Communication-Censored Linearized ADMM for Decentralized Consensus Optimization

Weiyu Li, Yaohua Liu, Zhi Tian, and Qing Ling

***Abstract*—In this paper, we propose a communication- and computation-efficient algorithm to solve a convex consensus optimization problem defined over a decentralized network. A remarkable existing algorithm to solve this problem is the alternating direction method of multipliers (ADMM), in which at every iteration every node updates its local variable through combining neighboring variables and solving an optimization subproblem. The proposed algorithm, called as communication-censored linearized ADMM (COLA), leverages a linearization technique to reduce the iteration-wise computation cost of ADMM and uses a communication-censoring strategy to alleviate the communication cost. To be specific, COLA introduces successive linearization approximations to the local cost functions such that the resultant computation is first-order and light-weight. Since the linearization technique slows down the convergence speed, COLA further adopts the communication-censoring strategy to avoid transmissions of less informative messages. A node is allowed to transmit only if the distance between the current local variable and its previously transmitted one is larger than a censoring threshold. COLA is proven to be convergent when the local cost functions have Lipschitz continuous gradients and the censoring threshold is summable. When the local cost functions are further strongly convex, we establish the linear (sublinear) convergence rate of COLA, given that the censoring threshold linearly (sublinearly) decays to 0. Numerical experiments corroborate with the theoretical findings and demonstrate the satisfactory communication-computation tradeoff of COLA.**

***Index Terms*—Decentralized network, consensus optimization, communication-censoring strategy, linearized approximation, alternating direction method of multipliers.**

Manuscript received November 23, 2018; revised July 16, 2019; accepted October 15, 2019. Date of publication December 6, 2019; date of current version December 24, 2019. The work of Z. Tian was supported by the US NSF under Grants 1527396 and 1741338. The work of Q. Ling was supported by the China NSF under Grants 61573331 and 61973324. This paper was presented in part at the 44th IEEE International Conference on Acoustics, Speech, and Signal Processing, Brighton, U.K., May 12–17, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gesualdo Scutari. *(Corresponding author: Qing Ling.)*

W. Li is with the School of Gifted Young, University of Science and Technology of China, Hefei 230026, China (e-mail: liweiyu@mail.ustc.edu.cn).

Y. Liu is with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: vcifer@mail.ustc.edu.cn).

Z. Tian is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: ztian1@gmu.edu).

Q. Ling is with the School of Data and Computer Science and the Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: lingqing556@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TSIPN.2019.2957719

I. Introduction

In this paper, we consider solving a convex consensus optimization problem

$$\tilde{x}^* = \arg\min_{\tilde{x}} \sum_{i=1}^{n} f_i(\tilde{x}), \tag{1}$$

which is defined over a bidirectionally connected decentralized network consisting of n nodes. All the nodes cooperate to find an optimal argument $\tilde{x}^*$ of the common optimization variable $\tilde{x} \in \mathcal{R}^p$, but the convex local cost function $f_i(\tilde{x}) : \mathcal{R}^p \to \mathcal{R}$ held by every node i is kept private. We focus on the scenario that the nodes are unable to afford complicated computation, while the communication resources are also limited. Our goal is to devise a communication-efficient decentralized algorithm, which relies on light-weight computation, to solve (1).

Decentralized consensus optimization has attracted extensive interest in recent years. Problems in the form of (1) are involved in a variety of research areas, including wireless sensor networks [1]–[3], communication networks [4], [5], multi-robot networks [6], [7], smart grids [8]–[10], machine learning systems [11]–[13], to name a few. Popular algorithms to solve (1) span from the primal domain to the dual domain. The primal domain algorithms, such as sub-gradient descent [14]–[16], dual averaging [17]–[19] and network Newton [20], have to use diminishing step sizes to guarantee exact convergence to an optimal solution, and thus suffer from slow convergence. On the other hand, (1) can be reformulated as a constrained optimization problem and solved by dual domain algorithms, among which the celebrated alternating direction method of multipliers (ADMM) is able to achieve fast and exact convergence [2], [21]–[23]. When ADMM is implemented in a synchronous manner, at every iteration, every node solves an optimization subproblem dependent on its local cost function, and then exchanges the calculated local variable with its neighbors. Therefore, if the local cost functions are not in simple forms, solving the subproblems is computationally demanding. To alleviate the computation cost, the decentralized linearized ADMM (DLM) replaces the local cost functions in ADMM by their linear approximations, and attains a dual domain method with light-weight computation [24], [25]. Similar techniques have also been applied to develop other first-order dual domain algorithms, such as EXTRA [26], NEXT [27], and gradient tracking methods [28]–[32]. If computing the inverse of a Hessian matrix is affordable at a node, one can replace the local

