Stochastic Subgradient Algorithms for Strongly Convex Optimization Over Distributed Networks

Muhammed O. Sayin, N. Denizcan Vanli, Suleyman S. Kozat, *Senior Member, IEEE*, and Tamer Başar, *Life Fellow, IEEE*

Abstract—We study diffusion and consensus based optimization of a sum of unknown convex objective functions over distributed networks. The only access to these functions is through stochastic gradient oracles, each of which is only available at a different node; and a limited number of gradient oracle calls is allowed at each node. In this framework, we introduce a convex optimization algorithm based on stochastic subgradient descent (SSD) updates. We use a carefully designed time-dependent weighted averaging of the SSD iterates, which yields a convergence rate of $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient updates for each node on a network of N nodes, where $0 \leq \sigma < 1$ denotes the second largest singular value of the communication matrix. This rate of convergence matches the performance lower bound up to constant terms. Similar to the SSD algorithm, the computational complexity of the proposed algorithm also scales linearly with the dimensionality of the data. Furthermore, the communication load of the proposed method is the same as the communication load of the SSD algorithm. Thus, the proposed algorithm is highly efficient in terms of complexity and communication load. We illustrate the merits of the algorithm with respect to the state-of-art methods over benchmark real life data sets.

Index Terms—Distributed processing, convex optimization, online learning, diffusion strategies, consensus strategies

1 Introduction

The demand for large-scale networks consisting of multiple agents (i.e., nodes) [1] with different objectives is steadily growing due to their increased efficiency and scalability compared to centralized distributed structures [2], [3], [4], [5], [6]. A wide range of problems in the context of distributed and parallel processing can be considered as a minimization of a sum of objective functions, where each function (or information on each function) is available only to a single agent or node [7], [8], [9]. In such practical applications, it is essential to process the information in a decentralized manner since transferring the objective functions as well as the entire resources (e.g., data) may not be feasible or possible [10], [11], [12], [13]. For example, in a distributed data mining scenario, privacy considerations may prohibit sharing of the objective functions [7], [8], [9]. Similarly, in a distributed wireless network, energy considerations may limit the communication rate between agents [14], [15], [16], [17]. In such settings, parallel or distributed processing algorithms, where each node performs its own processing and shares information subsequently, are preferable over the centralized methods [18], [19], [20], [21].

Here, we consider minimization of a sum of unknown convex objective functions, where each agent (or node) observes only its particular objective function via the stochastic gradient oracles. Particularly, we seek to minimize this sum of functions with a limited number of gradient oracle calls at each agent. In this framework, we introduce a distributed online convex optimization algorithm based on stochastic subgradient descent (SSD) iterates that efficiently minimizes this cost function. Specifically, each agent uses a time-dependent weighted combination of the SSD iterates and achieves the presented performance guarantees, which matches the lower bounds presented in [22], only with a relatively small excess term caused by the unknown network model. The proposed method is comprehensive, in that any communication strategy, such as the diffusion [3] and the consensus [6] strategies, are incorporated into our algorithm in a straightforward manner as shown in the paper. We compare the performance of our algorithm with respect to the state-of-the-art methods [6], [11], [23] in the literature and present substantial performance improvements for various well-known network topologies and benchmark data sets.

The distributed network framework is successfully used in wireless sensor networks [24], [25], [26], [27], [28], [29], as well as for convex optimization via projected subgradient techniques [6], [7], [8], [9], [10], [11]. In [11], the authors demonstrate the performance of the least mean squares (LMS) algorithm over distributed networks using different diffusion strategies. We emphasize that this problem can also be cast as a distributed convex optimization problem, and hence our results here can be applied to these problems

- *M. O. Sayin and T. Başar are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801. E-mail: {sayin2, basar1}@illinois.edu.*
- *N. D. Vanli is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139. E-mail: denizcan@mit.edu.*
- *S. S. Kozat is with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey. E-mail: kozat@ee.bilkent.edu.tr.*

Manuscript received 12 Jan. 2016; revised 11 May 2017; accepted 1 June 2017. Date of publication 7 June 2017; date of current version 11 Dec. 2017. (Corresponding author: Muhammed O. Sayin.) Recommended for acceptance by J. Cortés. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TNSE.2017.2713396

in a straightforward manner. In [10], the authors consider the cooperative optimization of the cost function under convex inequality constraints. However, the problem formulation as well as the convergence results in this paper are substantially different from the ones in [10]. In particular, in [10], agents seek to minimize an approximation of the original optimization problem through penalty functions while we directly consider the original optimization problem. Furthermore, Reference [10] provides an upper bound on the mean square error as the number of iterates goes to infinity for sufficiently small step sizes. Yet that upper bound goes to zero as the step size goes to zero. On the other hand, here, we not only show that through the proposed approach each agent achieves the minimum cost for a certain step size, but also provide an upper bound on the convergence rate, while that upper bound matches the lower bound provided in [22] up to constant terms.

In [2], [6], the authors present a (constrained in [6] and unconstrained in [2]) deterministic analysis of the SSD iterates and our results build on them by illustrating a stronger convergence bound in expectation while also providing MSD analyses of the SSD iterates. Similarly, a regret analysis is conducted for every possible input stream in an online and distributed manner in [30] for general convex cost functions; and in [31] under Lipschitz continuous and strongly convex cost functions, where the latter achieves a regret diminishing at a faster rate of $O(\log(T)/T)$ (after T iterates). On the contrary, we study the distributed online convex optimization problem in the expectation sense (with respect to the data statistics), i.e., not in an individual sequence manner, where we show that SSD iterates achieve the optimal convergence rate of $O(1/T)$. In [7], [8], [9], the authors consider the distributed convex optimization problem and present probability-1 and mean square convergence results of the SSD iterates. In this paper, on the other hand, we provide the expected convergence rate of our algorithm and the mean square deviation (MSD) of the SSD iterates at any time instant.

Similar convergence analyses have recently been carried out in the computational learning theory literature [22], [23], [32], [33], [34]. In [32], the authors provide deterministic bounds on the learning performance (i.e., regret) of the SSD algorithm. In [33], these analyses are extended and a regret-optimal learning algorithm is proposed. Along similar lines, in [23], the authors describe a method to make the SSD algorithm optimal for strongly convex optimization. However, these approaches rely on the smoothness of the optimization problem. In [34], a different method to achieve the optimal convergence rate is proposed and its performance is analyzed. In this paper, however, convex optimization is performed over a network of localized learners, unlike in [23], [32], [33], [34]. Our results entail convergence rates over any unknown communication graph, and in this sense build upon the analyses of the centralized learners. Furthermore, unlike [23], [33], our algorithm does not require the optimization problem to be sufficiently smooth.

Distributed convex optimization appears in a wide range of practical applications in wireless sensor networks and real-time control systems [3], [4], [5]. We introduce a comprehensive approach to this setup by proposing an online algorithm, whose expected performance is asymptotically the same as the performance of the optimal centralized processor. Our results are generic for any probability distribution on the data, not necessarily Gaussian, unlike the conventional works in the literature [11], [12]. Our experiments over different network topologies, various data sets and cost functions demonstrate the superiority and robustness of our approach with respect to the state-of-the-art methods in the literature.

Our main contributions can thus be summarized as follows.

1) We introduce a distributed online convex optimization algorithm based on SSD iterates, which achieves an optimal convergence rate of $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient updates, for each and every node of the network, where N is the number of nodes. We emphasize that this convergence rate is optimal since it achieves the lower bounds presented in [22] up to constant terms.
2) We show that MSD between the time weighted average and the optimal solution is also upper bounded by $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient updates while MSD between the average of the iterates (which can be attained if the agents continue to exchange information without gradient updates) and the optimal solution is upper bounded by $O\left(\frac{\sqrt{N}}{(1-\sigma)T}\right)$.
3) Our analyses can be extended to analyze the performances of the diffusion and consensus strategies in a straightforward manner as illustrated in the paper.
4) We demonstrate that the algorithm introduced outperforms the state-of-the-art methods in terms of normalized accumulated error and MSD from the optimal solution under various network topologies and benchmark data sets.

The organization of the paper is as follows. In Section 2, we introduce the distributed convex optimization framework and provide the notations. We then introduce the main result of the paper, i.e., an SSD based convex optimization algorithm, in Section 3 and analyze the convergence rate of the algorithm. In Section 4, we demonstrate the performance of our algorithm with respect to the state-of-the-art methods through simulations and then conclude the paper with several remarks in Section 5.

2 Problem Formulation

2.1 Notation and Preliminaries

Throughout the paper, all vectors are column vectors and represented by boldface lowercase letters. Matrices are represented by boldface uppercase letters. For a matrix $\boldsymbol{H}$, $||\boldsymbol{H}||_F$ is the Frobenius norm. For a vector $\boldsymbol{x}$, $||\boldsymbol{x}|| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$ is the ℓ^2-norm. $\mathbf{0}$ (and $\mathbf{1}$) denotes a vector with all zero (and one) elements and the dimensions can be understood from the context. For a matrix $\boldsymbol{H}$, $\boldsymbol{H}_{ij}$ represents its entry at the ith row and jth column.

For a non-empty, closed and convex set $\mathcal{W} \subset \mathbb{R}^m$, $\Pi_{\mathcal{W}}$ denotes the Euclidean projection onto $\mathcal{W}$, i.e.,

$$\Pi_{\mathcal{W}}(\boldsymbol{w}_0) = \arg\min_{\boldsymbol{w}\in\mathcal{W}} ||\boldsymbol{w} - \boldsymbol{w}_0||. \tag{1}$$

We say that a convex function (possibly non-smooth) $f : \mathbb{R}^m \to \mathbb{R}$ on the convex domain $\mathcal{W}$ has the subgradient set $\partial f(\cdot) \subset \mathbb{R}^m$ at a point $\boldsymbol{w}_0 \in \mathcal{W}$ if

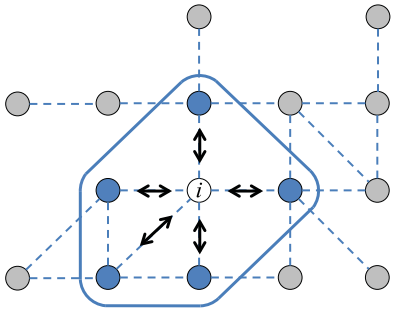


Fig. 1. The neighborhood of agent-i over the distributed network.

$$\boldsymbol{g} \in \partial f(\boldsymbol{w}_0) \Leftrightarrow f(\boldsymbol{w}) \geq f(\boldsymbol{w}_0) + \boldsymbol{g}^T(\boldsymbol{w} - \boldsymbol{w}_0)\ \forall \boldsymbol{w} \in \mathcal{W}.$$

Furthermore, we say that f is λ-strongly convex on $\mathcal{W}$ if, and only if, for all $\boldsymbol{w}, \boldsymbol{w}_0 \in \mathcal{W}$ and $\boldsymbol{g} \in \partial f(\boldsymbol{w}_0)$, we have

$$f(\boldsymbol{w}) \geq f(\boldsymbol{w}_0) + \boldsymbol{g}^T(\boldsymbol{w} - \boldsymbol{w}_0) + \frac{\lambda}{2}||\boldsymbol{w} - \boldsymbol{w}_0||^2. \tag{2}$$

2.2 System Overview

Consider a static and connected network of N-agents with processing and communication capabilities. Over the network, each agent has connections with certain other agents, i.e., the ones in his/her neighborhood, and can exchange information with them. We can represent such a network through an undirected graph, where the vertices and the edges correspond to the agents and the communication links between them, respectively, as seen in Fig. 1.

