Log-Scale Quantization in Distributed First-Order Methods: Gradient-based Learning from Distributed Data

Mohammadreza Doostmohammadian, Muhammad I. Qureshi, Mohammad Hossein Khalesi,
Hamid R. Rabiee, *Senior Member, IEEE*, Usman A. Khan, *Senior Member, IEEE*

Abstract—Decentralized strategies are of interest for learning from large-scale data over networks. This paper studies learning over a network of geographically distributed nodes/agents subject to quantization. Each node possesses a private local cost function, collectively contributing to a global cost function, which the considered methodology aims to minimize. In contrast to many existing papers, the information exchange among nodes is log-quantized to address limited network-bandwidth in practical situations. We consider a first-order computationally efficient distributed optimization algorithm (with no extra inner consensus loop) that leverages node-level gradient correction based on local data and network-level gradient aggregation only over nearby nodes. This method only requires balanced networks with no need for stochastic weight design. It can handle log-scale quantized data exchange over possibly time-varying and switching network setups. We study convergence over both structured networks (for example, training over data-centers) and ad-hoc multi-agent networks (for example, training over dynamic robotic networks). Through experimental validation, we show that (i) structured networks generally result in a smaller optimality gap, and (ii) log-scale quantization leads to a smaller optimality gap compared to uniform quantization.

Note to Practitioners—Motivated by recent developments in cloud computing, parallel processing, and the availability of low-cost CPUs and communication networks, this paper considers distributed and decentralized algorithms for machine learning and optimization. These algorithms are particularly relevant for decentralized data mining, where data sets are distributed across a network of computing nodes. A practical example of this is the classification of images over a networked data centre. In real-world scenarios, practical model nonlinearities such as data quantization must be addressed for information exchange among the computing nodes. This work emphasizes the importance of handling log-scale quantization and compares its performance over uniform quantization. By exploring these quantization methods, we aim to determine which is more accurate in terms of optimality gap and learning residual. Moreover, we study the impact of the structure of the information-sharing network on reducing the optimality gap and improving the convergence rate of distributed algorithms. As contemporary distributed and networked data mining systems demand highly accurate algorithms with fast convergence for real-time applications, our research emphasizes the benefit of structured networks under logarithmic quantization information-exchange. Our findings can be extended to different machine learning algorithms, offering pathways to more accurate and faster data mining solutions.

M. Doostmohammadian and M. H. Khalesi are with Faculty of Mechanical Engineering, Semnan University, Semnan, Iran, {doost,mhkhalesi}@semnan.ac.ir. M. I. Qureshi is with the Electrical and Computer Engineering Department, Tufts University, Medford, USA muhammad.qureshi@tufts.edu. H. R. Rabiee is with the Computer Engineering Department, Sharif University of Technology, Tehran, Iran rabiee@sharif.edu. U. A. Khan holds concurrent appointments as a Professor of ECE/CS at Tufts University and as an Amazon Scholar. This paper describes work performed at Tufts University and is not associated with Amazon, khan@ece.tufts.edu.

I. INTRODUCTION

LEARNING from geographically distributed data has emerged as a critical challenge in various domains such as sensor networks, edge computing, and decentralized systems [1]–[3]. The scenario involves individual nodes possessing private local cost functions, collectively contributing to a global cost that needs to be minimized [4]–[7]. This paper focuses on addressing this challenge through gradient-based optimization over networks, targeting scenarios where the cost functions are locally known at distributed nodes communicating over quantized channels. Data quantization, the process of representing data with a reduced number of bits, is often employed to reduce the communication load in such decentralized setups and is proven as an effective way to reduce communication costs, for example, in machine learning applications [8], [9]. One common quantisation type is logarithmic or log-scale that efficiently assigns more bits to represent smaller values and fewer bits to larger values [10], [11]. This is beneficial when dealing with gradients or weights that often have a wide dynamic range. Smaller values, which might be more critical for convergence, are represented with higher precision. This may increase the complexity of the distributed learning algorithm as we move from linear dynamics to non-linear dynamics due to quantization. Uniform quantization, on the other hand, simplifies the representation of values, which can lead to a faster communication process. However, the lack of precision, especially for small values, might affect performance in scenarios where accurate representation of gradients or weights is crucial, and lead to residual error [12], [13]. Therefore, studying and comparing uniform and log-scale quantization for distributed optimization is of interest for distributed data mining applications.

The existing literature on distributed optimization mostly consider linear function of the state variables, where the focus is mainly on reducing the optimality gap and improving the convergence rate. The primary distributed optimization methods are **GP** [14] and **DGD** algorithm [15], which are further extended to **DSGD** [16] and **SGP** [17], [18] by adopting stochastic gradient methods. Algorithm **ADDOPT** [19] and its stochastic version **S-ADDOPT** [20] further accelerate the convergence rate using gradient tracking. Similarly, momentum-based approaches improve the convergence rate using heavy-ball method [21], Nesterov gradient method [22], [23], or a combination of both [24]. On the other hand, **PushSVRG**

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3526967

2

[25] and **PushSAGA** [26], [27] algorithms use variance reduction to eliminate the uncertainty caused by the stochastic gradients. Few other distributed optimization works use sign-based nonlinearity in the node dynamics to address finite-time and fixed-time convergence [28]–[30]. All these methods assume unlimited bandwidth to communicate arbitrary real numbers which in practice is not possible as all realistic communication protocols rely on finite precision. The effect of uniform quantization over channels with limited data-rate and the associated optimality gap (or residual) is discussed in [31]–[33]. The work [34] proposes a novel quantized distributed gradient tracking algorithm to minimize the finite sum of local costs over digraphs with linear convergence rate. The paper [35] combines the gradient tracking Push-Pull method with communication compression (quantized links). Similarly, communication compression technique compatible with a general class of operators unifying both unbiased and biased compressors is discussed in [36]. Distributed quantized gradient-tracking under a dynamic encoding and decoding scheme is considered in [37]. What is missing from the existing literature is the consideration of log-scale quantization for distributed optimization to address resource-constrained communication networks with limited-bandwidth. This is primarily addressed for distributed resource allocation and constraint-coupled optimization [38], but not for distributed learning over networks.

The main contributions of this paper are as follows: (i) we develop a log-scale quantized first-order distributed optimization method that employs network-level gradient tracking. This follows the real-world applications in which quantization is prevalent in networking setups. The proposed solution can be generalized to consider other sector-bound nonlinearities[1] in the data-transmission channels. (ii) We develop the convergence analysis of the proposed log-scale quantized algorithm and provide the bound on the gradient-tracking step-size. (iii) Moreover, unlike existing methods, our approach eliminates the need for *weight-stochastic* design on the network. This is done by adopting a *weight-balanced* design instead, simplifying implementation in the face of node/link failure and enhancing adaptability to *time-varying* setups in the presence of link-level nonlinearities. Through empirical validation, we analyze the convergence of our method for both academic and real-world data-learning setups with distributed and heterogeneous data while considering quantization. This work further explores optimization methodologies that can adapt to both structured networks and ad-hoc multi-agent networks. Structured networks embody a pre-defined organized framework, imposing order and predefined communication patterns on the participating nodes. In contrast, ad-hoc multi-agent networks dynamically adapt and emerge based on the specific demands of the distributed environment, for example, based on the nearest neighbour rule [39]. This paper performs a comparative numerical analysis, unravelling the trade-offs and advantages inherent in using log-scale quantization in contrast to uniform quantization.

---

[1] A nonlinear function $f(x)$ is called sector-bound if it lies in a sector typically defined by two lines passing through the origin, creating an angular region. In mathematical form, there exist positive factors $\overline{\mathcal{K}}$ and $\underline{\mathcal{K}}$ for which we have $\underline{\mathcal{K}}x \leq f(x) \leq \overline{\mathcal{K}}x$.

The rest of the paper is organized as follows. Section II states the problem and terminologies. The proposed distributed algorithm and its convergence analysis are presented in Section III. Experimental validations are given in Section IV. Finally, we conclude the paper in Section V.

II. Assumptions, Terminology, and Problem Statement

The distributed learning problem is defined as optimizing certain cost functions as a sum of some local cost functions distributed over a network of nodes/agents. In machine learning and data mining, the cost function (also known as the loss function or objective function) measures the difference between the predicted values of a model and the ground truth in the training dataset. The cost function quantifies how well or poorly a model performs by assigning a numerical value to the error or deviation between predicted and actual values. The optimization algorithm then adjusts the model's parameters to minimize this cost, effectively improving the model's performance. For example, different loss functions are used for data classification, regression, and support-vector-machine. The global cost is defined as the average of local losses at different nodes; mathematically, this is defined as

$$\min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \tag{1}$$

where each local loss $f_i(\mathbf{x})$ is private to node/agent i. Particularly, each local loss f_i might be decomposable into a finite sum of m_i component cost functions, i.e.,

$$f_i(\mathbf{x}) = \frac{1}{m_i} \sum_{j=1}^{m_i} f_{i,j}(\mathbf{x}). \tag{2}$$

In distributed learning and optimization setup, the training cost might be associated with error-back-propagation in Neural Networks [20], classification gap in SVM [40], or regression training cost [41].