cost functions by their quadratic approximations. The resultant second-order algorithms, DQM and ESOM, have faster convergence than their first-order counterparts [33], [34]. Between the first- and second-order algorithms, a recent work in [35] develops a primal-dual quasi-Newton method that approximates the second-order information with local gradients. The lower complexity bounds and rate-optimal algorithms of decentralized optimization are developed in [36]–[38]. Note that the communication cost in the aforementioned algorithms is proportional to the number of iterations, since after a given number of iterations every node needs to communicate with its neighbors.

In all decentralized algorithms, there is an essential communication-computation tradeoff [39]–[43]. An algorithm with light-weight iteration-wise computation generally needs more number of iterations, and in consequence more communication cost, to reach a target accuracy. For example, compared with ADMM, DLM enjoys simple gradient-based computation, but suffers from relatively slow convergence speed and high communication cost. In this paper, we aim at achieving a favorable communication-computation tradeoff in a decentralized network, where the nodes are only affordable to light-weight gradient-based computation. The limitation on the computation power may come from that the nodes are equipped with cheap computing units in a wireless sensor network, or from that using higher-order information is prohibitively time-consuming for finding a high-dimensional solution in a machine learning system.

Given the constraint on the computation cost, we adopt the communication-censoring strategy to further save the communication cost. The basic idea of the communication-censoring strategy is to only allow transmissions of informative messages over the network. A simple yet powerful protocol is to prevent a node from transmitting a variable that is close to its previously transmitted one, where the "closeness" is determined by comparing the Euclidean distance with a predefined time-varying censoring threshold. The communication-censoring strategy is tightly related to event-triggered control of continuous-time networks [44]–[46], and finds successful applications in discrete-time decentralized optimization [47]–[50]. It has been combined with primal domain methods such as sub-gradient descent [47] and dual averaging [48], as well as dual domain methods such as dual decomposition [49] and ADMM [50]. However, similar to their uncensored counterparts, the primal domain methods in [47], [48] have to use diminishing step sizes to guarantee exact convergence. On the other hand, the dual domain methods in [49], [50] require the nodes to solve computationally demanding subproblems. Our proposed algorithm, called as communication-censored linearized ADMM (COLA), combines the communication-censoring strategy with the first-order dual domain method DLM. Particularly, we modify the standard communication-censoring strategy in [47]–[50] to fit for the special algorithmic structure of DLM so as to attain better performance. We rigorously establish convergence as well as sublinear and linear convergence rates of COLA. To the best of our knowledge, COLA is the first communication-censored method that only uses gradient information but achieves linear convergence.

Starting from the derivation of the classical ADMM in Section II-A, we introduce COLA in Section II-B. COLA modifies ADMM in two aspects. First, linearizing the local cost functions enables approximately solving the time-consuming subproblems in ADMM, and thus saves computation. Second, the communication-censoring strategy is applied to remedy the poor communication efficiency caused by the linearization step. To further demonstrate the design principles of COLA, its tradeoff between communication and computation is discussed and compared with those of several existing dual domain algorithms in Section II-C. In Section III-B, we prove that when the censoring threshold is properly chosen, COLA converges to an optimal solution of (1) (Theorem 1). Moreover, when the local cost functions are strongly convex, the linear and sublinear convergence rates of COLA are established (Theorems 2 and 3). The analysis provides guidelines for choosing the parameters of COLA to reduce computation and communication costs. Section IV presents numerical experiments and demonstrates the communication-computation tradeoff of COLA. Section V summarizes our work.