Each agent seeks to minimize $F : \mathbb{R}^m \to \mathbb{R}$, which is a sum of λ-strongly convex (possibly non-smooth) local cost functions $F_i : \mathbb{R}^m \to \mathbb{R}$, for $i = 1, \ldots, N$, i.e., each agent aims to

$$\min_{\boldsymbol{w} \in \mathcal{W}} F(\boldsymbol{w}) = \min_{\boldsymbol{w} \in \mathcal{W}} \sum_{i=1}^{N} F_i(\boldsymbol{w}), \tag{3}$$

where $\mathcal{W} \subseteq \mathbb{R}^m$ is a non-empty, closed and convex set. However, the cost function F is *unknown* to the agents, and each agent i has only access to F via at most T stochastic subgradient oracles[1] of the corresponding local cost function F_i.

Let $(\Omega_i, \mathcal{F}_i, \mathbf{P}_i)$, for $i = 1, \ldots, N$, denote the probability spaces, describing the uncertainty associated with individual agents. Here, Ω_i is the outcome space, $\mathcal{F}_i$ is a suitable σ-algebra over Ω_i, and $\mathbf{P}_i$ is the probability distribution over Ω_i. Furthermore, let $(\Omega, \mathcal{F}, \mathbf{P})$ be the joint probability space over those spaces. After agent-i's call at instant t, for any given point $\boldsymbol{w}_i \in \mathcal{W}$, the gradient oracle *independently* draws a sample $\omega_{t,i} \in \Omega_i$ according to the distribution $\mathbf{P}_i$, and produces a vector $\hat{\boldsymbol{g}}_{t,i}(\omega_{t,i})$ such that

$$\mathbb{E}_{\mathbf{P}_i}\{\hat{\boldsymbol{g}}_{t,i}(\omega_i)\} \in \partial F_i(\boldsymbol{w}_i),$$

where $\partial F_i(\boldsymbol{w}_i)$ denotes the sub-differential set of F_i at $\boldsymbol{w}_i$. For notational simplicity, henceforth, we denote the expectation taken with respect to the probability distribution $\mathbf{P}_i$ by $\mathbb{E}_i\{\cdot\}$ instead of $\mathbb{E}_{\mathbf{P}_i}\{\cdot\}$. Correspondingly, we use $\mathbb{E}\{\cdot\}$ instead of $\mathbb{E}_{\mathbf{P}}\{\cdot\}$.

Although the aim of each agent is to minimize F over $\mathcal{W}$ rather than the local cost F_i, the agents can only call local subgradient oracles $\hat{\boldsymbol{g}}_{i,t}$, for $1 \leq t \leq T$. In particular, other local cost functions are totally unknown. Therefore, the agents exchange information with each other within the neighborhoods to mitigate the access restriction.

2.3 Special Cases

We note that this problem formulation is general enough, covering for example the following scenario as a special case. Consider that the local cost functions are given by $F_i(\boldsymbol{w}_i) = \mathbb{E}_i\{f_i(\omega_i; \boldsymbol{w}_i)\}$, where $f_i(\omega_i; \boldsymbol{w}_i)$ is a certain local loss function, which is a strongly convex function of $\boldsymbol{w}_i$ for any fix $\omega_i \in \Omega_i$. At each instant t, a new sample $\omega_{t,i}$ is drawn from Ω_i independently according to the distribution $\mathbf{P}_i$, and agent-i has access to a corresponding subgradient of $f_i(\omega_{t,i}; \boldsymbol{w}_i)$ at $\boldsymbol{w}_i = \boldsymbol{w}_{t,i}$, i.e.,

$$\hat{\boldsymbol{g}}_{i,t}(\omega_{t,i}) \in \partial f_i(\omega_{t,i}; \boldsymbol{w}_{t,i}).$$

As an example, the local loss function could be as follows:

$$f_i(\omega_i; \boldsymbol{w}_{t,i}) = \ell(\omega_i; \boldsymbol{w}_{t,i}) + \frac{\lambda}{2}\big|\big|\boldsymbol{w}_{t,i}\big|\big|^2, \tag{4}$$

where $\ell(\omega_i; \boldsymbol{w}_{t,i})$ is a Lipschitz-continuous convex loss function with respect to the second variable $\boldsymbol{w}_{t,i}$, which has been extensively studied in the literature [23], [32], [33], [34] as a λ-strongly convex loss function involving regularity terms.[2]

Here, the aim of each agent is to minimize the sum of the expected losses (where the expectations are taken over the random variables ω_i's) over the convex set $\mathcal{W}$. To continue with our example in (4), each agent seeks to minimize

$$\sum_{i=1}^{N} \mathbb{E}_i\{f_i(\omega_i; \boldsymbol{w}_{t,i})\} = \sum_{i=1}^{N} \mathbb{E}_i\{\ell(\omega_i; \boldsymbol{w}_{t,i})\} + \frac{\lambda}{2}\big|\big|\boldsymbol{w}_{t,i}\big|\big|^2. \tag{5}$$

We emphasize that the formulation in (5) covers a wide range of practical loss functions. As an example, for $d_i : \Omega_i \to \mathbb{R}$ and $\boldsymbol{u}_i : \Omega_i \to \mathbb{R}^m$, when $\ell(\omega_i; \boldsymbol{w}_{t,i}) = (d_i(\omega_i) - \boldsymbol{w}_{t,i}^T \boldsymbol{u}_i(\omega_i))^2$, we consider the regularized squared error loss; and when $\ell(\omega_i; \boldsymbol{w}_{t,i}) = \max\{0, 1 - d_i(\omega_i)\boldsymbol{w}_{t,i}^T \boldsymbol{u}_i(\omega_i)\}$, we consider the hinge loss. Since we make no assumptions on the loss function $f_i(\omega_i; \boldsymbol{w}_{t,i})$ other than strong convexity, one can also use different loss functions with their corresponding subgradients and our results would still hold.

3 Main Results

In this section, we present the main results of the paper, where we introduce an algorithm based on the SSD updates, which leads to a rate of convergence bounded from above by $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T iterates, where N is the number of agents (nodes). In particular, the rate of convergence of the algorithm for agent-i is given by

$$\mathbb{E}\{F(\bar{\boldsymbol{w}}_i)\} - \min_{\boldsymbol{w} \in \mathcal{W}} F(\boldsymbol{w}) \leq O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right),$$

where $\bar{\boldsymbol{w}}_i$ is the minimizer produced by agent-i, and the expectation is taken over the randomness of the subgradient

1. The agents have a limited budget to call the gradient oracle.

2. Note that λ in the regularization term is the same with λ in the strong convexity definition (2). In particular, the regularization term ensures that f_i is λ-strongly convex even when ℓ is not strongly convex.

oracles, i.e., with respect to the joint distribution **P**. In order to achieve this performance, the proposed method uses time dependent weighted averages of the SSD updates at each agent together with the adapt-then-combine diffusion strategy [11]. However, as later explained in this section (See Remark 1), our algorithm can be extended to cover consensus in a straightforward manner.

At each time instant t, each agent i has a pre-computed pseudo-solution of problem (3), denoted by $\boldsymbol{w}_{t,i}$. With this pseudo-solution agent-i calls the local subgradient oracle and receives $\hat{\boldsymbol{g}}_{t,i}$. Then, agent-$i$ computes the iterate $\boldsymbol{\phi}_{t+1,i}$ by projecting the SSD update of $\boldsymbol{w}_{t,i}$ onto $\mathcal{W}$ as follows:

$$\boldsymbol{\phi}_{t+1,i} = \Pi_{\mathcal{W}}\big(\boldsymbol{w}_{t,i} - \mu_{t,i}\hat{\boldsymbol{g}}_{t,i}\big),$$

where $\mu_{t,i} > 0$ is a step size. In order to mitigate the access restriction to the other oracles, agent-i exchanges $\boldsymbol{\phi}_{t+1,i}$ with his/her neighbors and computes

$$\boldsymbol{w}_{t+1,i} = \sum_{j=1}^{N} \boldsymbol{H}_{ji}\boldsymbol{\phi}_{t+1,j}, \tag{6}$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is the communication matrix of the graph such that $\boldsymbol{H}_{ji}$'s are the combination weights in (6), and the weight $\boldsymbol{H}_{ji}$ for any i, j is nonzero if, and only if, i and j are neighbors. We assume that $\boldsymbol{H}$ is an irreducible and a periodic doubly stochastic matrix, i.e., $\boldsymbol{H}_{i,j} \geq 0 \ \forall i, j$ and $\boldsymbol{H}\mathbf{1} = \boldsymbol{H}^T\mathbf{1} = \mathbf{1}$. We emphasize that this assumption is not restrictive, and previous analyses in the literature also make similar assumptions [6], [7], [8], [9]. Furthermore, the assumption holds for many communication strategies such as the Metropolis rule [3]. At each instant, agent-i also computes a time-variant weighted average as follows:

$$\bar{\boldsymbol{w}}_{t+1,i} = \frac{t}{t+2}\bar{\boldsymbol{w}}_{t,i} + \frac{2}{t+2}\boldsymbol{w}_{t+1,i}. \tag{7}$$

After consuming the budget to call subgradient oracles, i.e., after T calls, agent-i has $\bar{\boldsymbol{w}}_i = \bar{\boldsymbol{w}}_{T+1,i}$ as the minimizer of F. The complete description of the algorithm can be found in Algorithm 1.

Algorithm 1. Time Variable Weighting (TVW)

Initialize $\bar{\boldsymbol{w}}_{1,i} = \boldsymbol{w}_{1,i} \in \mathcal{W}, \forall i$, arbitrarily.
for $t = 1$ **to** T **do**
  for $i = 1$ **to** N **do**
    Call the subgradient oracle to obtain $\hat{\boldsymbol{g}}_{t,i}$ for $\boldsymbol{w}_{t,i}$.
    $\boldsymbol{\psi}_{t+1,i} = \boldsymbol{w}_{t,i} - \mu_{t,i}\hat{\boldsymbol{g}}_{t,i}$ % SSD update
    $\boldsymbol{\phi}_{t+1,i} = \Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i})$ % Projection
    Exchange $\boldsymbol{\phi}_{t+1,i}$ with the neighbors.
    $\boldsymbol{w}_{t+1,i} = \sum_{j=1}^{N} \boldsymbol{H}_{ji}\boldsymbol{\phi}_{t+1,j}$ % Diffusion
    $\bar{\boldsymbol{w}}_{t+1,i} = \frac{t}{t+2}\bar{\boldsymbol{w}}_{t,i} + \frac{2}{t+2}\boldsymbol{w}_{t+1,i}$ % Weighting
    if $t = T$ **then**
      $\bar{\boldsymbol{w}}_i = \bar{\boldsymbol{w}}_{t+1,i}$ % Solution
    end if
  end for
end for

To achieve the aforementioned result, we first introduce the following lemma, which provides an upper bound on the performance of the average parameter vector.

