x}_{i,t+1}\rangle&\leq\|y_{i,t}-\hat{x}_{i,t+1}\|\,\|\nabla_{i,t}\|_*\\ &\leq L\|y_{i,t}-\hat{x}_{i,t+1}\|\\ &\leq\frac{1}{2\eta_t}\|y_{i,t}-\hat{x}_{i,t+1}\|^2+\frac{\eta_t}{2}L^2 \qquad (21)\end{aligned}$$

where the last line is due to arithmetic mean–geometric mean inequality. Next, we recall update (8b) to bound the second term in (20) using Assumptions 1 and 2 as

$$\begin{aligned}\langle\nabla_{i,t},x_{i,t}-y_{i,t}\rangle&=\langle\nabla_{i,t},x_{i,t}-\bar{x}_t+\bar{x}_t-y_{i,t}\rangle\\ &=\langle\nabla_{i,t},x_{i,t}-\bar{x}_t\rangle\\ &\quad+\sum_{j=1}^{n}[W]_{ij}\langle\nabla_{i,t},\bar{x}_t-x_{j,t}\rangle\\ &\leq L\|x_{i,t}-\bar{x}_t\|+L\sum_{j=1}^{n}[W]_{ij}\|x_{j,t}-\bar{x}_t\|\\ &\leq 2L^2\sqrt{n}\sum_{\tau=0}^{t-1}\eta_\tau\sigma_2^{t-\tau-1}(W) \qquad (22)\end{aligned}$$

where in the last line we appealed to Lemma 1. Finally, we apply Lemma 6 to (20) to get

$$\begin{aligned}\langle\nabla_{i,t},\hat{x}_{i,t+1}-x_t^\star\rangle&\leq\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,y_{i,t})-\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,\hat{x}_{i,t+1})\\ &\quad-\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(\hat{x}_{i,t+1},y_{i,t})\\ &\leq\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,y_{i,t})-\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,\hat{x}_{i,t+1})\\ &\quad-\frac{1}{2\eta_t}\|\hat{x}_{i,t+1}-y_{i,t}\|^2 \qquad (23)\end{aligned}$$

since the Bregman divergence satisfies $\mathcal{D}_{\mathcal{R}}(x,y)\geq\frac{1}{2}\|x-y\|^2$ for any $x,y\in\mathcal{X}$. Substituting (21)–(23) into the bound (20), we derive

$$\begin{aligned}f_{i,t}(x_{i,t})-f_{i,t}(x_t^\star)&\leq\frac{\eta_t}{2}L^2+2L^2\sqrt{n}\sum_{\tau=0}^{t-1}\eta_\tau\sigma_2^{t-\tau-1}(W)\\ &\quad+\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,y_{i,t})-\frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star,\hat{x}_{i,t+1}). \qquad (24)\end{aligned}$$

Summing over $t\in[T]$ and averaging over $i\in[n]$ and applying Lemma 2 completes the proof. ■

***D. Proof of Theorem 3:*** To bound the regret defined in (5), we start with

$$\begin{aligned}f_t(x_{i,t})-f_t(x_t^\star)&=f_t(x_{i,t})-f_t(\bar{x}_t)+f_t(\bar{x}_t)-f_t(x_t^\star)\\ &\leq L\|x_{i,t}-\bar{x}_t\|+f_t(\bar{x}_t)-f_t(x_t^\star)\\ &=\frac{1}{n}\sum_{j=1}^{n}f_{j,t}(\bar{x}_t)-\frac{1}{n}\sum_{j=1}^{n}f_{j,t}(x_t^\star)\\ &\quad+L\|x_{i,t}-\bar{x}_t\|\end{aligned}$$

where we used the Lipschitz continuity of $f_t(\cdot)$ (see Assumption 1) in the second line. Using the Lipschitz continuity of $f_{i,t}(\cdot)$ for $i \in [n]$, we simplify above as follows:

$$f_t(x_{i,t}) - f_t(x_t^\star) \leq \frac{1}{n}\sum_{j=1}^n f_{j,t}(x_{j,t}) - \frac{1}{n}\sum_{j=1}^n f_{j,t}(x_t^\star) + L\left\|x_{i,t} - \bar{x}_t\right\| + \frac{L}{n}\sum_{j=1}^n \left\|x_{j,t} - \bar{x}_t\right\|. \tag{25}$$

Summing over $t \in [T]$ and $i \in [n]$ and applying Lemmas 1 and 7 completes the proof. ■

***E. Proof of Theorem 5:*** We need to rework the proof of Theorem 3 and replace the true gradients by the stochastic gradients. Following the lines in the proof of Lemma 1, (13) will be changed to

$$\left\|\mathbf{y}_{i,t} - \hat{\mathbf{x}}_{i,t+1}\right\| \leq \eta_t \left\|\boldsymbol{\nabla}_{i,t}\right\|_*$$

yielding

$$\left\|\mathbf{x}_{i,t+1} - \bar{\mathbf{x}}_{t+1}\right\| \leq \sqrt{n}\sum_{\tau=0}^t \eta_\tau \left\|\boldsymbol{\nabla}_{i,\tau}\right\|_* \sigma_2^{t-\tau}(W). \tag{26}$$