A. Notations

For matrices $A \in \mathcal{R}^{a\times n}$ and $B \in \mathcal{R}^{b\times n}$, $[A; B] \in \mathcal{R}^{(a+b)\times n}$ stacks the two matrices by rows. Define the inner product of two vectors v_1 and v_2 as $\langle v_1, v_2\rangle := v_1^T v_2$, which naturally induces the Euclidean norm $\|v\| := \sqrt{\langle v, v\rangle}$ of a vector v. For a matrix M, define $\lambda_{\min}(M)$ as the smallest eigenvalue, $\sigma_{\max}(M)$ as the largest singular value, and $\tilde{\sigma}_{\min}(M)$ as the smallest nonzero singular value. When M is a block matrix, $(M)_{i,j}$ denotes its (i, j)-th block.

Throughout the paper, we consider a bidirectionally connected network $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of n nodes and $\mathcal{A} = \{1, \ldots, m\}$ is the set of m directed arcs. Nodes i and j are called as neighbors if $(i, j) \in \mathcal{A}$ and $(j, i) \in \mathcal{A}$. We denote the set of node i's neighbors as $\mathcal{N}_i$ with cardinality $d_{ii} = |\mathcal{N}_i|$. Further define the extended block arc source matrix $A_s \in \mathcal{R}^{mp\times np}$ containing $m \times n$ square blocks $(A_s)_{e,i} \in \mathcal{R}^{p\times p}$. The block $(A_s)_{e,i} = I_p$ if the arc $e = (i, j) \in \mathcal{A}$ and is null otherwise, where I_p is the p-dimensional identity matrix. Likewise, define the extended block arc destination matrix $A_d \in \mathcal{R}^{mp\times np}$, whose block $(A_d)_{e,j} \in \mathcal{R}^{p\times p}$ is not null but I_p if and only if the arc $e = (i, j) \in \mathcal{A}$ terminates at node j. Then, define the extended oriented incidence matrix as $G_o = A_s - A_d$ and the unoriented one as $G_u = A_s + A_d$. The oriented Laplacian is written as $L_o = \frac{1}{2}G_o^T G_o$ and the unoriented Laplacian $L_u = \frac{1}{2}G_u^T G_u$. The degree matrix is defined as $D = \frac{1}{2}(L_o + L_u)$, which is block diagonal with diagonal blocks $D_{i,i} = d_{ii} I_p$.

II. Algorithm Development

In this section, we propose COLA, the communication-censored linearized ADMM to solve the decentralized consensus optimization problem (1). Rooted on ADMM, COLA features

in two ingredients, *linearization* to reduce the computation cost and *communication censoring* to reduce the communication cost. We shall first introduce the development of ADMM in Section II-A, and then combine the linearization and communication-censoring techniques to devise COLA in Section II-B. The tradeoff between computation and communication is discussed in Section II-C.

A. ADMM: Alternating Direction Method of Multipliers

ADMM is a powerful tool to solve a structured optimization problem with two blocks of variables, which are separable in the cost function and subject to a linear equality constraint. To rewrite (1) into the standard bivariate form, we introduce local variables $x_i \in \mathcal{R}^p$ as copies of $\tilde{x}$ at nodes i, and auxiliary variables $z_{ij} \in \mathcal{R}^p$ at arcs $(i,j) \in \mathcal{A}$. Since the network is connected, (1) is equivalent to

$$\begin{aligned} \min_{\{x_i\},\{z_{ij}\}} \quad & \sum_{i=1}^{n} f_i(x_i), \\ \text{s.t.} \quad & x_i = z_{ij},\ x_j = z_{ij},\ \forall (i,j) \in \mathcal{A}. \end{aligned} \tag{2}$$

An optimal solution of (2) satisfies $x_i^* = \tilde{x}^*$ and $z_{ij}^* = \tilde{x}^*$, where $\tilde{x}^*$ is an optimal solution of (1).

Concatenate the variables as $x = [x_1; \ldots; x_n] \in \mathcal{R}^{np}$ and $z = [z_1; \ldots; z_m] \in \mathcal{R}^{mp}$, introduce the aggregate function $f(x) := \sum_{i=1}^n f_i(x_i)$, and denote $A := [A_s; A_d] \in \mathcal{R}^{2mp \times np}$ and $B := [-I_{mp}; -I_{mp}]$. The matrix form of (2) is

$$\min_{x,z}\ f(x), \text{s.t. } Ax + Bz = 0, \tag{3}$$

which is the standard bivariate form handled by ADMM, except that the variable z is absent in the cost function.