Lemma 1. *Assume that for any given $\boldsymbol{w} \in \mathcal{W}$, the expected squared norm of any produced subgradient oracle is bounded by G^2, i.e., $\mathbb{E}_i||\hat{\boldsymbol{g}}_i||^2 \leq G^2 \ \forall i$ and $\mu_{t,i} = \mu_t$. Let*

$$\boldsymbol{w}_t := \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{w}_{t,i} \text{ and } \boldsymbol{w}^* := \arg\min_{\boldsymbol{w}\in\mathcal{W}} F(\boldsymbol{w}). \tag{8}$$

Then, Algorithm 1 yields[3]

$$\begin{aligned}\mathbb{E}||\boldsymbol{w}_{t+1} - \boldsymbol{w}^*||^2 &- (1-\lambda\mu_t)\,\mathbb{E}||\boldsymbol{w}_t - \boldsymbol{w}^*||^2 \\ &\leq \frac{2\mu_t}{N}\left[F(\boldsymbol{w}^*) - \mathbb{E}\{F(\boldsymbol{w}_t)\}\right] + 4G^2\mu_t^2 \\ &+ \frac{2\mu_t G}{N}\sum_{i=1}^{N}\left(2\sqrt{\mathbb{E}\big||\boldsymbol{w}_t - \boldsymbol{w}_{t,i}||^2} + \sqrt{\mathbb{E}\big||\boldsymbol{w}_t - \boldsymbol{\psi}_{t+1,i}||^2}\right).\end{aligned}$$

This lemma provides an upper bound on the rate of convergence and the squared deviation of the average parameter vector $\boldsymbol{w}_t$. It provides an intermediate step to relate the performance of the parameter vector at each agent to the best parameter vector. We point out that the assumption in Lemma 1 is practically a boundedness condition that is widely used to analyze the performance of SSD based algorithms [33], [34]. We emphasize that our algorithm does not need to know this upper bound and it is only used in our theoretical derivations.

Proof. In order to efficiently manage the recursions, we first consider the projection operation and let

$$\boldsymbol{x}_{t,i} := \Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i}) - \boldsymbol{\psi}_{t+1,i}. \tag{9}$$

Then, we can compactly represent the averaged estimation parameter $\boldsymbol{w}_t$ (defined in (8)) in a recursive manner as follows [6]

$$\begin{aligned}\boldsymbol{w}_{t+1} &= \frac{1}{N}\sum_{j=1}^{N}\left[\sum_{i=1}^{N}\boldsymbol{H}_{ij}\big(\boldsymbol{w}_{t,i} - \mu_t\hat{\boldsymbol{g}}_{t,i} + \boldsymbol{x}_{t,i}\big)\right] \\ &= \boldsymbol{w}_t + \frac{1}{N}\sum_{i=1}^{N}\big(\boldsymbol{x}_{t,i} - \mu_t\hat{\boldsymbol{g}}_{t,i}\big),\end{aligned} \tag{10}$$

where the last line follows since $\boldsymbol{H}$ is doubly stochastic, i.e., $\boldsymbol{H}\mathbf{1} = \mathbf{1}$.

Hence, the squared deviation of these average iterates with respect to any $\boldsymbol{w}^*$ can be obtained as follows:

$$\begin{aligned}||\boldsymbol{w}_{t+1} - \boldsymbol{w}^*||^2 &= \left|\left|\boldsymbol{w}_t - \boldsymbol{w}^* + \frac{1}{N}\sum_{i=1}^{N}\big(\boldsymbol{x}_{t,i} - \mu_t\hat{\boldsymbol{g}}_{t,i}\big)\right|\right|^2 \\ &= ||\boldsymbol{w}_t - \boldsymbol{w}^*||^2 + \frac{1}{N^2}\left|\left|\sum_{i=1}^{N}(\boldsymbol{x}_{t,i} - \mu_t\hat{\boldsymbol{g}}_{t,i})\right|\right|^2 \\ &\quad + \frac{2}{N}\sum_{i=1}^{N}(\boldsymbol{x}_{t,i} - \mu_t\hat{\boldsymbol{g}}_{t,i})^T(\boldsymbol{w}_t - \boldsymbol{w}^*).\end{aligned} \tag{11}$$

We first upper bound the second term on the right hand side (RHS) of (11) through triangle inequality as follows:

3. Due to SSD update and information exchange, the parameters $\boldsymbol{w}_{t,i}$ and $\boldsymbol{\psi}_{t,i}$, for $t = 1, \ldots, T$ and $i = 1, \ldots, N$, are $\mathcal{F}$-measurable. Therefore, the expectation is taken with respect to the distribution **P**.

$$\frac{1}{N^2}\left|\left|\sum_{i=1}^{N}(\boldsymbol{x}_{t,i}-\mu_t\hat{\boldsymbol{g}}_{t,i})\right|\right|^2 \leq \frac{1}{N^2}\left(\sum_{i=1}^{N}||\boldsymbol{x}_{t,i}||+\mu_t||\hat{\boldsymbol{g}}_{t,i}||\right)^2. \tag{12}$$

We then note that

$$\begin{aligned}||\boldsymbol{x}_{t,i}|| &= ||\Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i})-\boldsymbol{\psi}_{t+1,i}|| \\ &\leq ||\boldsymbol{w}_{t,i}-\boldsymbol{\psi}_{t+1,i}|| \\ &= \mu_t||\hat{\boldsymbol{g}}_{t,i}||,\end{aligned} \tag{13}$$

where the second line follows from the definition of the projection operator (1). Thus, we can rewrite (12) as follows:

$$\frac{1}{N^2}\left|\left|\sum_{i=1}^{N}(\boldsymbol{x}_{t,i}-\mu_t\boldsymbol{g}_{t,i})\right|\right|^2 \leq \frac{4\mu_t^2}{N^2}\left(\sum_{i=1}^{N}||\boldsymbol{g}_{t,i}||\right)^2$$

and taking the expectation of both side with respect to $\mathbf{P}$, we obtain

$$\frac{1}{N^2}\mathbb{E}\left|\left|\sum_{i=1}^{N}(\boldsymbol{x}_{t,i}-\mu_t\boldsymbol{g}_{t,i})\right|\right|^2 \leq 4G^2\mu_t^2, \tag{14}$$

since $\mathbb{E}\{||\hat{\boldsymbol{g}}_{t,i}||||\hat{\boldsymbol{g}}_{t,j}||\} \leq \sqrt{\mathbb{E}||\boldsymbol{g}_{t,i}||^2}\sqrt{\mathbb{E}||\boldsymbol{g}_{t,j}||^2} \leq G^2$ for any $i \neq j$ due to Cauchy-Schwarz inequality and the boundedness assumption.

We next turn our attention to $[-\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*)]$ term in (11) and upper bound this term as follows:

$$\begin{aligned}-\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*) &= -\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}_{t,i}+\boldsymbol{w}_{t,i}-\boldsymbol{w}^*) \\ &\leq -\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}_{t,i})+F_i(\boldsymbol{w}^*)-F_i(\boldsymbol{w}_{t,i}) \\ &\quad -\frac{\lambda}{2}||\boldsymbol{w}^*-\boldsymbol{w}_{t,i}||^2\end{aligned} \tag{15}$$

$$\begin{aligned}&\leq -\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}_{t,i})+\bar{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}_{t,i})+F_i(\boldsymbol{w}^*) \\ &\quad -F_i(\boldsymbol{w}_t)-\frac{\lambda}{2}||\boldsymbol{w}^*-\boldsymbol{w}_{t,i}||^2-\frac{\lambda}{2}||\boldsymbol{w}_{t,i}-\boldsymbol{w}_t||^2\end{aligned} \tag{16}$$

$$\begin{aligned}&\leq F_i(\boldsymbol{w}^*)-F_i(\boldsymbol{w}_t)+\left(||\bar{\boldsymbol{g}}_{t,i}||+||\hat{\boldsymbol{g}}_{t,i}||\right)||\boldsymbol{w}_t-\boldsymbol{w}_{t,i}|| \\ &\quad -\frac{\lambda}{2}||\boldsymbol{w}^*-\boldsymbol{w}_{t,i}||^2-\frac{\lambda}{2}||\boldsymbol{w}_{t,i}-\boldsymbol{w}_t||^2,\end{aligned} \tag{17}$$

where $\bar{\boldsymbol{g}}_{t,i} \in \partial f_i(\boldsymbol{w}_t)$, (15) follows from the λ-strong convexity of F_i at $\boldsymbol{w}_{t,i}$, i.e.,

$$F_i(\boldsymbol{w}^*) \geq F_i(\boldsymbol{w}_{t,i})+\hat{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}^*-\boldsymbol{w}_{t,i})+\frac{\lambda}{2}||\boldsymbol{w}^*-\boldsymbol{w}_{t,i}||^2,$$

(16) also follows from the λ-strong convexity of F_i at $\boldsymbol{w}_t$, i.e.,

$$F_i(\boldsymbol{w}_{t,i}) \geq F_i(\boldsymbol{w}_t)+\bar{\boldsymbol{g}}_{t,i}^T(\boldsymbol{w}_{t,i}-\boldsymbol{w}_t)+\frac{\lambda}{2}||\boldsymbol{w}_{t,i}-\boldsymbol{w}_t||^2,$$

and (17) follows from the Cauchy-Schwarz inequality. Summing (17) from $i=1$ to N and taking expectation of both sides, we obtain

$$\begin{aligned}&-\mathbb{E}\left\{\sum_{i=1}^{N}\boldsymbol{g}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*)\right\} \\ &\leq 2G\sum_{i=1}^{N}\sqrt{\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}_{t,i}||^2}+F(\boldsymbol{w}^*)-\mathbb{E}\{F(\boldsymbol{w}_t)\} \\ &\quad -\frac{\lambda N}{2}\sum_{i=1}^{N}\frac{1}{N}\mathbb{E}\left\{||\boldsymbol{w}^*-\boldsymbol{w}_{t,i}||^2+||\boldsymbol{w}_{t,i}-\boldsymbol{w}_t||^2\right\} \\ &\leq F(\boldsymbol{w}^*)-\mathbb{E}\{F(\boldsymbol{w}_t)\}+2G\sum_{i=1}^{N}\sqrt{\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}_{t,i}||^2} \\ &\quad -\frac{\lambda N}{2}\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}^*||^2,\end{aligned} \tag{18}$$

where the first inequality follows from the Cauchy-Schwarz inequality and the boundedness assumption, and the last inequality follows from the Jensen's inequality due to the convexity of the norm operator.