We assume that the nodes share data (local gradient information or auxiliary variables) over a communication network $\mathcal{G}$. Each node locally communicates with its neighbouring nodes $j \in \mathcal{N}_i$ and shares relevant data over the links/channels. In real-world practical applications, the information shared over the channels/links in $\mathcal{G}$ might be subject to quantization (or other nonlinear functions). The nonlinear mapping $h_l(\cdot)$ is sign-preserving, odd, and sector-bound, i.e., it satisfies

$$0 < \underline{\mathcal{K}} \leq \frac{h_l(z)}{z} \leq \overline{\mathcal{K}} \tag{3}$$

An example of such nonlinear mapping is logarithmic (or log-scale) quantization defined as

$$q_l(z) = \text{sgn}(z) \exp\left(\rho \left[\frac{\log(|z|)}{\rho}\right]\right), \tag{4}$$

with $[\cdot]$ as rounding to the nearest integer, and $\text{sgn}(\cdot)$, $\exp(\cdot)$, and $\log(\cdot)$ as the sign, exponential, and (natural) logarithm functions, respectively. The parameter ρ is the logarithmic quantization level for which we have

$$\underline{\mathcal{K}} = 1 - \frac{\rho}{2} \leq \frac{q_l(z)}{z} \leq 1 + \frac{\rho}{2} = \overline{\mathcal{K}}. \tag{5}$$
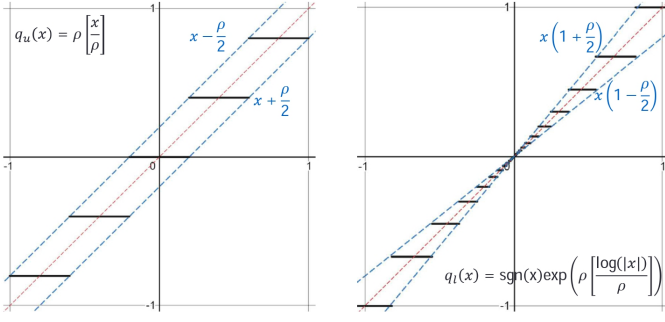
Fig. 1. Uniform quantization (as a non-sector-bound function) versus logarithmic quantization (as a sector-bound function). The logarithmic case leads to finer quantization around zero in contrast to the uniformly quantized case.

Note that, as illustrated in Fig. 1, this nonlinearity is sector-bound, i.e., it is upper/lower-bounded by the two lines $(1 \pm \frac{\rho}{2})z$. This is in contrast to uniform quantization which is not sector-bound, as defined below

$$q_u(z) = \rho \left[\frac{z}{\rho} \right], \qquad (6)$$

where ρ is the uniform quantization level. The two quantization functions are represented in Fig. 1 for comparison. It can be observed that, in contrast to the log-scale quantization, the upper/lower bounds of uniform quantization, defined as $z \pm \frac{\rho}{2}$, are biased from the origin. In this work, we study the convergence under log-scale quantization as compared to uniform quantization. These two are the main types of the nonlinearities governing the data-transmission setup; for example see the following references on uniform quantization [34]–[37] and on logarithmic quantization [38], [42]. These nonlinearities are employed to reduce the communication bandwidth and network traffic in resource-constrained networking setups.

The assumptions to ensure convergence to the unique minimum $\mathbf{x}^*$ of the global cost function F are as follows.

Assumption 1. *The data-transmission network is connected (at all times), weight-balanced (WB), and could be time-varying.*

The WB condition implies that the sum of the weights on incoming links and outgoing links are equal at all nodes. This holds for every time-instant and for all the configurations of the network topology. To give a motivation behind this assumption, note that the WB condition is more relaxed than weight-stochasticity assumption in related works which require the link weights to sum to one for all nodes.

Assumption 2. *Each local cost function f_i has L-smooth and is not necessarily convex, while the global cost F is strongly convex.*

The above assumption is motivated by the discussion in [43] and is used in many references therein saying that it is common to use strong convexity of F as many practical optimization objectives tend to be strongly convex near the minimum. These standard assumptions imply that the global cost function is differentiable and its second derivative is bounded above by some constant L and below by strong-convexity constant μ

where $0 < \mu < L$. Further, the global optimizer $\mathbf{x}^*$ of the cost F exists and is also unique. The following subsection further motivates the study in this paper.

A. Effect of Network Structure and Quantization

This paper investigates the effect of network structure by numerical experimentation. In distributed optimization and learning over networks, the choice of network topology plays a crucial role in determining the convergence rate and optimality gap of the learning algorithms [44], [45]. Different network topologies can have varying effects on the efficiency and speed of convergence [46]–[48]. A general experimental comparison between structured networks (such as exponential graphs[2]) and ad-hoc networking setups (such as geometric graphs[3]) is considered in this paper. Examples of structured networks include networked data centres or hierarchical social networks, and examples of ad-hoc random networks include collaborative robotic networks or wireless sensor networks. Structured networks (such as exponential graphs), characterized by well-defined and organized connections, can often lead to faster convergence rates [45]. In exponential graphs, where nodes have a hierarchical structure, information can propagate more efficiently through the network [49]. Furthermore, structured networks may exhibit lower optimality gaps due to better coordination among the nodes [44]. The organized nature of the network facilitates synchronized updates, potentially leading to quicker convergence to a near-optimal solution. On the other hand, ad-hoc multi-agent networks (such as random geometric graphs), which might lack a clear structure, can have a more complex convergence behaviour. The convergence rate may be slower with a higher optimality gap compared to structured networks because information propagation might not follow an organized pattern. Nodes may require more iterations to align their models and achieve a consensus on the optimal solution [50].

Another concern is to consider switching network topologies addressing time-varying setups, for example, in mobile sensor networks and collaborative swarm robotics. The change in the network topology might be due to packet drops or link failures. Recall that most existing linear algorithms [14]–[21], [23]–[26] necessitate stochastic weight design which is prone to change in the network topology. This is because any change in the network structure violates the stochastic condition and requires the redesign of stochastic weights, for example by applying the algorithms in [51]–[53]. In contrast, this work only requires WB design of matrices A_γ and B_γ that allows handling possible changes in the network topology. This is better illustrated in Fig. 2. To preserve the WB condition in the presence of packet drops, we assume that if a packet is dropped on $i \to j$ link then $j \to i$ link is automatically deleted or, in other words, the agent i does not apply the packet received from j (for example, by setting the associated weight equal to zero), and the entire link is removed. This assumption is

[2]In exponential graphs, as the number of nodes increases exponentially, the degree at each node increases linearly.

[3]In geometric graphs, two nodes are connected if they are in physical proximity.
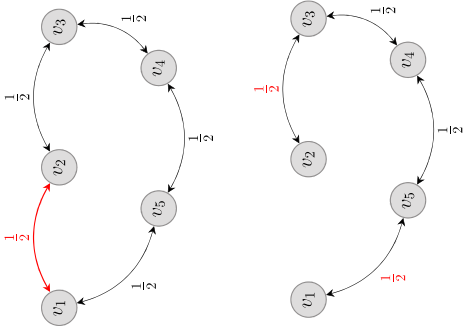
Fig. 2. This figure gives an example graph topology which is both stochastic and WB. The red link represents an unreliable channel that might be subject to failure or packet drop. By removing this link, the network preserves the WB condition but loses stochasticity. Therefore, many weight-stochastic algorithms in the literature need to *redesign* the weights for convergence.

justified in existing literature [52], [54] saying that assuming common knowledge, both agents i, j are aware of the delivery or loss of the packets over the mutual link.

We address more findings in terms of data quantization over channels. First, it should be noted that logarithmic quantization, as compared to uniform quantization, is more efficient in scenarios where the optimization problem involves a wide range of gradient magnitudes. This follows the sector-bound nature of the log-scale quantization, as illustrated in Fig. 1. Smaller values, which are key for convergence, can be represented more accurately (with more bits) under logarithmic quantization. This may lead to a smaller optimality gap and more efficient convergence, for example in optimal resource allocation [55]. The lack of precision in uniform quantization, especially for small values, might slow down convergence and increase the optimality gap in the range of values where accurate representation of gradients or weights is crucial [56]. On the other hand, using finer bits for small values under log-quantization increases the communication complexity as compared to the uniform quantization [57]. This implies that log-quantization, in general, adds more complexity to the communication network as compared to uniform quantization. In terms of network structure in the presence of quantization, although exponential graphs are structured and can facilitate efficient quantized communication, their optimality gap might be considerably affected by quantization inaccuracy. In other words, the efficacy of the network structure might be dominated by the effect of quantization. These comments are investigated by the numerical experiments in Section IV.