Introduce the augmented Lagrangian of (3) as

$$L(x,z,\lambda) = f(x) + \langle \lambda, Ax + Bz \rangle + \frac{c}{2}\|Ax + Bz\|^2,$$

where the penalty parameter $c > 0$ is an arbitrary positive constant and the Lagrange multiplier $\lambda := [\phi; \psi] \in \mathcal{R}^{2mp}$. The two vectors ϕ, $\psi \in \mathcal{R}^{mp}$ are the Lagrangian multipliers associated with the two constraints $A_s x - z = 0$ and $A_d x - z = 0$ respectively. At time k, the ADMM update follows

$$\begin{aligned} x^{k+1} &= \arg\min_x L(x, z^k, \lambda^k), \\ z^{k+1} &= \arg\min_z L(x^{k+1}, z, \lambda^k), \\ \lambda^{k+1} &= \lambda^k + c(Ax^{k+1} + Bz^{k+1}). \end{aligned}$$

According to [23], if the variables are initialized with $\phi^0 = -\psi^0$ and $G_u x^0 = 2z^0$, then we can eliminate z^{k+1} and replace λ^{k+1} by a lower-dimensional dual variable, such that the update is reduced to

$$x^{k+1} = \arg\min_x\ f(x) + \langle \mu^k - cL_u x^k, x \rangle + cx^T D x, \tag{4}$$

$$\mu^{k+1} = \mu^k + cL_o x^{k+1}, \tag{5}$$

where $\mu^k := G_o^T \phi^k \in \mathcal{R}^{np}$. By splitting $\mu^k = [\mu_1^k, \ldots, \mu_n^k]$, $\mu_i^k \in \mathcal{R}^p$ denotes the local dual variable of node i.

Using the definitions of $f(x)$, D, L_u and L_o, we describe how the decentralized ADMM is implemented. At time k, every node i updates its local primal variable x_i^{k+1} using its x_i^k and μ_i^k, as well as x_j^k from all neighbors j via

$$x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \left\langle \mu_i^k - c\sum_{j \in \mathcal{N}_i}(x_i^k + x_j^k), x_i \right\rangle + cd_{ii}x_i^2. \tag{6}$$

Then node i broadcasts its x_i^{k+1} to all neighbors. Finally, node i updates its local dual variable μ_i^{k+1} using its x_i^{k+1} and μ_i^k, as well as x_j^{k+1} from all neighbors j via

$$\mu_i^{k+1} = \mu_i^k + c\sum_{j \in \mathcal{N}_i}(x_i^{k+1} - x_j^{k+1}). \tag{7}$$

The costs of implementing ADMM are two-fold. The first is in computing the local primal and dual variables x_i^k and μ_i^k, in which the update of x_i^k in (6) is particularly demanding when the local cost function $f_i(x_i)$ is complicated. The second is in transmitting the local primal variables x_i^{k+1}, which is expensive when the bandwidth resource is limited.

B. COLA: Communication-Censored Linearized ADMM

COLA adopts two strategies to improve the computation and communication efficiency of ADMM: linearization and communication censoring. The linearization technique has been used in [24], [25] to devise DLM, a gradient-based variant of ADMM. DLM effectively reduces the computation cost of solving subproblems in ADMM, but sacrifices on the convergence speed and thus results in high communication cost. Therefore, we use the communication-censoring strategy to prevent transmissions of less informative messages. Note that though the communication-censoring strategy has been applied to improve the communication efficiency of sub-gradient descent, dual averaging, dual decomposition and ADMM [47]–[50], we customize it in COLA so as to achieve a satisfactory balance between communication and computation, as we shall explain below.

Linearization: Notice that the update of the primal variable x_i^{k+1} in (6), which usually has no explicit solution, dominates the computation cost of ADMM. Therefore, a computationally demanding inner loop should be used to solve x_i^{k+1}. To address this issue, [24], [25] linearizes the local cost functions at every iteration. To be specific, at time k, the function $f_i(x_i)$ in (6) is replaced by its quadratic approximation $f_i(x_i^k) + \langle \nabla f_i(x_i^k), x_i - x_i^k \rangle + \frac{\rho}{2}\|x_i - x_i^k\|^2$ at $x_i = x_i^k$, where $\rho > 0$ is a positive linearization parameter. Therefore, the primal variable is updated via

$$x_i^{k+1} = x_i^k - \frac{1}{2cd_{ii} + \rho}\left(\nabla f_i(x_i^k) + c\sum_{j \in \mathcal{N}_i}(x_i^k - x_j^k) + \mu_i^k\right). \tag{8}$$

Note that the main computation cost of (8) is in calculating the gradient $\nabla f_i(x_i^k)$, which is light-weight. The update of dual variable remains the same as (7) in ADMM.