We finally turn our attention to the $\boldsymbol{x}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*)$ term in (11) and write it as follows:

$$\begin{aligned}\boldsymbol{x}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*) &= \boldsymbol{x}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i})+\boldsymbol{x}_{t,i}^T(\boldsymbol{\psi}_{t+1,i}-\boldsymbol{w}^*) \\ &\leq \boldsymbol{x}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}),\end{aligned}$$

since

$$\begin{aligned}\boldsymbol{x}_{t,i}^T(\boldsymbol{\psi}_{t+1,i}-\boldsymbol{w}^*) &\leq -\boldsymbol{x}_{t,i}^T\boldsymbol{x}_{t,i}+(\boldsymbol{\psi}_{t+1,i}-\Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i}))^T \\ &\quad \times(\boldsymbol{w}^*-\Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i})) \\ &\leq 0,\end{aligned}$$

where $(\boldsymbol{\psi}_{t+1,i}-\Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i}))^T(\boldsymbol{w}^*-\Pi_{\mathcal{W}}(\boldsymbol{\psi}_{t+1,i})) \leq 0$ due to the Euclidean projection onto the convex set $\mathcal{W}$ [35]. Taking the expectation of both sides, we can upper bound this term as follows:

$$\begin{aligned}\mathbb{E}\,\boldsymbol{x}_{t,i}^T(\boldsymbol{w}_t-\boldsymbol{w}^*) &\leq \mathbb{E}\left[||\boldsymbol{x}_{t,i}||||\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}||\right] \\ &\leq G\mu_t\sqrt{\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}||^2}\end{aligned} \tag{19}$$

by first using the Cauchy-Schwarz inequality, and then using (13) and the bound $\mathbb{E}||\hat{\boldsymbol{g}}_{t,i}|| \leq \sqrt{\mathbb{E}||\hat{\boldsymbol{g}}_{i,t}||^2} \leq G$.

Putting (14), (18), and (19) back in (11), we obtain

$$\begin{aligned}&\mathbb{E}||\boldsymbol{w}_{t+1}-\boldsymbol{w}^*||^2-(1-\lambda\mu_t)\,\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}^*||^2 \\ &\leq \frac{2\mu_t}{N}\Big[F(\boldsymbol{w}^*)-\mathbb{E}\{F(\boldsymbol{w}_t)\}+G\sum_{i=1}^{N}\big(2\sqrt{\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}_{t,i}||^2} \\ &\quad +\sqrt{\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}||^2}\big)\Big]+4G^2\mu_t^2.\end{aligned} \tag{20}$$

This concludes the proof of Lemma 1. □

Having obtained an upper bound on the performance of the average parameter vector, we then consider the mean square deviation of the parameter vectors at each agent from the average parameter vector. This lemma will then be used to relate the performance of each individual agent to the performance of the fully connected distributed system.

Lemma 2. *In addition to the assumptions in Lemma 1, assume that the initial weights at each agent are identically initialized*

to avoid any bias,[4] *i.e.,* $\boldsymbol{w}_{1,i} = \boldsymbol{w}_{1,j}$, $\forall i, j$. *Then Algorithm 1 yields*

$$\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_t - \boldsymbol{w}_{t,i}\big|\big|^2} \leq 2G\sqrt{N}\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z, \tag{21}$$

and

$$\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_t - \boldsymbol{\psi}_{t+1,i}\big|\big|^2} \leq G\mu_t + 2G\sqrt{N}\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z, \tag{22}$$

where $0 \leq \sigma < 1$ *is the second largest singular value of the matrix* $\boldsymbol{H}$.

Proof. We first let

$$\boldsymbol{W}_t := [\boldsymbol{w}_{t,1}, \ldots, \boldsymbol{w}_{t,N}], \boldsymbol{G}_t := [\hat{\boldsymbol{g}}_{t,1}, \ldots, \hat{\boldsymbol{g}}_{t,N}], \text{ and}$$
$$\boldsymbol{X}_t := [\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,N}].$$

Then, we obtain the recursion on $\boldsymbol{W}_t$ as follows:

$$\boldsymbol{W}_t = \boldsymbol{W}_1\boldsymbol{H}^{t-1} + \sum_{z=1}^{t-1}(\boldsymbol{X}_{t-z} - \mu_{t-z}\boldsymbol{G}_{t-z})\boldsymbol{H}^z. \tag{23}$$

Letting $\boldsymbol{e}_i$ denote the basis function for the ith dimension, i.e., only the ith entry of $\boldsymbol{e}_i$ is 1 whereas the rest are 0, we have

$$\begin{aligned}
\big|\big|\boldsymbol{w}_t - \boldsymbol{w}_{t,i}\big|\big| &= \left|\left|\boldsymbol{W}_t\left(\frac{1}{N}\boldsymbol{1} - \boldsymbol{e}_i\right)\right|\right| \\
&\leq \sum_{z=1}^{t-1}\left|\left|(\boldsymbol{X}_{t-z} - \mu_{t-z}\boldsymbol{G}_{t-z})\left(\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right)\right|\right| \\
&\quad + \big|\big|\boldsymbol{w}_1 - \boldsymbol{w}_{1,i}\big|\big| \qquad (24) \\
&= \sum_{z=1}^{t-1}\left|\left|(\boldsymbol{X}_{t-z} - \mu_{t-z}\boldsymbol{G}_{t-z})\left(\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right)\right|\right| \\
&\leq \sum_{z=1}^{t-1}||\boldsymbol{X}_{t-z} - \mu_{t-z}\boldsymbol{G}_{t-z}||_F\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right|, \\
&\leq \sum_{z=1}^{t-1}(||\boldsymbol{X}_{t-z}||_F + \mu_{t-z}||\boldsymbol{G}_{t-z}||_F)\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right| \qquad (25) \\
&\leq 2\sum_{z=1}^{t-1}\mu_{t-z}||\boldsymbol{G}_{t-z}||_F\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right|
\end{aligned}$$

where (24) follows due to the unbiased initialization assumption, i.e.,

$$\boldsymbol{w}_1 = \boldsymbol{w}_{1,i} = \boldsymbol{w}_{1,j}, \forall i, j \in \{1, \ldots, N\}$$

and (25) follows from (13).

We first consider the term $\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right|$ of (25) and define the matrix $\boldsymbol{B} := \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$. Then, we can write

$$\begin{aligned}
\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right| &= ||\boldsymbol{B}\boldsymbol{e}_i - \boldsymbol{H}^z\boldsymbol{e}_i|| \\
&= ||(\boldsymbol{B} - \boldsymbol{H})^z\boldsymbol{e}_i||,
\end{aligned} \tag{26}$$

where the last line follows since $\boldsymbol{B}^z = \boldsymbol{B}$, $\forall z \geq 1$.

4. This is basically an unbiasedness condition, which is reasonable since the objective weight $\boldsymbol{w}^*$ is completely unknown to us. Even though the initial weights are not identical, our analyses still hold, albeit with small additional excess terms.

Now, let

$$\sigma_1(\boldsymbol{A}) \geq \sigma_2(\boldsymbol{A}) \geq \cdots \geq \sigma_N(\boldsymbol{A})$$

denote the singular values of a matrix $\boldsymbol{A}$. Then, we can upper bound (26) as follows:

$$||(\boldsymbol{B} - \boldsymbol{H})^z\boldsymbol{e}_i|| \leq \sigma_1(\boldsymbol{B} - \boldsymbol{H})\big|\big|(\boldsymbol{B} - \boldsymbol{H})^{z-1}\boldsymbol{e}_i\big|\big|,$$

$\forall z \geq 1$. Therefore, using the above recursion z times to (26), we obtain

$$\begin{aligned}
||(\boldsymbol{B} - \boldsymbol{H})^z\boldsymbol{e}_i|| &\leq \sigma_1^z(\boldsymbol{B} - \boldsymbol{H})||\boldsymbol{e}_i|| \\
&= \sigma_1^z(\boldsymbol{B} - \boldsymbol{H}).
\end{aligned} \tag{27}$$

We note that $\boldsymbol{H}$ and $\boldsymbol{B}$ are doubly stochastic matrices; and $\boldsymbol{B}$ is a rank-1 matrix. Let $\lambda_1(\boldsymbol{A}) \geq \lambda_2(\boldsymbol{A}) \geq \cdots \geq \lambda_N(\boldsymbol{A})$ denote the eigenvalues of a symmetric matrix $\boldsymbol{A}$ and $\Lambda(A) := \{\lambda_1(\boldsymbol{A}), \ldots, \lambda_N(\boldsymbol{A})\}$. Since $\boldsymbol{B}$ is a rank-1 matrix, we have $\lambda_1(\boldsymbol{B}) = 1$ and $\lambda_k(\boldsymbol{B}) = 0$ for $k > 1$ [36]. We want to compute the largest singular value of $\boldsymbol{B} - \boldsymbol{H}$, yet $\boldsymbol{B} - \boldsymbol{H}$ is not a symmetric matrix. Therefore, we check the eigen-spectrum of $(\boldsymbol{B} - \boldsymbol{H})^T(\boldsymbol{B} - \boldsymbol{H}) = \boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{B}$, and the matrices $\boldsymbol{H}^T\boldsymbol{H}$ and $\boldsymbol{B}$ are commuting. This yields

$$\Lambda(\boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{B}) \subset \{\lambda_1 - \lambda_2 : \lambda_1 \in \Lambda(\boldsymbol{H}^T\boldsymbol{H}), \lambda_2 \in \Lambda(\boldsymbol{B})\}.$$

Furthermore, $(\boldsymbol{B} - \boldsymbol{H})^T(\boldsymbol{B} - \boldsymbol{H})\boldsymbol{1} = (\boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{B})\boldsymbol{1} = \boldsymbol{0}$, which implies that the eigen-spectrum of $(\boldsymbol{B} - \boldsymbol{H})^T(\boldsymbol{B} - \boldsymbol{H})$ is equal to the eigen-spectrum of $\boldsymbol{H}^T\boldsymbol{H}$, except the largest eigenvalue of $\boldsymbol{H}^T\boldsymbol{H}$, i.e., $\lambda_1(\boldsymbol{H}^T\boldsymbol{H}) = 1$. Instead of that eigenvalue, the eigen-spectrum of $(\boldsymbol{B} - \boldsymbol{H})^T(\boldsymbol{B} - \boldsymbol{H})$ includes 0. Thus, we have

$$\sigma_1(\boldsymbol{B} - \boldsymbol{H}) = \sigma_2(\boldsymbol{H}), \tag{28}$$

and combining (26), (27), and (28), we obtain

$$\left|\left|\frac{1}{N}\boldsymbol{1} - \boldsymbol{H}^z\boldsymbol{e}_i\right|\right| \leq \sigma_2^z(\boldsymbol{H}). \tag{29}$$

From here on, we denote $\sigma := \sigma_2(\boldsymbol{H})$ for notational simplicity. We also note that $0 \leq \sigma < 1$ since $\boldsymbol{H}$ is an irreducible and aperiodic doubly stochastic matrix [30], [37].