III. DISTRIBUTED OPTIMIZATION OVER NETWORKS: ALGORITHM AND MAIN RESULTS

A. Algorithm

This section presents our main algorithm for learning over networks. The methodology is based on two WB matrices $A_\gamma = [a_{ij}^\gamma] \in \mathbb{R}^{n \times n}$ and $B_\gamma = [b_{ij}^\gamma] \in \mathbb{R}^{n \times n}$ that capture the weights for data-sharing over the network, one for consensus on the states and the other for gradient-tracking dynamics. The index $\gamma : k \mapsto \Gamma$ denotes the associated network topology that may change over time, whereas Γ denotes the set of all possible

topologies of the data-sharing network. The graph is all-time connected, i.e., all the network topologies in Γ are connected. The two matrices A_γ and B_γ can be designed independently, matrix A_γ to characterize the weights on the local state value $\mathbf{x}_i$ and matrix B_γ to characterize the weights on the auxiliary variable $\mathbf{y}_i$ for gradient tracking. These weight matrices are WB over $\mathcal{G}$ as discussed before, where the entries satisfy the following:

$$A_\gamma(i,j) = \begin{cases} a_{ij}^\gamma > 0, & \text{If } j \in \mathcal{N}_i \\ 0, & \text{Otherwise.} \end{cases} \quad \sum_{j=1}^n a_{ij}^\gamma = \sum_{i=1}^n a_{ij}^\gamma \quad (7)$$

$$B_\gamma(i,j) = \begin{cases} b_{ij}^\gamma > 0, & \text{If } j \in \mathcal{N}_i \\ 0, & \text{Otherwise.} \end{cases} \quad \sum_{j=1}^n b_{ij}^\gamma = \sum_{i=1}^n b_{ij}^\gamma \quad (8)$$

Let $\overline{A}_\gamma$ and $\overline{B}_\gamma$ denote the Laplacian matrices associated with A_γ and B_γ, respectivly. The definition follows as

$$\overline{A}_\gamma(i,j) = \begin{cases} -\sum_{i=1}^n a_{ij}^\gamma, & i = j \\ a_{ij}^\gamma, & i \neq j. \end{cases} \quad (9)$$

$$\overline{B}_\gamma(i,j) = \begin{cases} -\sum_{i=1}^n b_{ij}^\gamma, & i = j \\ b_{ij}^\gamma, & i \neq j. \end{cases} \quad (10)$$

It is known that for connected networks the eigenvalues of these Laplacian matrices are all in the left-half-plane except one zero eigenvalue [54]. The absolute value of the second largest eigenvalue of the laplacian matrices, denoted by $|\lambda_2^A|$ and $|\lambda_2^B|$, play a key role in the convergence and learning rate. These values are known as the algebraic connectivity. Given the weight matrices and switching signal γ, the local learning and optimization formulation is formally described below:

$$\dot{\mathbf{x}}_i = \sum_{j=1}^n a_{ij}^\gamma (h_l(\mathbf{x}_j) - h_l(\mathbf{x}_i)) - \alpha \mathbf{y}_i, \quad (11)$$

$$\dot{\mathbf{y}}_i = \sum_{j=1}^n b_{ij}^\gamma (h_l(\mathbf{y}_j) - h_l(\mathbf{y}_i)) + \partial_t \nabla f_i(\mathbf{x}_i), \quad (12)$$

The solution by (11)-(12) is composed of two dynamics: one to derive the agents to reach agreement on the optimization variable $\mathbf{x}$ and one to track the consensus on the local gradients. For the latter, we introduce an auxiliary variable $\mathbf{y}$ that tracks the average of gradients in a consensus-based setup and this variable derive the first dynamics toward the global minimum, while both dynamics address log-quantized communication. The log-scale quantization keeps the algorithm efficient in terms of communication with better performance as more resources are allocated near the origin. In contrast to most existing methods with weight-stochastic matrices (introduced in Section I), the proposed dynamics work under WB design and under switching network topologies. These make it easier to handle link-failure as discussed in Fig. 2. Moreover, the link (or channel) nonlinearity $h_l(\cdot)$ is not considered in the existing literature, which allows to address log-scale data quantization by setting $h_l(z) = q_l(z)$ given in Eq. (4). The discrete-time version of our methodology is summarized in Algorithm 1.

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3526967

5

Algorithm 1: Local learning at each node i.

1 **Data:** $f_{i,j}(\mathbf{x})$, $\mathcal{G}_\gamma$, A_γ, B_γ, α
2 **Initialization:** $\mathbf{y}_i(0) = \mathbf{0}_p$, random $\mathbf{x}_i(0)$
3 **for** $k = 0, 1, 2, \ldots$ **do**
4 $\quad$ Each node i receives $h_l(\mathbf{x}_j^k)$ and $h_l(\mathbf{y}_j^k)$ from $j \in \mathcal{N}_i$;
5 $\quad$ $\mathbf{x}_i^{k+1} \leftarrow \mathbf{x}_i^k + \sum_{j=1}^n a_{ij}^\gamma(h_l(\mathbf{x}_j^k) - h_l(\mathbf{x}_i^k)) - \alpha \mathbf{y}_i^k$;
6 $\quad$ $w_i^k \leftarrow \nabla f_i(\mathbf{x}_i^{k+1}) - \nabla f_i(\mathbf{x}_i^k)$;
7 $\quad$ $\mathbf{y}_i^{k+1} \leftarrow \mathbf{y}_i^k + \sum_{j=1}^n b_{ij}^\gamma(h_l(\mathbf{y}_j^k) - h_l(\mathbf{y}_i^k)) + w_i^k$;
8 **Return:** optimal $\mathbf{x}^*$ and F^*;

Note that the learning rate tightly depends on the network topology and the algebraic connectivity. Summing the states over all the nodes we have

$$\sum_{i=1}^n \mathbf{y}_i^{k+1} = \sum_{i=1}^n \mathbf{y}_i^k + \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{k+1}) - \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^k), \quad (13)$$

$$\sum_{i=1}^n \mathbf{x}_i^{k+1} = \sum_{i=1}^n \mathbf{x}_i^k - \alpha \sum_{i=1}^n \mathbf{y}_i^k. \quad (14)$$

This follows the WB assumption which implies that

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij}^\gamma(h_l(\mathbf{x}_j^k) - h_l(\mathbf{x}_i^k)) = 0$$

$$\sum_{i=1}^n \sum_{j=1}^n b_{ij}^\gamma(h_l(\mathbf{y}_j^k) - h_l(\mathbf{y}_i^k)) = 0$$

By setting the initial condition as $\mathbf{y}_i(0) = \mathbf{0}_p$ we obtain the following,

$$\sum_{i=1}^n \mathbf{x}_i^{k+1} = -\alpha \sum_{i=1}^n \mathbf{y}_i^{k+1} = -\alpha \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{k+1}), \quad (15)$$

This represents the consensus-type gradient-tracking dynamics that capture the essence of the proposed methodology. This formulation further allows for addressing sector-bound odd nonlinearities h_l (for example log-scale quantization) without violating the tracking property. This along with the time-varying network topology advances the recent state-of-the-art algorithms for data-mining and optimization [20], [25], [26], [31]–[33].

Recall that most of the existing algorithms prove convergence for ideal networking conditions and in the absence of nonlinearities (e.g., quantization and clipping). However, in the presence of nonlinearities and non-ideal networking conditions, the solution may result in a certain optimality gap, which also depends on the structure of the underlying network $\mathcal{G}_\gamma$. The role of the network structure and quantization is also discussed in the simulation section.

B. Proof of Convergence

This section proves the convergence of the distributed dynamics (11)-(12) using eigen-spectrum perturbation-based analysis. We also state some relevant lemmas. The proofs in this section hold for any sign-preserving, odd, sector-bound nonlinear mapping $h_l(\cdot)$, which also includes log-scale

quantization $h_l(z) = q_l(z)$ defined by (4). In the rest of the paper, for notation simplicity, we drop the dependence (t, γ) unless where it is needed.

Lemma 1. *[58] Consider the square matrix $P(\alpha)$ of size n which depends on parameter $\alpha \in \mathbb{R}_{\geq 0}$. Let $P(0)$ has $N < n$ equal eigenvalues $\lambda_1 = \ldots = \lambda_N$, associated with right and left unit eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_N$ and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ (which are linearly independent). Let $P' = \partial_\alpha P(\alpha)|_{\alpha=0}$ and $\lambda_i(\alpha)$ represent its i-th eigenvalue $\lambda_i, i \in \{1, \ldots, N\}$. Then, $\partial_\alpha \lambda_i|_{\alpha=0}$ is the i-th eigenvalue of,*

$$\begin{pmatrix} \mathbf{u}_1^\top P' \mathbf{v}_1 & \ldots & \mathbf{u}_1^\top P' \mathbf{v}_N \\ & \ddots & \\ \mathbf{u}_N^\top P' \mathbf{v}_1 & \ldots & \mathbf{u}_N^\top P' \mathbf{v}_N \end{pmatrix}.$$

Lemma 2. *[59] Given the balanced digraph $\mathcal{G}$ its Laplacian matrix $\overline{A}$ (or $\overline{B}$) has all its eigenvalue in the left-half-plane except one isolated zero eigenvalue with associated non-negative left eigenvector $\mathbf{u}_1^\top$ satisfying $\sum_{i=1}^n u_{1,i} > 0$.*

Theorem 1. *Let Assumptions 1 and 2 hold. For sufficiently small α, all eigenvalues of the dynamics (11)-(12) are in the left-half-plane except m zero eigenvalue.*

Proof. For the linear case (i.e., setting $h_l(z) = z$) the proposed dynamics (11)-(12) can be written in compact form as

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \end{pmatrix} = M(\alpha) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \quad (16)$$

$$M(\alpha) = \begin{pmatrix} \overline{A} \otimes I_m & -\alpha I_{mn} \\ H(\overline{A} \otimes I_m) & \overline{B} \otimes I_m - \alpha H \end{pmatrix}. \quad (17)$$

where $H := \text{diag}[\nabla^2 f_i(\mathbf{x})]$ and $M(\alpha) = M_0 + \alpha M_1$ with

$$M_0 = \begin{pmatrix} \overline{A} \otimes I_m & \mathbf{0}_{mn \times mn} \\ H(\overline{A} \otimes I_m) & \overline{B} \otimes I_m \end{pmatrix},$$

$$M_1 = \begin{pmatrix} \mathbf{0}_{mn \times mn} & -I_{mn} \\ \mathbf{0}_{mn \times mn} & -H \end{pmatrix},$$