Communication censoring: The linearization technique significantly reduces the computation cost of ADMM, but slows

Algorithm 1: COLA Run by Node i.

Require: Initialize local variables to $x_i^0 = 0$, $\mu_i^0 = 0$, $\hat{x}_i^0 = 0$ and $\hat{x}_j^0 = 0$ for all $j \in \mathcal{N}_i$.

1: **for** times $k = 0, 1, \cdots$ **do**

2: Compute local primal variable x_i^{k+1} by

$$x_i^{k+1} = x_i^k - \frac{1}{2cd_{ii} + \rho} \left(\nabla f_i(x_i^k) + c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^k - \hat{x}_j^k) + \mu_i^k \right).$$

3: Compute $\xi_i^{k+1} = \|\hat{x}_i^k - x_i^{k+1}\|$.

4: If $\xi_i^{k+1} \geq \tau^{k+1}$, transmit x_i^{k+1} to neighbors and let $\hat{x}_i^{k+1} = x_i^{k+1}$; else do not transmit and let $\hat{x}_i^{k+1} = \hat{x}_i^k$.

5: If receive x_j^{k+1} from any neighbor j, let $\hat{x}_j^{k+1} = x_j^{k+1}$; else let $\hat{x}_j^{k+1} = \hat{x}_j^k$.

6: Update local dual variable μ_i^{k+1} as

$$\mu_i^{k+1} = \mu_i^k + c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^{k+1} - \hat{x}_j^{k+1}).$$

7: **end for**

down the convergence speed, and hence results in high communication cost. Hence, we introduce the communication-censoring strategy to further reduce the communication cost. Intuitively, when x_i^{k+1} is close to x_i^k, it is not necessary for node i to transmit both of them to neighbors. Motivated by this fact, the communication-censoring strategy prevents transmissions of less informative messages so as to reduce the communication cost.

To rigorously explain the communication-censoring strategy, define a state variable $\hat{x}_i^k \in \mathcal{R}^p$ as the latest value that node i has transmitted to neighbors before time k. At time k, after calculating x_i^{k+1}, node i evaluates the difference between $\hat{x}_i^k$ and x_i^{k+1} by their Euclidean distance $\xi_i^{k+1} = \|\hat{x}_i^k - x_i^{k+1}\|$, and then compares the difference with a predefined censoring threshold $\tau^{k+1} \geq 0$. Node i is allowed to transmit x_i^{k+1} to neighbors and update $\hat{x}_i^{k+1} = x_i^{k+1}$, if and only if $\xi_i^{k+1} \geq \tau^{k+1}$. Otherwise, the transmission is censored and $\hat{x}_i^{k+1} = \hat{x}_i^k$. With the state variable $\hat{x}_i^k$, COLA changes the DLM updates in (8) and (7) to

$$x_i^{k+1} = x_i^k - \frac{1}{2cd_{ii} + \rho} \left(\nabla f_i(x_i^k) + c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^k - \hat{x}_j^k) + \mu_i^k \right), \tag{9}$$

$$\mu_i^{k+1} = \mu_i^k + c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^{k+1} - \hat{x}_j^{k+1}). \tag{10}$$