On the other hand, at the beginning of Lemma 7, we should use the stochastic gradient as

$$\begin{aligned} f_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(x_t^\star) &\leq \langle \nabla_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle \\ &= \langle \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle + \langle \nabla_{i,t} - \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle \\ &= \langle \boldsymbol{\nabla}_{i,t}, \hat{\mathbf{x}}_{i,t+1} - x_t^\star\rangle + \langle \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - \mathbf{y}_{i,t}\rangle \\ &\quad + \langle \boldsymbol{\nabla}_{i,t}, \mathbf{y}_{i,t} - \hat{\mathbf{x}}_{i,t+1}\rangle + \langle \nabla_{i,t} - \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle. \end{aligned}$$

As in Lemma 1, any bound involving $L$ that was originally an upper bound on the exact gradient must be replaced by the norm of stochastic gradient, which changes inequality (24) to

$$\begin{aligned} & f_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(x_t^\star) \leq \\ & \frac{\eta_t}{2}\left\|\boldsymbol{\nabla}_{i,t}\right\|_*^2 + \sqrt{n}\left\|\boldsymbol{\nabla}_{i,t}\right\|_* \sum_{\tau=0}^{t-1}\eta_\tau \left\|\boldsymbol{\nabla}_{i,\tau}\right\|_* \sigma_2^{t-\tau-1}(W) \\ & + \sqrt{n}\left\|\boldsymbol{\nabla}_{i,t}\right\|_* \sum_{j=1}^n [W]_{ij}\sum_{\tau=0}^{t-1}\eta_\tau \left\|\boldsymbol{\nabla}_{j,\tau}\right\|_* \sigma_2^{t-\tau-1}(W) \\ & + \frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star, \mathbf{y}_{i,t}) - \frac{1}{\eta_t}\mathcal{D}_{\mathcal{R}}(x_t^\star, \hat{\mathbf{x}}_{i,t+1}) \\ & + \langle \nabla_{i,t} - \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle. \end{aligned}$$

Then, taking expectation from above, using law of total expectation, and noting that $\mathbf{x}_{i,t} \in \mathcal{F}_{t-1}$, for the last term, we obtain

$$\begin{aligned} & \mathbb{E}\left[\langle \nabla_{i,t} - \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\langle \nabla_{i,t} - \boldsymbol{\nabla}_{i,t}, \mathbf{x}_{i,t} - x_t^\star\rangle \middle| \mathcal{F}_{t-1}\right]\right] \\ &= \mathbb{E}\left[\langle \mathbb{E}\left[\nabla_{i,t} - \boldsymbol{\nabla}_{i,t} \middle| \mathcal{F}_{t-1}\right], \mathbf{x}_{i,t} - x_t^\star\rangle\right] \\ &= 0 \end{aligned}$$

using condition (9). By the bounded second moment condition (9), we also have

$$\mathbb{E}\left[\left\|\boldsymbol{\nabla}_{i,t}\right\|_*^2\right] \leq G^2 \qquad \mathbb{E}\left[\left\|\boldsymbol{\nabla}_{i,t}\right\|_* \left\|\boldsymbol{\nabla}_{j,\tau}\right\|_*\right] \leq G^2$$

where the second inequality is due to Cauchy–Schwarz inequality. Summing over $i \in [n]$ and $t \in [T]$, we apply Lemma 2 to get the same as bound as Lemma 7, except that $L$ is replaced by $G$. Then, the proof is finished once we return to (25). ■

## References


[1] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

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

[3] V. Lesser, C. L. Ortiz Jr, and M. Tambe, *Distributed Sensor Networks: A multiagent Perspective*, vol. 9. New York, NY, USA: Springer, 2012.

[4] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Trans. Automat. Control*, vol. 61, no. 11, pp. 3256–3268, Nov. 2016.

[5] A. Nedic, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *IEEE Trans. Automat. Control*, 2017, to be published.

[6] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Cambridge, MA, USA, Massachusetts Inst. Technol., 1984.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[8] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Automat. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[9] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[10] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Automat. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[11] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Trans. Automat. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[12] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.

[13] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 928–935.

[14] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[15] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 398–406.

[16] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *Proc. IEEE Conf. Decision Control*, 2013, pp. 1484–1489.

[17] D. Mateos-Núnez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 1, pp. 23–37, Jan.–Jun. 2014.

[18] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Trans. Control Netw. Syst.*, 2015, to be published.

[19] D. Yudin and A. Nemirovskii, *Problem Complexity and Method Efficiency in Optimization*. Hoboken, NJ, USA: Wiley, 1983.

[20] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, 2003.