Now, considering the nonlinearity $h_l(z) = q_l(z)$[4], one can linearize the nonlinear dynamics (11)-(12) at every time-instant t as its operating point. It is known that the stability of the linearization at every operating point implies the stability of the nonlinear dynamics [60]. To study the stability of the linearized dynamics, we have

$$M_q(t, \alpha, \gamma) = M_q^0 + \alpha M^1 \quad (18)$$

$$\underline{\mathcal{K}} M^0 \preceq M_q^0 \preceq \overline{\mathcal{K}} M^0 \quad (19)$$

$$M_q^0 = Z(t) M^0, \ \underline{\mathcal{K}} I_n \preceq Z(t) \preceq \overline{\mathcal{K}} I_n \quad (20)$$

where M_q^0 is the linearized version of M^0 for the nonlinear dynamics, $Z(t) := \text{diag}[\zeta(t)]$, column vector $\zeta(t) = [\zeta_1(t); \zeta_2(t); \ldots; \zeta_n(t)]$ with $\zeta_i(t) = \frac{q_l(\mathbf{x}_i)}{\mathbf{x}_i}$ (or equivalently $q_l(\mathbf{x}(t)) = Z(t)\mathbf{x}(t)$ at every time-instant t). Recall from Eq. (3) that $\underline{\mathcal{K}} \leq \zeta_i(t) \leq \overline{\mathcal{K}}$. Therefore, we have

$$\overline{A}_Z = \overline{A} Z(t), \ \overline{B}_Z = \overline{B} Z(t) \quad (21)$$

---

[4]Note that the proof holds for general sign-preserving sector-bound nonlinearities. We particularly state the solution for log-scale quantization as an example.

The above helps to relate the eigen-spectrum of the linear dynamics (16) to that of the nonlinear case associated with M_q. Note that $Z(t)$ is a diagonal matrix from its definition, and we have

$$\det(\overline{A}_Z - \lambda I_{mn}) = \det(\overline{A} - \lambda Z(t)^{-1}), \quad (22)$$

$$\det(\overline{B}_Z - \lambda I_{mn}) = \det(\overline{B} - \lambda Z(t)^{-1}), \quad (23)$$

with λ denoting the eigenvalue. Therefore, from (19)-(20), one can relate the eigenspectrum of the linearized version with that of the linear case as

$$\underline{\mathcal{K}}\sigma(M^0) \le \sigma(M_q^0) \le \overline{\mathcal{K}}\sigma(M^0) \quad (24)$$

where the eigenspectrum of the linear case follows the block triangular form of M_0 as,

$$\sigma(M^0) = \sigma(\overline{A} \otimes I_m) \cup \sigma(\overline{B} \otimes I_m) \quad (25)$$

For the rest of the proof, we study the eigenspectrum perturbation of M^0 and then extend the results to M_q^0 using Eq. (24). From Lemma 2, both Laplacian matrices $\overline{A}$ and $\overline{B}$ have one isolated zero eigenvalue and the rest in the left-half-plane for all cases of the switching networks. Therefore, m sets of (possibly changing) eigenvalues of M_0, associated with dimensions $j = \{1, \ldots, m\}$ are in the form,

$$\mathrm{Re}\{\lambda_{2n,j}\} \le \ldots \le \mathrm{Re}\{\lambda_{3,j}\} < \lambda_{2,j} = \lambda_{1,j} = 0,$$

Next, we consider αM_1 as a perturbation to matrix M_0. This is done using Lemma 1. In particular, we check how the perturbation term αM_1 affects the two zero eigenvalues $\lambda_{1,j}$ and $\lambda_{2,j}$ of M_0. Denote the perturbed zero eigenvalues by $\lambda_{1,j}(\alpha)$ and $\lambda_{2,j}(\alpha)$ associated to M. For all WB and connected switching network topologies, the right and left unit eigenvectors of $\lambda_{1,j}$, $\lambda_{2,j}$ follow from Lemma 2 and [59] as[5],

$$V = [V_1 \ V_2] = \frac{1}{\sqrt{n}} \begin{pmatrix} \mathbf{1}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{1}_n \end{pmatrix} \otimes I_m \quad (26)$$

$$U^\top = [U_1 \ U_2]^\top = \begin{pmatrix} \mathbf{u}_1 & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{u}_2 \end{pmatrix}^\top \otimes I_m \quad (27)$$

Recalling $M(\alpha) = M_0 + \alpha M_1$ we have $\partial_\alpha dM(\alpha)|_{\alpha=0} = M_1$. Then, Lemma 1 implies that,

$$U^\top M_1 V = \begin{pmatrix} \mathbf{0}_{m \times m} & \times \\ \mathbf{0}_{m \times m} & -(\mathbf{u}_2 \otimes I_m)^\top H(\frac{1}{\sqrt{n}} \mathbf{1}_n \otimes I_m) \end{pmatrix}. \quad (28)$$

The definition of H, Lemma 2, and Assumption 2 implies that

$$-(\mathbf{u}_2 \otimes I_m)^\top H(\frac{1}{\sqrt{n}} \mathbf{1}_n \otimes I_m) = -\sum_{i=1}^n \frac{u_{2,i}}{\sqrt{n}} \boldsymbol{\nabla}^2 f_i(\mathbf{x}_i) \prec 0, \quad (29)$$

where $u_{2,i}$ is the ith element of $\mathbf{u}_2$. Then, using Lemma 1, the perturbations $\partial_\alpha d\lambda_{1,j}|_{\alpha=0}$ and $\partial_\alpha d\lambda_{2,j}|_{\alpha=0}$ are defined based on the eigenvalues of the matrix in (28). Since this matrix is triangular and from Eq. (29), $\partial_\alpha \lambda_{1,j}|_{\alpha=0} = 0$ and $\partial_\alpha \lambda_{2,j}|_{\alpha=0} < 0$. Therefore, for all sets of switching

---

[5]The normalizing factors of the unit vectors might be ignored as in the followings we only care about the sign of the terms in $U^\top M_1 V$, not the exact values.

network Laplacians, the perturbation analysis imply that αM_1 pushes m zero eigenvalues $\lambda_{2,j}(\alpha)$ of M toward the left-half-plane and the other m zero eigenvalues $\lambda_{1,j}(\alpha)$ remain at zero. This implies that sufficiently small α gives the eigen-spectrum of M as

$$\mathrm{Re}\{\lambda_{2n,j}(\alpha)\} \le \ldots \le \mathrm{Re}\{\lambda_{3,j}(\alpha)\} \le \lambda_{2,j}(\alpha) < \lambda_{1,j}(\alpha) = 0, \quad (30)$$

This completes the proof. $\qquad \square$

Theorem 2. *Given that the Assumptions (1)-(2) and conditions in Theorem 1 hold, for $0 < \alpha < \frac{\min\{|\mathrm{Re}\{\lambda_2^A\}|, |\mathrm{Re}\{\lambda_2^B\}|\}}{L\overline{\mathcal{K}}}$, the proposed dynamics (11)-(12) converges to the optimizer $[\mathbf{x}^* \otimes \mathbf{1}_n; \mathbf{0}_{nm}]$.*

Proof. First, we determine the admissible bound on α for proof of convergence. Recall from [61, Appendix] that one can relate the spectrum of $M(\alpha)$ in (17) to α. For ease of notation, from this point onward we derive the bound for $m = 1$ (but it holds for any $m > 1$). Performing row/column permutations in [61, Eq. (18)] $\sigma(M)$ can be determined from the following,

$$\det(\alpha I_n)\det(H(\overline{A}) + (\overline{B} - \alpha H - \lambda I_n)(\frac{1}{\alpha})(\overline{A} - \lambda I_n)) = 0.$$

which can be simplified as

$$\det(I_n)\det((\overline{A} - \lambda I_n)(\overline{B} - \lambda I_n) + \alpha\lambda H) = 0 \quad (31)$$

Similar to the proof of Theorem 1, for stability we need to find the admissible range of α values for which the eigenvalues λ remain in the left-half-plane, except one zero eigenvalue. The analysis here is based on the fact that the eigenvalues are continuous functions of the matrix elements [62]. It is clear that $\alpha = 0$ satisfies Eq. (31) and gives $\det((\overline{W} - \lambda I_n)(\overline{A} - \lambda I_n)) = 0$. This leads to the eigen-spectrum following as $\sigma(M) = \sigma(\overline{A}) \cup \sigma(\overline{W})$ with two zero eigenvalues for all switching network topologies. We need to find the other root $\overline{\alpha} > 0$, which gives the admissible range as $0 < \alpha < \overline{\alpha}$ for the stability of $M(\alpha)$. This follows the continuity of $\sigma(M)$ as a function of α [62]. With some abuse of notation, for any $\mathrm{Re}\{\lambda\} < 0$ one can reformulate (31) as

$$\det((\overline{B} - \lambda I_n \pm \sqrt{\alpha|\lambda|H})(\overline{A} - \lambda I_n \mp \sqrt{\alpha|\lambda|H}) \\ \pm \sqrt{\alpha|\lambda|H}(\overline{A} - \overline{B})) = 0 \quad (32)$$