Using (29) in (25), we obtain

$$\big|\big|\boldsymbol{w}_t - \boldsymbol{w}_{t,i}\big|\big| \leq 2\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z||\boldsymbol{G}_{t-z}||_F. \tag{30}$$

Taking the expectation of both sides and noting that

$$\begin{aligned}
\mathbb{E}||\boldsymbol{G}_{t-z}||_F^2 &= \mathbb{E}\left\{\sum_{i=1}^{N}\big|\big|\hat{\boldsymbol{g}}_{t-z,i}\big|\big|^2\right\} \\
&\leq G^2N,
\end{aligned}$$

we can rewrite (30) as follows:

$$\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_t - \boldsymbol{w}_{t,i}\big|\big|^2} \leq 2G\sqrt{N}\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z. \tag{31}$$

An upper bound for the term $\big|\big|\boldsymbol{w}_t - \boldsymbol{\psi}_{t+1,i}\big|\big|$ can be obtained as

$$\begin{aligned}\big|\big|\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}\big|\big| &= \big|\big|\boldsymbol{w}_t-\boldsymbol{w}_{t,i}+\mu_t\hat{\boldsymbol{g}}_{t,i}\big|\big| \\ &\leq \big|\big|\boldsymbol{w}_t-\boldsymbol{w}_{t,i}\big|\big|+\mu_t\big|\big|\hat{\boldsymbol{g}}_{t,i}\big|\big|,\end{aligned}$$

where the last line follows from the triangle inequality. Taking square and then expectation of both sides, we obtain the following upper bound

$$\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_t-\boldsymbol{\psi}_{t+1,i}\big|\big|^2}\leq G\mu_t+2G\sqrt{N}\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z. \tag{32}$$

This concludes the proof of Lemma 2. □

The results in Lemmas 1 and 2 are combined in the following theorem to obtain a regret bound on the performance of the proposed algorithm. This theorem illustrates the convergence rate of our algorithm (i.e., Algorithm 1) over distributed networks. The upper bound on the regret $O\Big(\frac{N\sqrt{N}}{(1-\sigma)T}\Big)$ follows since each agent can only have access to the other subgradient oracles through the exchange of information among the agents. Reference [22] provides a lower bound on the rate of convergence of any algorithm to minimize a Lipschitz and strongly convex function for a single agent system with T oracle calls as $O\big(\frac{1}{T}\big)$. Over a centralized network, the lower bound becomes $O\big(\frac{1}{NT}\big)$ since at each time instant, the centralized processor has access to N oracles instead of 1. Therefore, the upper bound on the rate of convergence, i.e., $O\Big(\frac{N\sqrt{N}}{(1-\sigma)T}\Big)$, matches the lower bounds presented in [22] up to constant terms,[5] hence the shown dependency of the convergence rate of the algorithm on T is optimal.

The computational complexity of the algorithm introduced is on the order of the computational complexity of the SSD iterates up to constant terms. Furthermore, the communication load of the proposed method is the same as the communication load of the SSD algorithm. On the other hand, by using a time-dependent averaging of the SSD iterates, our algorithm achieves a substantially improved performance as shown in Theorem 1 and illustrated through our simulations in Section 4.

Theorem 1. *Under the assumptions in Lemmas 1 and 2, Algorithm 1 with learning rate $\mu_t=\frac{2}{\lambda(t+1)}$ and weighted parameters $\bar{\boldsymbol{w}}_{t,i}$ achieves the following guaranteed convergence rate*

$$\mathbb{E}\big\{F\big(\bar{\boldsymbol{w}}_{T+1,i}\big)\big\}-F(\boldsymbol{w}^*)\leq\frac{4NG^2}{\lambda(T+2)}\left(3+\frac{8\sigma\sqrt{N}}{1-\sigma}\right), \tag{33}$$

for all $T\geq 1$, where $0\leq\sigma<1$ is the second largest singular value of the matrix $\boldsymbol{H}$.

This theorem says that although the agents use only local gradient oracle calls to train their parameter vectors, they asymptotically achieve the performance of the centralized processor because of the information diffusion over the network. This result shows that each agent acquires the information contained in the gradient oracles at every other agent and suffers no regret asymptotically as the number of gradient oracle calls at each agent increases.

5. The number of agents, N, is fixed and independent of the budget to call the oracles, i.e., T.

Proof. According to Lemmas 1 and 2, we have

$$\begin{aligned}&\mathbb{E}\{F(\boldsymbol{w}_t)\}-F(\boldsymbol{w}^*)\\ &\quad\leq\frac{N}{2\mu_t}\mathbb{E}\Big\{(1-\lambda\mu_t)||\boldsymbol{w}_t-\boldsymbol{w}^*||^2-||\boldsymbol{w}_{t+1}-\boldsymbol{w}^*||^2\Big\}\\ &\qquad+3NG^2\mu_t+6N\sqrt{N}G^2\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z.\end{aligned} \tag{34}$$

From the convexity of the cost functions, we also have

$$\mathbb{E}\big\{F_i(\boldsymbol{w}_t)-F_i(\boldsymbol{w}_{t,j})\big\}\geq\mathbb{E}\Big\{\hat{\boldsymbol{g}}_{t,i,j}^T(\boldsymbol{w}_t-\boldsymbol{w}_{t,j})\Big\}, \tag{35}$$

$\forall i,j\in\{1,\ldots,N\}$, where

$$\boldsymbol{g}_{t,i,j}\in\partial F_i(\boldsymbol{w}_{t,j}).$$

Here, we can rewrite (35) as follows:

$$\begin{aligned}\mathbb{E}\big\{F_i(\boldsymbol{w}_{t,j})-F_i(\boldsymbol{w}_t)\big\}&\leq\mathbb{E}\Big\{\hat{\boldsymbol{g}}_{t,i,j}^T(\boldsymbol{w}_{t,j}-\boldsymbol{w}_t)\Big\}\\ &\leq\mathbb{E}\big[\big|\big|\hat{\boldsymbol{g}}_{t,i,j}\big|\big|\big|\big|\boldsymbol{w}_{t,j}-\boldsymbol{w}_t\big|\big|\big]\\ &\leq G\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_{t,j}-\boldsymbol{w}_t\big|\big|^2},\end{aligned} \tag{36}$$

where the second line follows from Cauchy Schwarz inequality and the last line follows from the boundedness assumption. Summing (36) from $i=1$ to N, we obtain

$$\mathbb{E}\big\{F(\boldsymbol{w}_{t,j})-F(\boldsymbol{w}_t)\big\}\leq NG\sqrt{\mathbb{E}\big|\big|\boldsymbol{w}_{t,j}-\boldsymbol{w}_t\big|\big|^2}. \tag{37}$$

Using Lemma 2 in (37), we get

$$\mathbb{E}\big\{F(\boldsymbol{w}_{t,j})-F(\boldsymbol{w}_t)\big\}\leq 2N\sqrt{N}G^2\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z. \tag{38}$$

We then add (34) and (38) to obtain

$$\begin{aligned}&\mathbb{E}\big\{F(\boldsymbol{w}_{t,j})\big\}-F(\boldsymbol{w}^*)\\ &\quad\leq\frac{N}{2\mu_t}\mathbb{E}\Big\{(1-\lambda\mu_t)||\boldsymbol{w}_t-\boldsymbol{w}^*||^2-||\boldsymbol{w}_{t+1}-\boldsymbol{w}^*||^2\Big\}\\ &\qquad+3NG^2\mu_t+8N\sqrt{N}G^2\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z.\end{aligned} \tag{39}$$

Multiplying both sides of (39) by t and summing from $t=1$ to T yields [34]

$$\begin{aligned}&\sum_{t=1}^{T}t\left[\mathbb{E}\big\{F(\boldsymbol{w}_{t,j})\big\}-F(\boldsymbol{w}^*)\right]\\ &\leq\frac{N(1-\lambda\mu_1)}{2\mu_1}\mathbb{E}||\boldsymbol{w}_1-\boldsymbol{w}^*||^2-\frac{TN}{2\mu_T}\mathbb{E}||\boldsymbol{w}_{T+1}-\boldsymbol{w}^*||^2\\ &\quad+\sum_{t=2}^{T}\frac{N}{2}\left(\frac{t(1-\lambda\mu_t)}{\mu_t}-\frac{t-1}{\mu_{t-1}}\right)\mathbb{E}||\boldsymbol{w}_t-\boldsymbol{w}^*||^2\\ &\quad+3NG^2\sum_{t=1}^{T}t\mu_t+8N\sqrt{N}G^2\sum_{t=1}^{T}t\sum_{z=1}^{t-1}\mu_{t-z}\sigma^z.\end{aligned} \tag{40}$$

Next, we observe that

$$\sum_{t=1}^{T}\sum_{z=1}^{t-1} t\mu_{t-z}\sigma^z = \sum_{t=1}^{T}\sum_{z=1}^{T} t\mu_{t-z}\sigma^z I_{\{t>z\}} \tag{41}$$

$$= \sum_{z=1}^{T}\sigma^z \sum_{t=z+1}^{T} t\mu_{t-z}$$

$$\leq \sum_{z=1}^{T}\sigma^z \sum_{t=1}^{T} t\mu_t \tag{42}$$

$$\leq \frac{\sigma}{1+\sigma}\sum_{t=1}^{T} t\mu_t,$$

where $I_{\{t>z\}}$ is the indicator function and (42) follows since $0 \leq \sigma < 1$. Using (42) in (40) and inserting $\mu_t = \frac{2}{\lambda(t+1)}$, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T} t\big[\mathbb{E}\{F(\boldsymbol{w}_{t,j})\} - F(\boldsymbol{w}^*)\big] \\
&\quad\leq -\frac{\lambda NT(T+1)}{4}\,\mathbb{E}||\boldsymbol{w}_{T+1} - \boldsymbol{w}^*||^2 \\
&\qquad + \Big(3NG^2 + 8N\sqrt{N}G^2\frac{\sigma}{1-\sigma}\Big)\sum_{t=1}^{T}\frac{2t}{\lambda(t+1)} \qquad (43)\\
&\quad\leq -\frac{\lambda NT(T+1)}{4}\,\mathbb{E}||\boldsymbol{w}_{T+1} - \boldsymbol{w}^*||^2 \\
&\qquad + \frac{2NG^2T}{\lambda}\Big(3 + 8\sqrt{N}\frac{\sigma}{1-\sigma}\Big),
\end{aligned}$$

where the last line follows since $\frac{t}{t+1} \leq 1$. Dividing both sides of (43) by $\sum_{t=1}^{T} t = \frac{T(T+1)}{2}$, we obtain

$$\begin{aligned}
&\mathbb{E}\left\{\frac{2}{T(T+1)}\sum_{t=1}^{T} t\big[F(\boldsymbol{w}_{t,j}) - F(\boldsymbol{w}^*)\big]\right\} \\
&\qquad\leq -\frac{\lambda N}{2}\,\mathbb{E}||\boldsymbol{w}_{T+1} - \boldsymbol{w}^*||^2 \qquad (44)\\
&\qquad\quad + \frac{4NG^2}{\lambda(T+1)}\Big(3 + 8\sqrt{N}\frac{\sigma}{1-\sigma}\Big).
\end{aligned}$$

Since F_i's are convex for all $i \in \{1,\ldots,N\}$, F is also convex. Thus, from Jensen's inequality, we can write

$$\begin{aligned}
&\mathbb{E}\left\{F\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t\,\boldsymbol{w}_{t,j}\right)\right\} - F(\boldsymbol{w}^*) \\
&\qquad\leq \mathbb{E}\left\{\frac{2}{T(T+1)}\sum_{t=1}^{T} t\big(F(\boldsymbol{w}_{t,j}) - F(\boldsymbol{w}^*)\big)\right\}.
\end{aligned} \tag{45}$$

Combining (44) and (45), we obtain

$$\begin{aligned}
&\mathbb{E}\left\{F\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t\,\boldsymbol{w}_{t,j}\right)\right\} - F(\boldsymbol{w}^*) \\
&\qquad\leq -\frac{\lambda N}{2}\,\mathbb{E}||\boldsymbol{w}_{T+1} - \boldsymbol{w}^*||^2 \qquad (46)\\
&\qquad\quad + \frac{4NG^2}{\lambda(T+1)}\Big(3 + 8\sqrt{N}\frac{\sigma}{1-\sigma}\Big).
\end{aligned}$$