Stacking the state variables in $\hat{x} = [\hat{x}_1; \ldots; \hat{x}_n] \in \mathcal{R}^{np}$, we can write (9) and (10) in the matrix form of

$$x^{k+1} = x^k - (2cD + \rho I)^{-1} \left(\nabla f(x^k) + cL_o \hat{x}^k + \mu^k \right), \tag{11}$$

$$\mu^{k+1} = \mu^k + cL_o \hat{x}^{k+1}. \tag{12}$$

COLA run by node i is outlined in Algorithm 1. At time 0, node i initializes its local variables to $x_i^0 = 0$, $\mu_i^0 = 0$, $\hat{x}_i^0 = 0$ and $\hat{x}_j^0 = 0$ for all $j \in \mathcal{N}_i$. For all times k, node i first computes its local primal variable x_i^{k+1} by (9). The computation of x_i^{k+1} at node i is based on its latest local primal-dual variables x_i^k and μ_i^k, the latest broadcast information $\hat{x}_i^k$ of itself and $\hat{x}_j^k$ from its neighbors j, as well as the gradient of the local cost function $f_i(x_i)$ at $x_i = x_i^k$. Then ξ_i^k, the difference between the newly computed primal variable x_i^{k+1} and the previously transmitted $\hat{x}_i^k$ is calculated and denoted by ξ_i^{k+1}. If $\xi_i^{k+1} \geq \tau^{k+1}$, meaning that the difference exceeds the threshold to communicate, node i transmits x_i^{k+1} to neighbors and lets $\hat{x}_i^{k+1} = x_i^{k+1}$. Otherwise, node i does not transmit and lets $\hat{x}_i^{k+1} = \hat{x}_i^k$. On the other hand, if node i receives x_j^{k+1} from any neighbor j, then it lets $\hat{x}_j^{k+1} = x_j^{k+1}$. Otherwise, it lets $\hat{x}_j^{k+1} = \hat{x}_j^k$. Observe that this communication protocol guarantees that node i and its neighbors store the same state variable $\hat{x}_i^{k+1}$. Finally, the local dual variable μ_i^{k+1} is updated by (10).

Remark 1: Comparing (8) and (7) with (9) and (10), we observe that the only difference between DLM and COLA is replacing $c \sum_{j \in \mathcal{N}_i} (x_i^k - x_j^k)$ by $c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^k - \hat{x}_j^k)$ in the primal-dual updates. This is not the standard strategy used in the other communication-censored algorithms [47]–[50], where all the local primal variables x_i are replaced by the state variables $\hat{x}_i$. We customize the communication censoring strategy for COLA and keep the local primal variable x_i^k in $x_i^k - \frac{1}{2cd_{ii}+\rho} \nabla f_i(x_i^k)$ as it is, because x_i^k has already been available for node i, and is more up-to-date than $\hat{x}_i^k$. Recall that the term $x_i^k - \frac{1}{2cd_{ii}+\rho} \nabla f_i(x_i^k)$ comes from the linearization of $f_i(x_i)$. Intuitively, linearization around $x_i = x_i^k$ leads to faster convergence than linearization around $x_i = \hat{x}_i^k$, which has been validated in our preliminary numerical experiments. On the other hand, we do not change the state variables $\hat{x}_i^k$ by the corresponding local primal variables x_i^k in the term $c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^k - \hat{x}_j^k)$ in both primal and dual updates. Note that x_i^k and $\hat{x}_i^k$ are not equal when communication censoring happens and the error between them is determined by the censoring threshold τ^k, while the dual update (10) accumulates all the previous differences between the neighboring state variables $\hat{x}_i^k$ and $\hat{x}_j^k$. Thus, replacing the state variables $\hat{x}_i^k$ therein by the corresponding local primal variables x_i^k shall accumulate the errors, and result in instability or even divergence of the recursion.

The censoring threshold τ^k is a critical factor that influences the communication-computation tradeoff of COLA. Setting a large τ^k prevents less-informative transmissions, and thus reduces the iteration-wise communication cost, though the recursion needs more number of iterations and hence more computation cost to reach a target accuracy. However, a too large τ^k slows down the convergence speed, which in turn increases both the overall computation and communication costs. Since τ^k sets an upper bound for the distance between x_i^k and $\hat{x}_i^k$, a small improvement of the local primal variable x_i^k cannot be accepted to the state variable $\hat{x}_i^k$ and diffused to the network. In this sense, the primal variable cannot converge faster than τ^k. We shall give rigorous analysis on this issue in the theoretical analysis.

If we expect to obtain a linear rate of convergence, a choice for the censoring threshold will be

$$\tau^k = \alpha \cdot (\beta)^k, \tag{13}$$

where $\beta \in (0,1)$ and $\alpha > 0$ are constants. If τ^k is set as $\alpha \cdot (k)^{-r}$ with $r > 1$, a sublinear rate depending on r will be derived. A special case is $\tau^k = 0$ for all times k, meaning that there is no censoring and COLA degenerates to DLM.