This follows the diagonal form of H. First, we consider the same consensus matrix for both variables $\mathbf{x}, \mathbf{y}$. Set $\overline{A} = \overline{B}$ and we have

$$\det(\overline{A} - \lambda I_n \mp \sqrt{\alpha|\lambda|H}) = \det(\overline{A} - \lambda(I_n \mp \sqrt{\frac{\alpha H}{|\lambda|}})) = 0$$

This implies that fo $\lambda \in \sigma(\overline{A})$, its perturbed eigenvalue is $\lambda(1 \pm \sqrt{\frac{\alpha H}{|\lambda|}})$. Therefore, for $\lambda \ne 0$, the min value of $\overline{\alpha}$ that makes this term zero (at the edge of instability) satisfies the following,

$$\overline{\alpha} = \underset{\alpha}{\mathrm{argmin}} |1 - \sqrt{\frac{\alpha H}{|\lambda|}}| \ge \frac{\min\{|\mathrm{Re}\{\lambda_j\}| \ne 0\}}{\max\{H_{ii}\}} = \frac{|\mathrm{Re}\{\lambda_2\}|}{L} \quad (33)$$

where we recalled $H \preceq LI_n$ from Assumption 2. Recalling perturbed formulation in Theorem 1, this gives the admissible range for α for which $M(\alpha)$ has all its eigenvalues at the left-half-plane except one zero as

$$0 < \alpha < \overline{\alpha} := \frac{|\operatorname{Re}\{\lambda_2\}|}{L} \tag{34}$$

For general nonlinear dynamics (11)-(12) satisfying Eq. (19)-(20), the admissible range changes to

$$0 < \alpha < \overline{\alpha} := \frac{|\operatorname{Re}\{\lambda_2\}|}{L\overline{\mathcal{K}}} \tag{35}$$

For the more general case of $\overline{A} \neq \overline{B}$, this is generalized as,

$$0 < \alpha < \frac{\min\{|\operatorname{Re}\{\lambda_2^A\}|, |\operatorname{Re}\{\lambda_2^B\}|\}}{L\overline{\mathcal{K}}} =: \overline{\alpha} \tag{36}$$

This admissible range of α depends only on $\sigma(\overline{A}), \sigma(\overline{B})$ and holds for any value of $m \geq 1$. Next, we prove convergence under this admissible bound. It is clear that the optimizer $[\mathbf{x}^* \otimes \mathbf{1}_n; \mathbf{0}_{nm}] \in \mathbb{R}^{2mn}$ is the invariant set of dynamics (11)-(12) and belongs to null-space of M_q. Recall that for $0 < \alpha < \frac{\min\{|\operatorname{Re}\{\lambda_2^A\}|, |\operatorname{Re}\{\lambda_2^B\}|\}}{L\overline{\mathcal{K}}}$, the eigenvalues of the linearized system dynamics associated with M_q are stable except one isolated zero eigenvalue. This is irrespective of the time-variation of $\mathcal{G}$ and M_q. Define the variable δ as the distance between the system state and the optimizer,

$$\delta = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} - \begin{pmatrix} \mathbf{x}^* \\ \mathbf{0}_{mn} \end{pmatrix} \in \mathbb{R}^{2mn}.$$

and the positive-semi-definite Lyapunov function $\mathcal{V}(\delta) = \frac{1}{2}\delta^\top\delta = \frac{1}{2}\|\delta\|_2^2$. As $\delta \to \mathbf{0}_{2mn}$ the state variables converge to the optimizer. We have,

$$\dot{\delta} = M_q \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} - M_q \begin{pmatrix} \mathbf{x}^* \\ \mathbf{0}_{mn} \end{pmatrix} = M_q\delta. \tag{37}$$

This implies that $\dot{\mathcal{V}} = \delta^\top\dot{\delta} = \delta^\top M_q\delta$ and from [59, Sections VIII-IX], we have

$$\dot{\mathcal{V}} = \delta^\top M_q\delta \leq \max_{1 \leq j \leq m} \operatorname{Re}\{\lambda_{2,j}\}\delta^\top\delta, \tag{38}$$

where $\max_{1 \leq j \leq m} \operatorname{Re}\{\lambda_{2,j}\}$ is the real part of the largest negative eigenvalue of M_q at all times (following from Theorem 1). Note that $\dot{\mathcal{V}}$ is negative-definite for all $\delta \neq \mathbf{0}_{2mn}$ and is zero for $\delta = \mathbf{0}_{2mn}$. From Lyapunov theorem, this network of integrators is asymptotically globally stable and $\delta \to \mathbf{0}_{2mn}$ implies that the solution converges to the optimizer. $\qquad\square$

C. Discussions

We now provide some important remarks in the context of Theorems 1 and 2 and the sector-bound property.

Remark 1. *For convergence we only need $(\mathbf{1}_n \otimes I_p)^\top H(\mathbf{1}_n \otimes I_p) \succ 0$, i.e., strong-convexity of the global objective function $F(\mathbf{x})$. Thus, the local cost functions $f_i(\mathbf{x}_i)$ might be non-convex, i.e., $\nabla^2 f_i(\mathbf{x}_i)$ is not necessarily positive while their summation is positive to ensure that $F(\mathbf{x})$ is strongly convex. An academic example of such a case is given in the simulation. Therefore, the convexity assumption in this paper is more relaxed as*

compared to many existing literature. Note that the log-scale quantization in this work is independent of Assumption 2.

Remark 2. *From Theorem 1, the proposed dynamics (11)-(12) has one zero eigenvalue related to the agreement (or consensus) on the states, i.e., all the auxiliary states reach consensus on $\mathbf{y}_i = 0$ and all the main states reach consensus on $\mathbf{x}_i = \mathbf{x}^*$. On the other hand, the rest of the eigenvalues remain in the left-half-plane which results in the stability of the proposed dynamics and convergence toward the optimal point.*

Remark 3. *Recall that one can approximate the convergence rate (or decay rate) of the linearized system dynamics given by Eq. (37) which is concluded from the dynamics (11)-(12). Following the eigen-spectrum of the system matrix $M_q(\alpha)$ and using Eq. (38), the convergence rate is (at least) $\exp(\max_{1 \leq j \leq m} \operatorname{Re}\{\lambda_{2,j}\}t)$. This shows linear convergence in log-scale which is approved by the simulations in the next section. Also, for the special case of no quantization (i.e., $h_l(\mathbf{x}_i) = \mathbf{x}_i$ and $h_l(\mathbf{y}_i) = \mathbf{y}_i$) and for strongly convex cost functions with stochastic weight matrices A_γ, B_γ the convergence rate can be calculated via similar analysis as in [63, Lemmas 3.2-3.4].*

Remark 4. *Assume that the linear version of the proposed algorithm 1 with $h_l(\mathbf{x}_j^k) = \mathbf{x}_j^k$ and $h_l(\mathbf{y}_j^k) = \mathbf{y}_j^k$ gives certain optimality gap. Then, recalling Eq. (3), one can see that $\underline{\mathcal{K}}(\mathbf{x}_j^k - \mathbf{x}_i^k) \leq h_l(\mathbf{x}_j^k) - h_l(\mathbf{x}_i^k) \leq \overline{\mathcal{K}}(\mathbf{x}_j^k - \mathbf{x}_i^k)$ and $\underline{\mathcal{K}}(\mathbf{y}_j^k - \mathbf{y}_i^k) \leq h_l(\mathbf{y}_j^k) - h_l(\mathbf{y}_i^k) \leq \overline{\mathcal{K}}(\mathbf{y}_j^k - \mathbf{y}_i^k)$. Therefore, the optimality gap is of the same order as in the linear case multiplied by a factor of $\overline{\mathcal{K}}$ and $\underline{\mathcal{K}}$. Thus, for the logarithmic quantization nonlinearity with $\overline{\mathcal{K}} = 1 + \frac{\rho}{2}$ and $\underline{\mathcal{K}} = 1 - \frac{\rho}{2}$, the optimality gap gets arbitrarily close to the linear case by choosing sufficiently small ρ. This is illustrated in the next section via numerical simulations.*

Remark 5. *Uniform quantization in contrast to logarithmic quantization is not sector-bound; as shown in Fig. 1, the bounds $x \pm \frac{\rho}{2}$ do not pass through the origin. Therefore, the results on convergence and optimality do not hold for uniformly quantized data. In fact, some works in the literature show that uniform quantization generally results in a larger optimality gap, see an example for quantized resource allocation in [64]. In the next section, we clearly show by numerical simulation that log-scale quantization as compared to uniform quantization results in a smaller optimality gap.*

IV. NUMERICAL EXPERIMENTS

In this section, we perform simulations on both real and academic setups to verify our results.