Note that the weighting step in Algorithm 1, i.e., (7), leads to

$$\begin{aligned}
\boldsymbol{w}_{T,j} &= \frac{\cancel{T-1}}{T+1}\frac{\cancel{T-2}}{T}\frac{\cancel{T-3}}{\cancel{T-1}}\cdots\frac{\cancel{2}}{\cancel{2+2}}\frac{1}{\cancel{1+2}}\frac{2}{\cancel{0+2}}\boldsymbol{w}_{1,j} \\
&\quad + \frac{\cancel{T-1}}{T+1}\frac{\cancel{T-2}}{T}\frac{\cancel{T-3}}{\cancel{T-1}}\cdots\frac{\cancel{3}}{\cancel{3+2}}\frac{2}{\cancel{2+2}}\frac{2}{\cancel{1+2}}\boldsymbol{w}_{2,j} \\
&\quad + \cdots + \frac{2}{T+1}\boldsymbol{w}_{T,j} \\
&= \frac{2}{T(T+1)}\sum_{t=1}^{T} t\,\boldsymbol{w}_{t,j}.
\end{aligned} \tag{47}$$

This concludes the proof of Theorem 1. □

Hence, using the weighted average $\bar{\boldsymbol{w}}_{t,i}$ instead of the original SSD iterates $\boldsymbol{w}_{t,i}$, we can achieve a convergence rate of $O\Big(\frac{N\sqrt{N}}{(1-\sigma)T}\Big)$. The denominator T of this regret bound follows since we use a time-varian't weighting of the SSD iterates. The linear dependency to the network size follows since we add N different cost functions, i.e., one corresponding to each agent. Finally, the sub-linear dependency to the network size results from the diffusion of the parameter vector over the distributed network.

We note that the upper bound in (33) includes σ, which can depend on the network size N. In particular, for different communication matrices, the corresponding upper bounds on the second largest singular value can be included in (33). As an example, Reference [38] shows that the second largest singular value of the lazy metropolis matrix, defined by

$$\boldsymbol{H}_{ij} = \begin{cases} \frac{1}{\max\{n_i,n_j\}}, & \text{if } j \in \mathcal{N}_i \setminus i \\ 0, & \text{if } j \notin \mathcal{N}_i \\ \frac{1}{2} - \frac{1}{2}\sum_{j\in\mathcal{N}_i\setminus i}\boldsymbol{H}_{ij}, & \text{if } i = j, \end{cases} \tag{48}$$

is bounded from above by $1 - \frac{1}{71N^2}$, which implies $\frac{\sigma}{1-\sigma} < O(N^2)$.

In the following corollary, we provide an MSD guarantee on the weighted parameters $\bar{\boldsymbol{w}}_{t,i}$.

Corollary 1. *Under the assumptions in Lemmas 1 and 2, Algorithm 1 with learning rate $\mu_t = \frac{2}{\lambda(t+1)}$ and weighted parameters $\bar{\boldsymbol{w}}_{t,i}$ guarantees the following MSD*

$$\mathbb{E}\big|\big|\bar{\boldsymbol{w}}_{T+1,i} - \boldsymbol{w}^*\big|\big|^2 \leq \frac{8NG^2}{\lambda^2(T+2)}\left(3 + \frac{8\sigma\sqrt{N}}{1-\sigma}\right),$$

for all $T \geq 1$, where $0 \leq \sigma < 1$ is the second largest singular value of the matrix $\boldsymbol{H}$.

Proof. This follows from Theorem 1 (33) and λ-strong convexity (2) of F at $\boldsymbol{w}^*$ since $\mathbf{0} \in \partial F(\boldsymbol{w}^*)$. □

In the following corollary, we then consider the performance of the average SSD iterate instead of the time-variant weighted iterate in (47). Note that even though the agents have consumed their budgets to call the subgradient oracle, they can continue to exchange information, which averages out the iterates. We show that the average SSD iterate achieves an MSD of $O\Big(\frac{\sqrt{N}}{(1-\sigma)T}\Big)$. This MSD follows due to the number of gradient oracle calls and diffusion regret over the distributed network.

Corollary 2. *Under the assumptions of Lemmas 1 and 2, Algorithm 1 with learning rate $\mu_t = \frac{2}{\lambda(t+1)}$ yields the following guaranteed MSD*

$$\mathbb{E}||\boldsymbol{w}_{T+1} - \boldsymbol{w}^*||^2 \leq \frac{8G^2}{\lambda^2(T+1)}\left(3 + \frac{8\sigma\sqrt{N}}{1-\sigma}\right). \quad (49)$$

for all $T \geq 1$, where $\boldsymbol{w}_t = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{w}_{t,i}$ and $0 \leq \sigma < 1$ is the second largest singular value of the matrix $\boldsymbol{H}$.

Proof. This follows from (46) and (47) since $\mathbb{E}\{F(\bar{\boldsymbol{w}}_{T,j})\} - F(\boldsymbol{w}^*) \geq 0$. □

Remark 1. Algorithm 1 can be generalized to apply to consensus in a straightforward manner, while the performance guarantee in Theorem 1 still holds up to constant terms, i.e., we still have a convergence rate of $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$. For the consensus strategy, the lines 5–8 of Algorithm 1 would be replaced by the following update

$$\boldsymbol{w}_{t+1,i} = \Pi_{\mathcal{W}}\left(\sum_{j=1}^{N}\boldsymbol{H}_{ji}\boldsymbol{w}_{t,j} - \mu_t\hat{\boldsymbol{g}}_{t,i}\right). \quad (50)$$

Hence, we have the following recursion on the parameter vectors

$$\boldsymbol{W}_t = \boldsymbol{W}_1\boldsymbol{H}^{t-1} - \sum_{z=1}^{t-1}(\boldsymbol{X}_{t-z} - \mu_{t-z}\boldsymbol{G}_{t-z})\boldsymbol{H}^{z-1},$$

instead of the one in (23). Under this modification, Lemma 2 can be updated as follows:

$$\sqrt{\mathbb{E}\left|\left|\boldsymbol{w}_t - \boldsymbol{w}_{t,i}\right|\right|^2} \leq 2G\sqrt{N}\sum_{z=1}^{t-1}\mu_{t-z}\sigma^{z-1}. \quad (51)$$

This loosens the upper bounds in (25) and (32) by a factor of $1/\sigma$ (note that $0 \leq \sigma < 1$). Therefore, diffusion strategies achieves a better convergence performance compared to the consensus strategy.

We note that the proposed algorithm leads to the theoretical bounds on the convergence rate for a certain step size, which is $\mu_t = \frac{2}{\lambda(t+1)}$. Otherwise, the algorithm can also be used with different step sizes yet not necessarily delivering such theoretical performance guarantees. Furthermore, even though all the agents use $\mu_t = \frac{2}{\lambda(t+1)}$, the only necessary information for them to keep how many times they have called the subgradient oracles and then they can all use the same step size $2/\lambda$ and scale it by $1/(t+1)$. We consider that the cost function F is λ-strongly convex, i.e., the agents have the knowledge of λ even though they do not know what the function is.

4 Simulations

In this section, we first examine the performance of the proposed algorithms for various distributed network topologies, namely the star, the circle, and a random network topologies (which are shown in Fig. 3). In all cases, we have a network of $N = 20$ agents where each agent i at time t, observes the data $d_{t,i} = \boldsymbol{w}_0^T\boldsymbol{u}_{t,i} + v_{t,i}$, $i = 1, \ldots, N$, where the regression vector $\boldsymbol{u}_{t,i}$ and the observation noise $v_{t,i}$ are generated from i.i.d. zero mean Gaussian processes for all $t \geq 1$. The variance of the observation noise is $\sigma_{v,i}^2 = 0.1$ for all $i = 1, \ldots, N$, whereas the auto-covariance matrix of the regression vector $\boldsymbol{u}_{t,i} \in \mathbb{R}^5$ is randomly chosen for each agent $i = 1, \ldots, N$ such that the signal-to-noise ratio (SNR) over the network varies between -15 dB to 10 dB (see Fig. 2). The parameter of interest, $\boldsymbol{w}_0 \in \mathbb{R}^5$, is randomly chosen from a zero mean Gaussian process and normalized to have a unit norm, i.e., $||\boldsymbol{w}_0|| = 1$. We use the well-known Metropolis combination rule [3] to set the combination weights as follows:

$$\boldsymbol{H}_{ij} = \begin{cases} \frac{1}{\max\{n_i, n_j\}}, & \text{if } j \in \mathcal{N}_i \setminus i \\ 0, & \text{if } j \notin \mathcal{N}_i \\ 1 - \sum_{j \in \mathcal{N}_i \setminus i}\boldsymbol{H}_{ij}, & \text{if } i = j \end{cases} \quad (52)$$

where n_i is the number of neighboring agents for agent i.

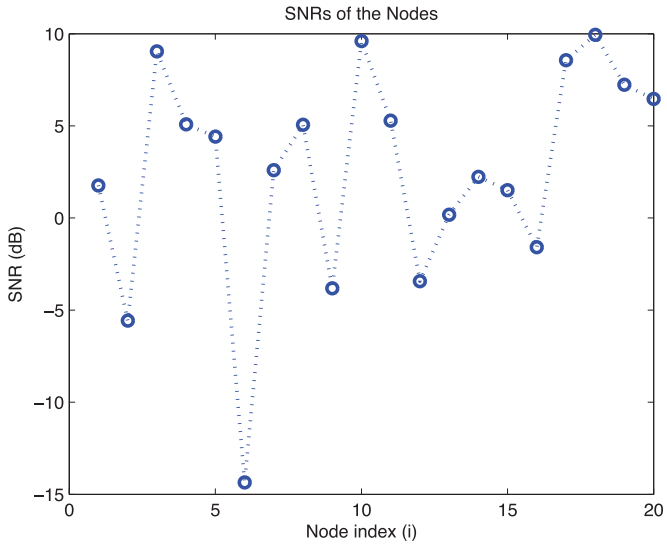
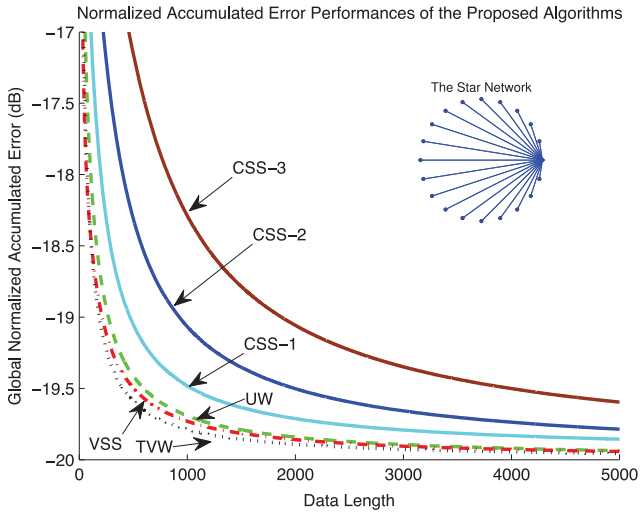


Fig. 2. SNRs of the agents in the distributed network.