[21] H. H. Bauschke and J. M. Borwein, "Joint and separate convexity of the Bregman distance," *Stud. Comput. Math.*, vol. 8, pp. 23–36, 2001.

[22] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, vol. 177, pp. 643–650, 2016.

[23] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *Proc. IEEE 6th Int. Workshop Comput. Adv. Multi-Sens. Adaptive Process.*, 2015, pp. 517–520.
[24] M. Raginsky and J. Bouvrie, "Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 6793–6800.
[25] A. Nedić, S. Lee, and M. Raginsky, "Decentralized online optimization with global objectives and local communication," in *Proc. IEEE Amer. Control Conf.*, 2015, pp. 4497–4503.
[26] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Oper. Res.*, vol. 63, no. 5, pp. 1227–1244, 2015.
[27] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *Proc. IEEE Conf. Decision Control*, 2016, pp. 7195–7201.
[28] C.-J. Lee, C.-K. Chiang, and M.-E. Wu, "Resisting dynamic strategies in gradually evolving worlds," in *Proc. IEEE 3rd Int. Conf. Robot, Vision Signal Process.*, 2015, pp. 191–194.
[29] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *arXiv preprint arXiv:1608.07544*, 2016.
[30] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591, Sep. 2016.
[31] T. Yang, L. Zhang, R. Jin, and J. Yi, "Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 449–457.
[32] S. Shahrampour and A. Jadbabaie, "An online optimization approach for multi-agent tracking of dynamic parameters in the presence of adversarial noise," in *Proc. IEEE Amer. Control Conf.*, 2017, pp. 3306–3311.
[33] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE Conf. Decision Control*, 2007, pp. 5492–5498.
[34] D. Gu, "Distributed particle filter for target tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3856–3861.
[35] U. Khan *et al.*, "On connectivity, observability, and stability in distributed estimation," in *Proc. IEEE Conf. Decision Control*, 2010, pp. 6639–6644.
[36] L. Qipeng, Z. Jiuhua, and W. Xiaofan, "Distributed detection via Bayesian updates and consensus," in *Proc. IEEE 34th Chin. Control Conf.*, 2015, pp. 6992–6997.
[37] L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou, "Improved dynamic regret for non-degeneracy functions," *arXiv preprint arXiv:1608.03933*, 2016.
[38] Y. Bar-Shalom, *Tracking and Data Association*. San Diego, CA, USA: Academic, 1987.
[39] D. Acemoglu, A. Nedić, and A. Ozdaglar, "Convergence of rule-of-thumb learning rules in social networks," in *Proc. IEEE Conf. Decision Control*, 2008, pp. 1714–1720.
[40] S. Shahrampour, S. Rakhlin, and A. Jadbabaie, "Online learning of dynamic parameters in social networks," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2013–2021.
[41] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed estimation of dynamic parameters: Regret analysis," in *Proc. Amer. Control Conf.*, Jul. 2016, pp. 1066–1071.
[42] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

**Shahin Shahrampour** received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2009, and the M.S.E. degree in electrical engineering, the M.A. degree in statistics, and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2012, 2014, and 2015, respectively.

He is currently a Postdoctoral Fellow in electrical engineering with Harvard University, Cambridge, MA, USA. His research interests include machine learning, online learning, optimization, and distributed algorithms.

**Ali Jadbabaie** (F'14) received the B.S. degree from Sharif University of Technology, Tehran, Iran, the M.S. degree in electrical and computer engineering from the University of New Mexico, Albuquerque, NM, USA, and the Ph.D. degree in control and dynamical systems from California Institute of Technology, Pasadena, CA, USA.

He is the JR East Professor of engineering and Associate Director of the Institute for Data, Systems and Society, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where he is a Professor with the Department of Civil and Environmental Engineering and the Laboratory for Information and Decision Systems. He also serves as the Director of the Sociotechnical Systems Research Center, MIT. He was a Postdoctoral Scholar at Yale University before joining the faculty at the University of Pennsylvania in July 2002. Prior to joining MIT faculty, he was the Alfred Fitler Moore a Professor of Network Science and held secondary appointments in computer and information science and operations, information, and decisions in the Wharton School. His current research interests include the interplay of dynamic systems and networks with a specific emphasis on multiagent coordination and control, distributed optimization, network science, and network economics.

Prof. Jadbabaie is the former inaugural Editor-in-Chief of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, a new interdisciplinary journal sponsored by several IEEE societies. He has received a National Science Foundation Career Award, an Office of Naval Research Young Investigator Award, the O. Hugo Schuck Best Paper Award from the American Automatic Control Council, and the George S. Axelby Best Paper Award from the IEEE Control Systems Society. His students have been winners and finalists of student best paper awards at various ACC and CDC conferences. He received the Vannevar Bush Fellowship (formerly known as National Security Science and a Engineering Faculty Fellowship) from the Office of Secretary of Defense in 2016.