A. Academic Example

We consider a network of $n = 16$ agents to minimize the following cost function in a distributed and collaborative way:

$$f_{i,j}(x_i) = 4x_i^2 + 3\sin^2(x_i) + a_{i,j}\cos(x_i) + b_{i,j}x_i, \tag{39}$$

with random parameters $a_{i,j}$ and $b_{i,j}$ in the range $[-10, 10]$ such that $\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j} = 0$ and $\sum_{i=1}^{n}\sum_{j=1}^{m} b_{i,j} = 0$ and $a_{i,j}, b_{i,j} \neq 0$. Note that $f_{i,j}(x_i)$ is not necessarily convex,
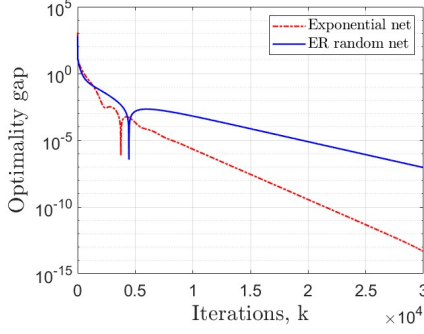
Fig. 3. Logarithmically quantized distributed optimization over an exponential network versus an ER random network of the same size.

i.e., $\nabla^2 f_i(\mathbf{x})$ might be negative at some points, while the global cost summing all $f_{i,j}(x_i)$ is strongly convex. We apply Algorithm 1 over random ad-hoc networks and structured exponential networks for both uniform and log-scale quantization. First, we compare the convergence rate and optimality gap over a time-varying (switching) exponential network (as a structured network) versus a randomly changing Erdos-Renyi (ER) network subject to log-scale quantization with $\rho = \frac{1}{128}$. For the exponential network, we change the link weights every 100 iterations and for the ER random network both the link weights and network structure change every 100 iterations. The result is shown in Fig. 3. The simulation results clearly verify that the solution converges over switching networks and also, as stated in Section II-A, the structured network gives a smaller optimality gap and faster convergence.

Next, we compare distributed optimization over exponential graphs under time-varying weights subject to uniform quantization versus log-scale quantization in Fig. 4 for different quantization levels ρ. Evidently from the figure, log-scale quantization results in a smaller optimality gap as compared to uniform quantization which verifies the statements in Section II-A. This shows the advantage of using logarithmic quantization over uniform quantization. Also, one can see that the log-scale quantization level has negligible effect on the convergence rate and optimality gap. Note that, from Remark 3, the convergence rate is linear in the log-scale. Therefore, one can approximate the convergence rate of the log-quantized case by the linear (non-quantized) solution. Also, note that considering the computational limitation of MATLAB the existing optimality gap of the log-quantized dynamics in the simulation is acceptable.

Next, we compare our proposed log-quantized algorithm with some existing literature: **TV-AB** [65], finite-time [66], and fixed-time [67] algorithms. It should be noted that none of these algorithms are quantized and these algorithms are designed to solve strongly-convex cost functions. Therefore, we need to decrease the non-convexity parameter of the cost model (39) and set $a_{i,j} \in [-1, 1]$ to make it strongly-convex for this simulation. For our log-quantized solution we set $\rho = \frac{1}{4}$. The comparison is given in Fig. 5. It is clear from the figure that our solution, although quantized, has the same performance as the non-quantized AB-type solution and shows better performance than fixed-time and finite-time solutions.
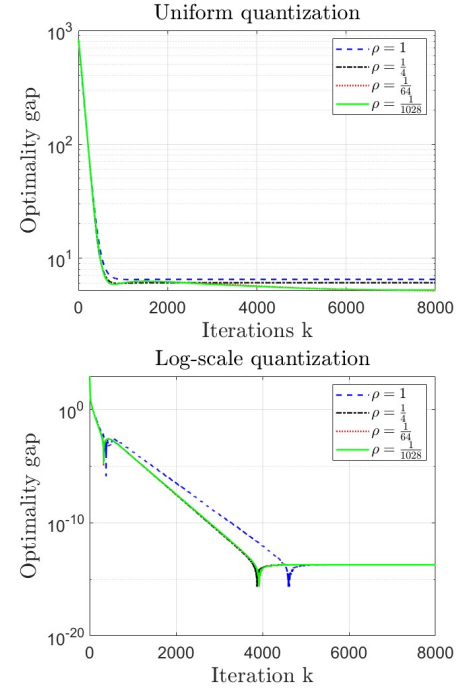

Fig. 4. Optimality gap of uniformly quantized versus logarithmically quantized distributed learning over exponential network.
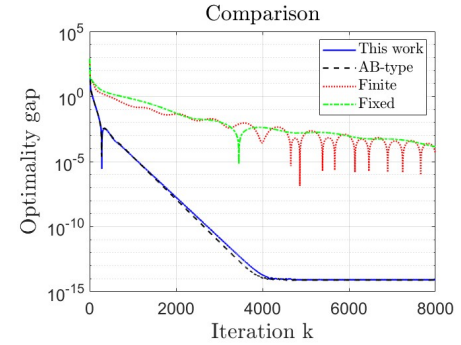

Fig. 5. The comparison of the proposed log-quantized algorithm with the existing non-quantized algorithms for a strongly-convex cost model.

B. Real Data-Set Example

In this section, we analyze some real data from [25]. We randomly select some images from the MNIST data set and classify them using logistic regression with a convex regularizer. The set of $N = 12000$ labelled images are used for classification, distributed among the $n = 16$ agents/nodes. The overall cost function is defined as

$$\min_{\mathbf{b},c} F(\mathbf{b}, c) = \frac{1}{n} \sum_{i=1}^{n} f_i \qquad (40)$$

where each node i has access to a batch of $m_i = 750$ sample data and locally minimizes the following training cost:

$$f_i(\mathbf{x}) = \frac{1}{m_i} \sum_{j=1}^{m_i} \ln(1 + \exp(-(\mathbf{b}^\top x_{i,j} + c) y_{i,j})) + \frac{\lambda}{2} \|\mathbf{b}\|_2^2. \qquad (41)$$

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3526967
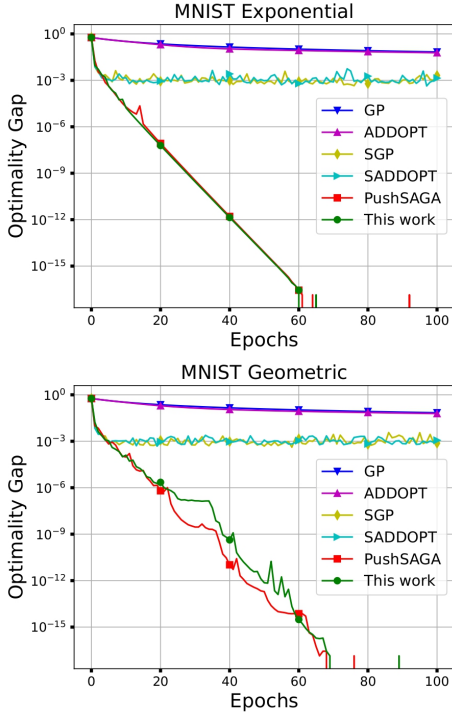
9

Fig. 6. Comparison between the performance of the proposed algorithm (in the absence of quantization) with some existing algorithms in the literature.
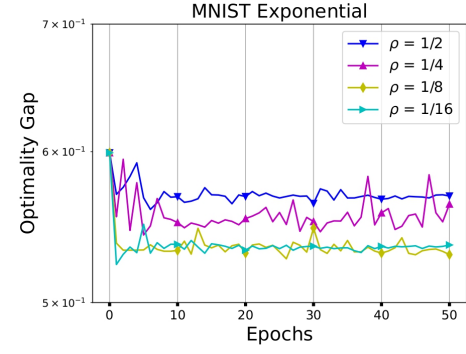

Fig. 7. Optimality gap of distributed learning of MNIST dataset over exponential graphs subject to uniform quantization with different quantization levels.
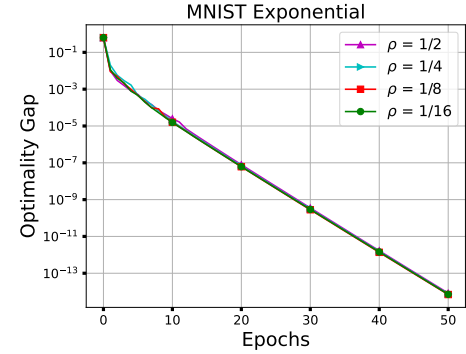

Fig. 8. Optimality gap of distributed learning of MNIST dataset over exponential graphs subject to log-scale quantization with different quantization levels.

with $\mathbf{b}, c$ as the parameters of the separating hyperplane. We run and compare the distributed training over both geometric (as an ad-hoc network) and exponential graphs (as a structured network). The optimality gap (also known as the optimization residual) in the figures is defined as $F(\overline{\mathbf{x}}^k) - F(\mathbf{x}^*)$ with $\overline{\mathbf{x}}^k = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^k$.

First, we compare some existing algorithms in the literature with the proposed Algorithm 1. The following algorithms are used for comparison: **GP** [14], **SGP** [17], [18], **S-ADDOPT** [20], **ADDOPT** [19], and **PushSAGA** [26]. Fig. 6 presents the comparison results.

Next, we compare training over exponential networks subject to uniform and log-scale quantization for different data quantization levels, see Fig. 7 and Fig. 8. As it can be observed, the uniform quantization results in a larger optimality gap in the training outcome. The same comparison is performed over the geometric network and the result is shown in Fig. 9 and Fig. 10. As expected, uniform quantization results in a larger optimality gap. Also, from Fig. 8 and 10 one can see that the convergence rates for different log-scale quantization rates do not change that much. It is worth noting from the figures that, the optimality gap is generally larger over geometric networks as a non-structured topology. However, the log-scale quantization adds more computational complexity to the algorithm as compared to the uniform quantization. This is because log-quantization is more precise around the origin and requires more bits for transmitting those values.