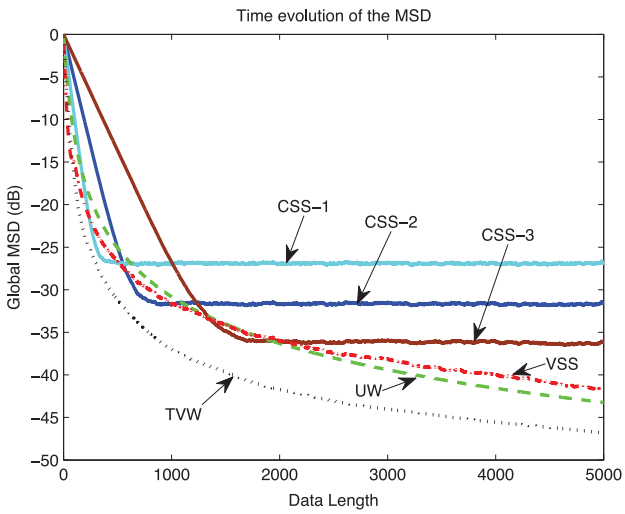
For this set of experiments, we consider the squared error loss, i.e., $\ell(\boldsymbol{w}_{t,i}; \boldsymbol{u}_{t,i}, d_{t,i}) = (d_{t,i} - \boldsymbol{w}_{t,i}^T\boldsymbol{u}_{t,i})^2$ as our loss function. In the figures, CSS represents the distributed constant step-size SSD algorithm of [11], VSS represents the distributed variable step-size SSD algorithm of [6], UW represents the distributed version of the uniform weighted SSD algorithm of [23], and TVW represents the distributed time variant weighted SSD algorithm introduced in this paper. The step-sizes of the CSS-1, CSS-2, and CSS-3 algorithms are set to 0.05, 0.1, and 0.2, respectively, at each agent and the learning rates of the VSS and UW algorithms are set to $1/(\lambda t)$ as noted in [6], [23], whereas the learning rate of the TVW algorithm is set to $2/(\lambda(t+1))$ as noted in Theorem 1, where $\lambda = 0.01$. These learning rates are chosen specifically to guarantee a fair performance comparison between these algorithms according to the corresponding algorithm descriptions stated in this paper and in [6], [23].

In the left column of Fig. 3, we compare the normalized time accumulated error performances of these algorithms under different network topologies in terms of the global normalized cumulative error (NCE) measure, i.e.,

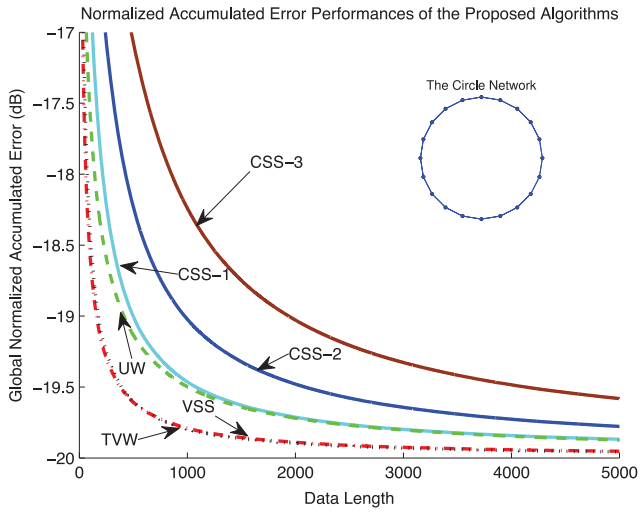
$$\text{NCE}(t) = \frac{1}{Nt}\sum_{i=1}^{N}\sum_{\tau=1}^{t}(d_{\tau,i} - \boldsymbol{w}_{\tau,i}^T\boldsymbol{u}_{\tau,i})^2.$$



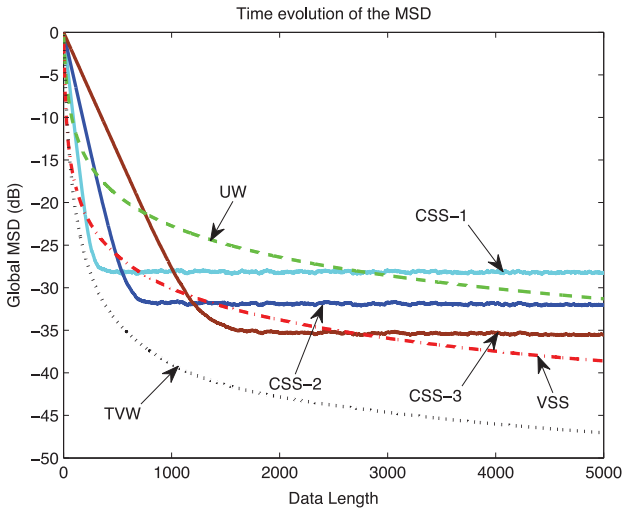
(a) Global NCE for the star network



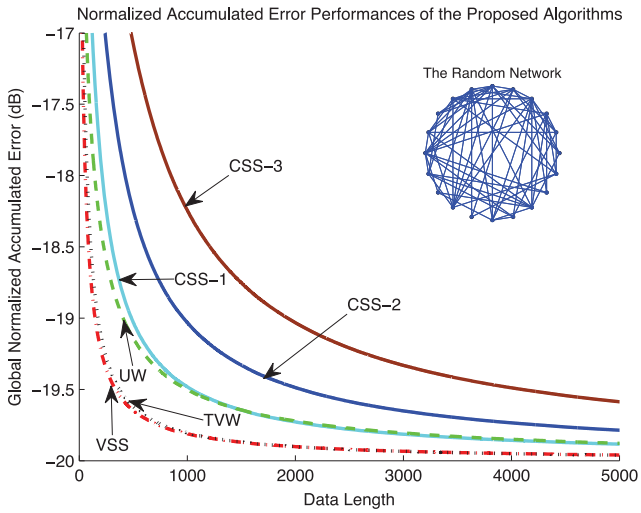
(b) Global MSD for the star network



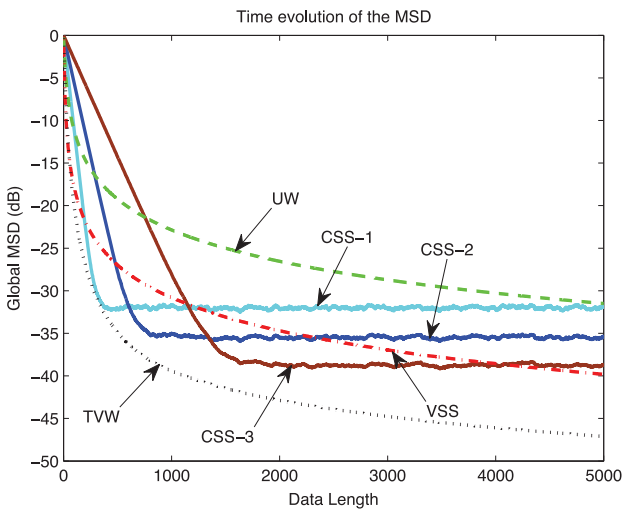
(c) Global NCE for the circle network



(d) Global MSD for the circle network



(e) Global NCE for the random network



(f) Global MSD for the random network

Fig. 3. NCE (left column) and MSD (right column) performances of the proposed algorithms under the star (first row), the circle (second row), and a random (third row) network topologies, under the squared error loss function averaged over 200 trials.

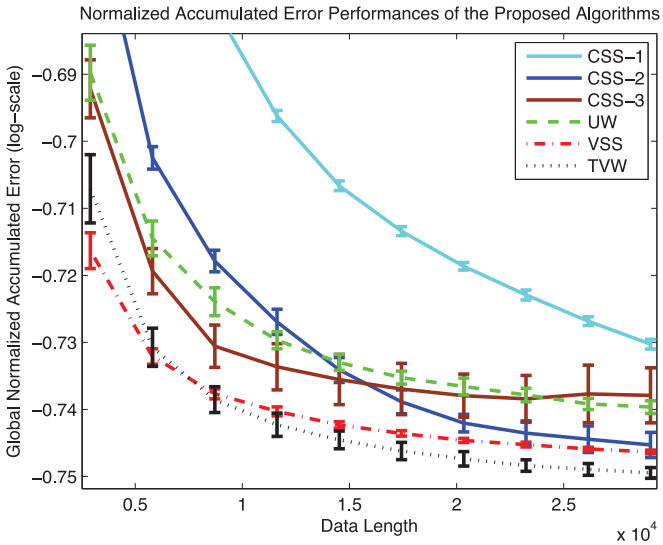


Fig. 4. Normalized accumulated errors of the six algorithms versus training data length for cover type data averaged over 250 trials for a network of size 20.

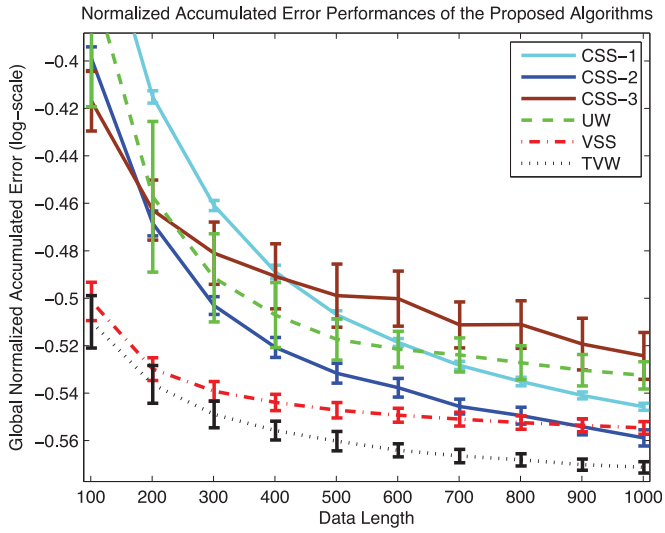


Fig. 5. Normalized accumulated errors of the six algorithms versus training data length for quantum data averaged over 100 trials for a network of size 50.

Additionally, in the right column of Fig. 3, we compare the performances of the algorithms in terms of the global MSD measure, i.e.,

$$\mathrm{MSD}(t) = \frac{1}{N}\sum_{i=1}^{N} \left|\left| \boldsymbol{w}_0 - \boldsymbol{w}_{t,i} \right|\right|^2. \tag{53}$$

In the figures, we have plotted the NCE and MSE performances of the proposed algorithms over 200 independent trials to avoid any bias.

As can be seen in the Fig. 3, the TVW algorithm substantially outperforms its competitors and achieves a much smaller error performance. This superior performance of our algorithm is achieved thanks to the time-dependent weighting of the regression parameters, used to obtain a faster convergence rate with respect to the rest of the algorithms. Hence, by using a certain time varying weighting of the SSD iterates, we obtain a significantly improved convergence performance compared to the state-of-the-art approaches in the literature. Furthermore, the performance of our algorithm is robust against the network topology, whereas the competitor algorithms may not provide satisfactory performances under different network topologies.