V. CONCLUSIONS

A. Concluding Remarks

This work proposes a distributed optimization setup for learning over networks. We particularly address nonlinear channels to take into account, for example, possible data quantization. We compare log-scale quantization (as a sector-bound nonlinearity) with uniform quantization (as a non-sector-bound nonlinearity) over both academic and real data. Our results show that the log-scale quantization leads to a smaller optimality gap for distributed training. We further compare
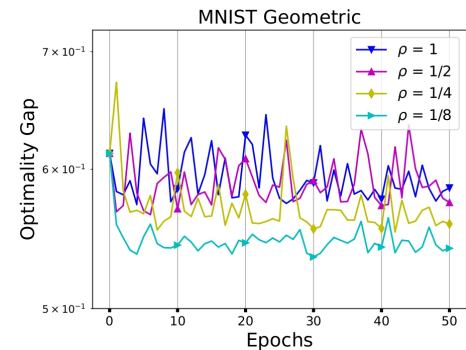

Fig. 9. Optimality gap of distributed learning of MNIST dataset over geometric graphs subject to uniform quantization with different quantization levels.

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3526967
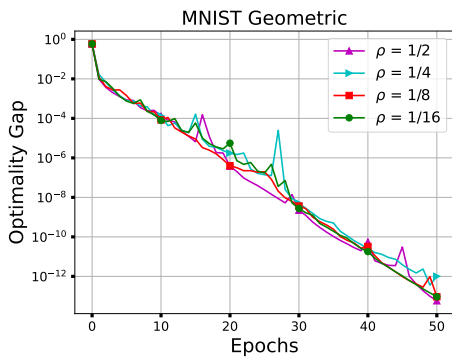
10

Fig. 10. Optimality gap of distributed learning of MNIST dataset over geometric graphs subject to log-scale quantization with different quantization levels.

the quantized setups over exponential and random (geometric and ER) networks, and, as expected, the exponential graph as a structured network results in a smaller optimality gap as compared to geometric and ER graphs representing ad-hoc multi-agent networks.

B. Future Directions

One direction of future research is to more relax Assumption 1 to uniform-connectivity of the network, where the union of the graphs over a finite interval is connected. More detailed theoretical analysis and comparison between log-scale versus uniform quantization for general distributed optimization methods and quantifying the optimality gap of the discretized version for large sampling periods are set as future research. Other future research efforts will focus on considering the effect of quantization on different machine learning techniques, for example, distributed minimization of error-back-propagation in Neural-Networks and distributed reinforcement learning. Application to distributed optimal control is another direction of research interest.

REFERENCES

[1] H. Raja and W. U. Bajwa, "Cloud K-SVD: a collaborative dictionary learning algorithm for big, distributed data," *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 173–188, 2015.

[2] S. Abbasian Dehkordi, K. Farajzadeh, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, and M. Abbasian Dehkordi, "A survey on data aggregation techniques in IoT sensor networks," *Wireless Networks*, vol. 26, pp. 1243–1263, 2020.

[3] M. A. Rahimian, F. Yu, and C. Hurtado, "Differentially private network data collection for influence maximization," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 2795–2797.

[4] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[5] S. Hashempour, A. A. Suratgar, and A. Afshar, "Distributed nonconvex optimization for energy efficiency in mobile ad hoc networks," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5683–5693, 2021.

[6] M. Doostmohammadian, W. Jiang, M. Liaquat, A. Aghasi, and H. Zarrabi, "Discretized distributed optimization over dynamic digraphs," *IEEE Transactions on Automation Science and Engineering*, 2024, Early access.

[7] E. Camponogara and H. F. Scherer, "Distributed optimization for model predictive control of linear dynamic networks with control-input and output constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 233–242, 2010.

[8] S. Oh, H. Sim, S. Lee, and J. Lee, "Automated log-scale quantization for low-cost deep neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 742–751.

[9] F. Sattler, A. Marban, R. Rischke, and W. Samek, "CFD: communication-efficient federated distillation via soft-label quantization and delta coding," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2025–2038, 2021.

[10] N. Khademi Kalantari and S. M. Ahadi, "A logarithmic quantization index modulation for perceptually better data hiding," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1504–1517, 2010.

[11] J. Cai, M. Takemoto, and H. Nakajo, "A deep look into logarithmic quantization of model parameters in neural networks," in *Proceedings of the 10th International Conference on Advances in Information Technology*, 2018, pp. 1–8.

[12] T. C. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus using probabilistic quantization," in *IEEE/SP 14th Workshop on Statistical Signal Processing*. IEEE, 2007, pp. 640–644.

[13] S. S. Pereira and A. Pages-Zamora, "Distributed consensus in wireless sensor networks with quantized information exchange," in *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*. IEEE, 2008, pp. 241–245.

[14] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[15] C. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 763–768, 2013.

[16] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, pp. 516–545, 2010.

[17] A. Spiridonoff, A. Olshevsky, and I. Paschalidis, "Robust asynchronous stochastic gradient-push: Asymptotically optimal and network-independent performance for strongly convex functions," *Journal of machine learning research*, vol. 21, no. 58, 2020.

[18] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

[19] C. Xi, R. Xin, and U. A Khan, "ADD-OPT: accelerated distributed directed optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1329–1339, 2017.

[20] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "S-ADDOPT: Decentralized stochastic first-order optimization over directed graphs," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 953–958, 2020.

[21] D. T. A. Nguyen, D. T. Nguyen, and A. Nedić, "Geometric convergence of distributed heavy-ball nash equilibrium algorithm over time-varying digraphs with unconstrained actions," *IEEE Control Systems Letters*, 2023.

[22] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[23] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2566–2581, 2019.

[24] D. T. A. Nguyen, D. T. Nguyen, and A. Nedić, "Accelerated ab/push-pull methods for distributed optimization over time-varying directed networks," *IEEE Transactions on Control of Network Systems*, 2023.

[25] M. I. Qureshi and U. A. Khan, "Stochastic first-order methods over distributed data," in *IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2022, pp. 405–409.

[26] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Push-SAGA: a decentralized stochastic algorithm with variance reduction over directed graphs," *IEEE Control Systems Letters*, vol. 6, pp. 1202–1207, 2021.

[27] R. Xin, U. A. Khan, and S. Kar, "A fast randomized incremental gradient method for decentralized nonconvex optimization," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5150–5165, 2021.

[28] X. Gong, Y. Cui, J. Shen, J. Xiong, and T. Huang, "Distributed optimization in prescribed-time: Theory and experiment," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 2, pp. 564–576, 2021.

[29] L. Liu and G. Yang, "Distributed fixed-time optimal resource management for microgrids," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 404–412, 2022.

[30] S. Chen, H. Jiang, Z. Yu, and F. Zhao, "Distributed optimization of single-integrator systems with prescribed-time convergence," *IEEE Systems Journal*, 2022.

[31] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization design for distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2107–2120, 2016.

[32] S. Horváth, D. Kovalev, K. Mishchenko, P. Richtárik, and S. Stich, "Stochastic distributed learning with gradient quantization and double-variance reduction," *Optimization Methods and Software*, vol. 38, no. 1, pp. 91–106, 2023.

[33] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[34] Y. Xiong, L. Wu, K. You, and L. Xie, "Quantized distributed gradient tracking algorithm with linear convergence in directed networks," *IEEE Transactions on Automatic Control*, vol. 68, no. 9, pp. 5638–5645, 2022.

[35] Z. Song, L. Shi, S. Pu, and M. Yan, "Compressed gradient tracking for decentralized optimization over general directed networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1775–1787, 2022.

[36] Y. Liao, Z. Li, K. Huang, and S. Pu, "A compressed gradient tracking method for decentralized optimization with linear convergence," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5622–5629, 2022.

[37] Y. Kajiyama, N. Hayashi, and S. Takai, "Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1254–1261, 2020.

[38] M. Doostmohammadian, A. Aghasi, M. Vrakopoulou, H. R. Rabiee, U. A. Khan, and T. Charalambous, "Distributed delay-tolerant strategies for equality-constraint sum-preserving resource allocation," *Systems & Control Letters*, vol. 182, pp. 105657, 2023.

[39] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[40] M. Doostmohammadian, A. Aghasi, T. Charalambous, and U. A. Khan, "Distributed support vector machines over dynamic balanced directed networks," *IEEE Control Systems Letters*, vol. 6, pp. 758–763, 2021.

[41] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *international conference on machine learning*. PMLR, 2017, pp. 3027–3036.

[42] M. Doostmohammadian, A. Aghasi, A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, K. H. Johansson, and T. Charalambous, "Fast-convergent anytime-feasible dynamics for distributed allocation of resources over switching sparse networks with quantized communication links," in *European Control Conference*. IEEE, 2022, pp. 84–89.

[43] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[44] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[45] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin, "Exponential graph is provably efficient for decentralized deep training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13975–13987, 2021.

[46] Y. Takezawa, R. Sato, H. Bao, K. Niwa, and M. Yamada, "Beyond exponential graph: Communication-efficient topologies for decentralized learning via finite-time convergence," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[47] M. Doostmohammadian, S. Kharazmi, and H. R. Rabiee, "How clustering affects the convergence of decentralized optimization over networks: a monte-carlo-based approach," *Social Network Analysis and Mining*, vol. 14, no. 1, pp. 135, 2024.

[48] A. Bandyopadhyay and S. Kar, "Impact of network structure on synchronization of hindmarsh–rose neurons coupled in structured network," *Applied Mathematics and Computation*, vol. 333, pp. 194–212, 2018.

[49] D. Lusher, J. Koskinen, and G. Robins, *Exponential random graph models for social networks: Theory, methods, and applications*, Cambridge University Press, 2013.

[50] A. Tahbaz-Salehi and A. Jadbabaie, "On consensus over random networks," in *44th Annual Allerton Conference on Communication, Control, and Computing*. University of Illinois at Urbana-Champaign, Coordinated Science Laboratory , 2006, pp. 1315–1321.

[51] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 102–133, 2009.

[52] N. H. Vaidya, C. N. Hadjicostis, and A. D. Dominguez-Garcia, "Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity," in *51st IEEE Conference on Decision and Control*, 2012, pp. 2761–2766.

[53] A. Datar, D. Schneider, F. Mirali, H. Werner, and H. Frey, "A memory weighted protocol for sampled-data systems subjected to packet dropouts," in *American Control Conference (ACC)*. IEEE, 2018, pp. 2485–2490.

[54] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *IEEE Proceedings*, vol. 95, no. 1, pp. 215–233, 2007.

[55] M. Doostmohammadian, A. Aghasi, M. Pirani, E. Nekouei, U. A. Khan, and T. Charalambous, "Fast-convergent anytime-feasible dynamics for distributed allocation of resources over switching sparse networks with quantized communication links," in *European control conference (ECC)*. IEEE, 2022, pp. 84–89.

[56] S. Oh, H. Sim, J. Kim, and J. Lee, "Non-uniform step size quantization for accurate post-training quantization," in *European Conference on Computer Vision*. Springer, 2022, pp. 658–673.

[57] J. Xu, J. Fan, B. Nan, C. Ding, L. Zheng, Z. Zou, and Y. Huan, "Aslog: An area-efficient cnn accelerator for per-channel logarithmic post-training quantization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[58] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.

[59] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[60] J. E. Slotine and W. Li, *Applied nonlinear control*, Prentice hall Englewood Cliffs, NJ, 1991.

[61] M. Doostmohammadian, M. Pirani, U. A. Khan, and T. Charalambous, "Consensus-based distributed estimation in the presence of heterogeneous, time-invariant delays," *IEEE Control Systems Letters*, vol. 6, pp. 1598 – 1603, 2021.

[62] G. W. Stewart and J. Sun, *Matrix perturbation theory*, Academic Press, 1990.

[63] D. Jakovetic, D. Bajovic, A. K. Sahu, and S. Kar, "Convergence rates for distributed stochastic optimization over random networks," in *IEEE Conference on Decision and Control*. IEEE, 2018, pp. 4238–4245.

[64] M. Doostmohammadian, A. Aghasi, A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. Hadjicostis, K. H. Johansson, and T. Charalambous, "Distributed CPU scheduling subject to nonlinear constraints," in *IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2022, pp. 746–751.

[65] F. Saadatniaki, R. Xin, and U. A. Khan, "Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4769–4780, 2020.

[66] L. Yao, Y. Yuan, S. Sundaram, and T. Yang, "Distributed finite-time optimization," in *IEEE 14th International Conference on Control and Automation (ICCA)*. IEEE, 2018, pp. 147–154.

[67] B. Ning, Q. Han, and Z. Zuo, "Distributed optimization for multi-agent systems: An edge-based fixed-time consensus approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 122–132, 2019.

Mohammadreza Doostmohammadian received his B.Sc. and M.Sc. in Mechanical Engineering from Sharif University of Technology, Iran, respectively in 2007 and 2010, where he worked on applications of control systems and robotics. He received his PhD in Electrical and Computer Engineering from Tufts University, MA, USA in 2015. During his PhD in Signal Processing and Robotic Networks (SPARTN) lab, he worked on control and signal processing over networks with applications in social networks. From 2015 to 2017 he was a postdoc researcher at ICT Innovation Center for Advanced Information and Communication Technology (AICT), School of Computer Engineering, Sharif University of Technology, with research on network epidemic, distributed algorithms, and complexity analysis of distributed estimation methods. He was a researcher at Iran Telecommunication Research Center (ITRC), Tehran, Iran in 2017 working on distributed control algorithms and estimation over IoT. Since 2017 he has been an Assistant Professor with the Mechatronics Department at Semnan University, Iran. From 2021 to 2022, he was a visiting researcher at the School of Electrical Engineering and Automation, Aalto University, Espoo, Finland, working on constrained and unconstrained distributed optimization techniques and their applications. His general research interests include distributed estimation, control, learning, and optimization over networks. He was the chair of the robotics and control session at the ISME-2018 conference, the session chair at the 1st Artificial Intelligent Systems Conference of Iran, 2022, and the Associate Editor of IEEE/IFAC CoDIT2024 conference.

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3526967

12

Muhammad I. Qureshi received the B.S. degree in electrical engineering from the University of Engineering and Technology Lahore, Pakistan, in 2017, and the Ph.D. degree in electrical and computer engineering from Tufts University, USA, in 2024. He was working as a Research Scientist at Tufts University under the supervision of Prof. Usman A. Khan. Currently, he is a researcher at Intel corporation.

Mohammad Hossein Khalesi received his B.Sc. degree in Mechanical Engineering and also in Aerospace Engineering, M.Sc. degree in Mechanical Engineering, and Ph.D. in Mechanical Engineering all from Sharif University of Technology, Tehran, Iran, respectively in 2011, 2013, and 2019. Currently, he is an assistant professor of Mechanical Engineering at Semnan University, Semnan, Iran.

Hamid R. Rabiee received his BS and MS degrees (with Great Distinction) in Electrical Engineering from CSULB, Long Beach, CA (1987, 1989), his EEE degree in Electrical and Computer Engineering from USC, Los Angeles, CA (1993), and his Ph.D. in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, in 1996. From 1993 to 1996 he was a Member of Technical Staff at AT&T Bell Laboratories. From 1996 to 1999 he worked as a Senior Software Engineer at Intel Corporation. He was also with PSU, OGI and OSU universities as an adjunct professor of Electrical and Computer Engineering from 1996-2000. Since September 2000, he has joined Sharif University of Technology, Tehran, Iran. He was also a visiting professor at the Imperial College of London for the 2017-2018 academic year. He is the founder of Sharif University Advanced Information and Communication Technology Research Institute (AICT), ICT Innovation Center, Advanced Technologies Incubator (SATI), Digital Media Laboratory (DML), Mobile Value Added Services Laboratory (VASL), Bioinformatics and Computational Biology Laboratory (BCB) and Cognitive Neuroengineering Research Center. He has also been the founder of many successful High-Tech start-up companies in the field of ICT as an entrepreneur. He is currently a Professor of Computer Engineering at Sharif University of Technology, and Director of AICT, DML, and VASL. He has been the initiator and director of many national and international level projects in the context of Iran National ICT Development Plan and UNDP International Open Source Network (IOSN). He has received numerous awards and honors for his Industrial, scientific and academic contributions. He has acted as chairman in a number of national and international conferences, and holds three patents. He is also a Member of IFIP Working Group 10.3 on Concurrent Systems, and a Senior Member of IEEE. His research interests include statistical machine learning, Bayesian statistics, data analytics and complex networks with applications in social networks, multimedia systems, cloud and IoT privacy, bioinformatics, and brain networks.

Usman A. Khan is a Professor of ECE and Computer Science at Tufts University. He is also an Amazon Scholar with Amazon Robotics. His research interests include artificial intelligence, machine learning, robotics, optimization and control, and stochastic dynamical systems. Recognition of his work includes the prestigious NSF Career award, several federally funded projects and NSF REU awards, an IEEE journal cover. He received the 2023 Tufts Inventor award, the 2022 EURASIP Best Paper Award for articles published in the *EURASIP Journal on Advances in Signal Processing* over 2017-2021, and Best Student Paper awards at the *IEEE International Conference on Networking, Sensing and Control* (2014), and at the *IEEE Asilomar Conference* (2014 and 2016). Dr. Khan received his B.S. degree in 2002 from University of Engineering and Technology, Pakistan, M.S. degree in 2004 from University of Wisconsin, and Ph.D. degree in 2009 from CMU, all in ECE. He was a postdoc in the GRASP lab at the U-Penn and also has held a Visiting position at KTH, Sweden. Dr. Khan is an *IEEE Senior Member* and was an elected full member of the *Sensor Array and Multichannel TC* with the *IEEE Signal Processing Society* from 2019-2022, where he was an Associate member from 2010 to 2019. He was an elected full member of the *IEEE Big Data Special Interest Group* from 2017 to 2019, and has served on the *IEEE Young Professionals Committee* and on *the IEEE Technical Activities Board*. He has served as an Associate Editor on several top-tier IEEE publications: *IEEE Transactions on Smart Grid* (2014-2017); *IEEE Control System Letters* (2018-2020), *IEEE Transactions on Signal and Information Processing over Networks* (2019-current), where he received the 2023 Outstanding Editorial Member award; *IEEE Open Journal of Signal Processing* (2019-current); and *IEEE Transactions on Signal Processing* (2021-2023), where currently he is a Senior Area Editor. He served as the Chief Editor for the *Proceedings of the IEEE* special issue on *Optimization for Data-driven Learning and Control* (Nov. 2020), and as a Guest Associate Editor for the *IEEE Control System Letters* special issue on *Learning and Control* (Nov. 2020). He served as the Technical Area Chair for the Networks track in *2020 IEEE Asilomar Conference on Signals Systems and Computers* and for the *Signal Processing for Self Aware and Social Autonomous Systems* track at the *1st IEEE International Conference on Autonomous Systems* (Aug. 2021).