We next consider the classification tasks over the benchmark data sets: Covertype[6] and quantum.[7] For this set of experiments, we consider the hinge loss, i.e., $\ell(\boldsymbol{w}_{t,i}; \boldsymbol{u}_{t,i}, d_{t,i}) = \max\{0, 1 - d_{t,i}\boldsymbol{w}_{t,i}^T \boldsymbol{u}_{t,i}\}^2$ as our loss function. The regularization constant is set to $\lambda = 1/T$, where the step sizes of the TVW, UW, and VSS algorithms are set as in the previous experiment. The step sizes of the CSS-1, CSS-2, and CSS-3 algorithms are set to 0.02, 0.05, and 0.1 for the covertype data set, whereas the step sizes of the CSS-1, CSS-2, and CSS-3 algorithms are set to 0.01, 0.02, and 0.05 for the quantum data set. These learning rates are chosen to illustrate the tradeoff between the convergence speed and the steady state performance of the constant step size SSD methods. The network sizes are set to $N = 20$ and $N = 50$ for the covertype and quantum data sets, respectively.

In Figs. 4 and 5, we illustrate the performances of the six algorithms for various training data lengths. In particular, we train the parameter vectors at each agent using a certain length of training data and test the performance of the final parameter vector over the entire data set. We provide averaged results over 250 and 100 independent trials for covertype and quantum data sets, respectively, and present the mean and variance of the normalized accumulated hinge errors. These figures illustrate that the proposed TVW algorithm significantly outperforms its competitors. Although the performances of the UW and VSS algorithms are comparably robust over different iterations, the TVW algorithm provides a smaller accumulated loss. On the other hand, the variances of the constant step size methods highly deteriorate as the step size increases. Although decreasing the step size yields more robust performance for these constant step size algorithms, the TVW algorithm

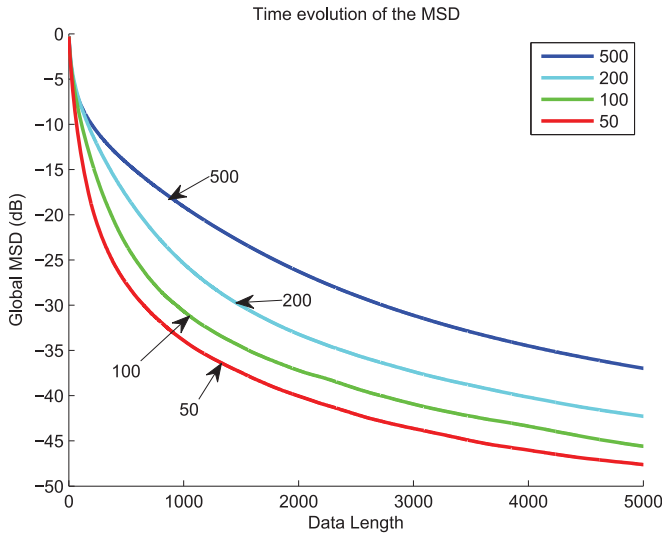


Fig. 6. Global MSD over the star networks with different sizes: 50, 100, 200, and 500.

6. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
7. http://osmot.cs.cornell.edu/kddcup/

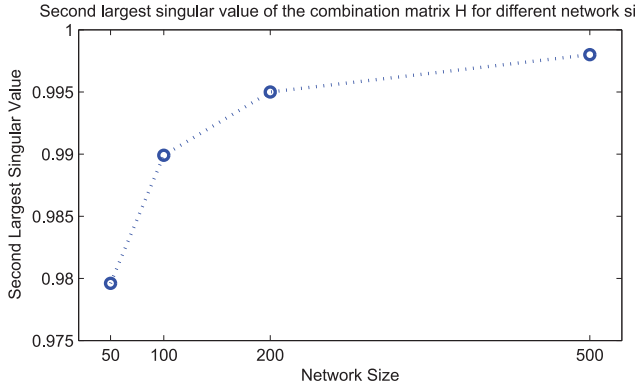


Fig. 7. Second largest singular value, σ, of the combination matrix H over the star networks with different sizes: 50, 100, 200, and 500.

provides a significantly smaller steady-state cumulative error with respect to these methods.

Finally, we note that the upper bounds in Theorem 1 and Corollary 1 directly depend on the number of agents in addition to the second largest singular value of the combination matrix, σ. In the following numerical examples, we examine how the MSD of the algorithm scales with the network size. To this end, we consider the setup for Fig. 3b with different network sizes: 50,100,200, and 500. Fig. 6 shows how the time evolution of the global MSD scales with increasing network sizes from 50 to 500. Furthermore, Fig. 7 shows how σ scales with the network size and correspondingly we observe that $1/(1-\sigma)$ scales with N. We also note that the global MSD measure (53) is averaged across the network. Therefore, the corresponding upper bound on the global MSD (See Corollary 1) scales with $\sqrt{N}/(1-\sigma)$. However, in Fig. 6, we observe that when the network size scales by 10, e.g., from 50 to 500, the global MSD scales by 10 dB rather than 15dB. This raises the possibility that the dependency of the upper bound on the network size might be tightened further and formulating the upper bound, which is also optimal in terms of network size complexity, can be an interesting future research direction.

5 Conclusion

We have studied distributed strongly convex optimization over distributed networks, where the aim is to minimize a sum of unknown convex objective functions. We have introduced an algorithm that uses a limited number of gradient oracle calls to these objective functions and achieves an optimal convergence rate of $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient updates at each agent. This level of performance is achieved by using a certain time-dependent weighting of the SSD iterates at each agent. Additionally, the weighted parameters achieve a guaranteed mean square deviation (MSD) of $O\left(\frac{N\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient updates. The computational complexity and the communication load of the proposed approach is the same as with the state-of-the-art methods in the literature up to constant terms. We have also proved that the average SSD iterate, which can be attained if the agents continue to exchange information without gradient updates, achieves a guaranteed MSD of $O\left(\frac{\sqrt{N}}{(1-\sigma)T}\right)$ after T gradient oracle calls. We have illustrated the superior convergence rate of our algorithm with respect to the state-of-the-art methods in the literature. Some future directions of research on this topic include the computation of convergence rate bounds for the heterogeneous case, where agents can have different step sizes and/or learning rates, and asynchronous distributed computation as in [39].

Acknowledgments

This work is supported in part by TUBITAK Contract No 115E917, in part by the U.S. Office of Naval Research (ONR) MURI grant N00014-16-1-2710, and in part by NSF under grant CCF 11-11342.

References

[1] A. Yazicioglu, M. Egerstedt, and J. Shamma, "Formation of robust multi-agent networks through self-organizing random regular graphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 2, no. 4, pp. 139–151, Oct.-Dec. 2015.

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

[3] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[4] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[5] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, Jul. 2006.

[6] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, Nov. 2013.

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

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

[9] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1291–1306, Jun. 2011.

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

[11] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[12] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

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

[14] S.-Y. Tu and A. H. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, Mar. 2014.

[15] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[16] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

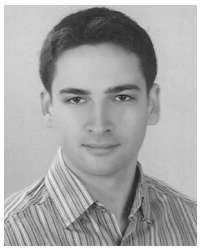
[17] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[18] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[19] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[20] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.
[21] Z. J. Towfic and A. H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2888–2903, Jun. 2015.
[22] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, "Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3235–3249, May 2012.
[23] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 449–456.
[24] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, Sep. 2014.
[25] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.
[26] G. Mateos, I. Schizas, and G. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.
[27] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3740–3754, Jul. 2012.
[28] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.
[29] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 707–724, Aug. 2011.
[30] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.
[31] K. Tsianos and M. Rabbat, "Distributed strongly convex optimization," in *Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2012, pp. 593–600.
[32] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2–3, pp. 169–192, Dec. 2007.
[33] E. Hazan and S. Kale, "Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 2489–2512, Jul. 2014.
[34] S. Lacoste-Julien, M. W. Schmidt, and F. Bach, "A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method," Dec. 2012. [Online]. Available: http://arxiv.org/pdf/1212.2002v2.pdf
[35] D. G. Luenberger, *Optimization by Vector Space Methods*. Hoboken, NJ, USA: Wiley, 1969.
[36] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, United Kingdom: Cambridge Univ. Press, 1985.
[37] D. Levin, Y. Peres, and E. Wilmer, *Markov Chains and Mixing Times*. Providence, Rhode Island, USA: American Mathematical Society, 2008.
[38] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2016.
[39] S. Li and T. Başar, "Asymptotic agreement and convergence of asynchronous stochastic algorithms," *IEEE Trans. Autom. Control*, vol. AC-32, no. 7, pp. 612–618, Jul. 1987.

Muhammed O. Sayin received the BS and MS degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013 and 2015, respectively. He is currently pursuing the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC). His current research interests include signaling games, dynamic games and decision theory, strategic decision making, and stochastic optimization.



N. Denizcan Vanli received the BS and MS degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013 and 2015, respectively. He is currently pursuing the PhD degree in electrical engineering and computer science at Massachusetts Institute of Technology, Cambridge, MA. His research interests include convex optimization, online learning, and distributed optimization.

Suleyman S. Kozat received the BS degree with full scholarship and high honors from Bilkent University, Turkey and the MS and PhD degrees in electrical and computer engineering from the University of Illinois at Urbana Champaign, Urbana, IL. After graduation, he joined IBM Research, T. J. Watson Research Lab, Yorktown, New York, as a research staff member (and later became a project leader) in the Pervasive Speech Technologies Group, where he focused on problems related to statistical signal processing and machine learning. While doing the PhD degree, he was also working as a research associate at Microsoft Research, Redmond, Washington, in the Cryptography and Anti-Piracy Group. He holds several patent inventions due to his research accomplishments at IBM Research and Microsoft Research. He is currently an associate professor at the Electrical and Electronics Engineering Department at Bilkent University. He is the elected president of the IEEE Signal Processing Society, Turkey Chapter. He coauthored more than 100 papers in refereed high impact journals and conference proceedings and has several patent inventions (currently used in several different Microsoft and IBM products such as the MSN and the ViaVoice). He holds many international and national awards. Overall, his research interests include cyber security, anomaly detection, big data, data intelligence, adaptive filtering and machine learning algorithms for signal processing. He is a senior member of the IEEE.

Tamer Başar received the BSEE degree from Robert College, Istanbul, the MS, MPhil, and PhD degrees from Yale University. He is with the University of Illinois at Urbana-Champaign, where he holds the academic positions of Swanlund endowed chair; Center for Advanced Study professor of Electrical and Computer Engineering; research professor at the Coordinated Science Laboratory; and research professor at the Information Trust Institute. He is also the director of the Center for Advanced Study. He is a member of the US National Academy of Engineering, member of the European Academy of Sciences, IFAC (International Federation of Automatic Control) and SIAM (Society for Industrial and Applied Mathematics), and has served as president of IEEE CSS (Control Systems Society), ISDG (International Society of Dynamic Games), and AACC (American Automatic Control Council). He has received several awards and recognitions over the years, including the highest awards of IEEE CSS, IFAC, AACC, and ISDG, the IEEE Control Systems Award, and a number of international honorary doctorates and professorships. He has more than 750 publications in systems, control, communications, and dynamic games, including books on non-cooperative dynamic game theory, robust control, network security, wireless and communication networks, and stochastic networked control. He was the Editor-in-Chief of Automatica between 2004 and 2014, and is currently editor of several book series. His current research interests include stochastic teams, games, and networks; distributed algorithms; security; and cyber-physical systems. He is a fellow of the IEEE.