C. Tradeoff Between Communication and Computation

Here we discuss the communication-computation tradeoff in ADMM, DLM, as well as their communication-censored versions, COCA and COLA.

Generally speaking, among the four algorithms, ADMM needs the least number of iterations to reach a target accuracy, but the computation cost of solving subproblems is often remarkable. DLM alleviates the iteration-wise computation cost through linearization, but requires more number of iterations and higher overall communication cost than ADMM.

The communication-censoring strategy in COCA and COLA adjusts the communication-computation tradeoff through tuning the censoring threshold τ^k. As we have discussed in Section II-B, a larger τ^k leads to more iterations and thus higher computation cost, but lower iteration-wise communication cost. Regarding the overall communication cost required to reach a target accuracy, there is a phase transition in tuning τ^k. When τ^k is too large, communication censoring is too often and much more number of iterations is necessary to compensate the information loss, which would deteriorate the overall communication cost.

Though COCA and COLA both adopt the communication-censoring strategy, their application scenarios are different. COCA fits for applications where computation of solving complicated subproblems is not an issue, but communication is the main bottleneck. Examples include distributed resource allocation in a data center network and collaborative target tracking in a radar network. On the contrary, COLA inherits the advantage of light-weight computation from DLM, and further reduces the communication cost on top of it. In this sense, COLA fits for applications where nodes are unable to afford solving complicated subproblems due to hardware or time constraints, such as an IoT network equipped with cheap computation units and a drone network cruising in a fast changing environment.

An illustration of the tradeoff between computation efficiency and communication efficiency in ADMM, DLM, COCA and COLA is given by Fig. 1.

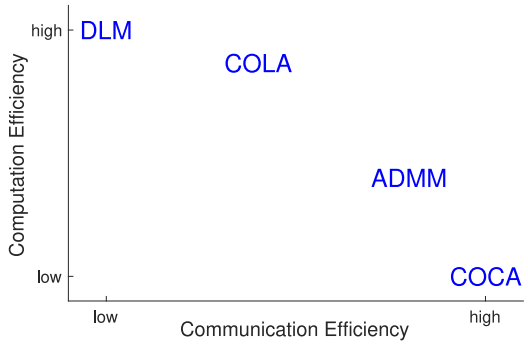

Fig. 1. Illustration of the tradeoff between computation efficiency and communication efficiency in ADMM, DLM, COCA, and COLA.

III. Convergence and Rates of Convergence

In this section, we prove that COLA converges to an optimal solution of the convex consensus optimization problem (1) under mild conditions. Further, if the local cost functions are strongly convex, COLA converges to the unique optimal solution of (1) at a linear or sublinear rate, depending on the choice of the censoring threshold. Section III-A provides assumptions and lemmas for the proofs. Section III-B analyzes the convergence of COLA, while linear and sublinear rates are established in Section III-C.

A. Preliminaries

We make the following assumptions for the analysis. Assumptions 1–4 are sufficient for proving the convergence of COLA to an optimal solution of (1). Further with Assumption 5, COLA is guaranteed to converge to the unique optimal solution of (1) at a linear (sublinear) rate when the censoring threshold is linearly (sublinearly) decaying to 0.

Assumption 1 (Network connectivity): The communication graph $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ is bidirectionally connected.

Assumption 2 (Convexity and differentiability): The local cost functions f_i are convex and differentiable.

Assumption 3 (Lipschitz continuous gradients): The gradients of the local cost functions ∇f_i are Lipschitz continuous with constant $M > 0$. That is, given any $\tilde{x}, \tilde{y} \in \mathcal{R}^p$, $\|\nabla f_i(\tilde{x}) - \nabla f_i(\tilde{y})\| \leq M\|\tilde{x} - \tilde{y}\|$ for any i.

Assumption 4 (Initialization): The dual variable μ of COLA is initialized in the column space of G_o^T. That is, there exists a vector $\phi^0 \in \mathcal{R}^{mp}$ such that $\mu^0 = G_o^T \phi^0$.

Assumption 5 (Strong convexity): The local cost functions f_i are strongly convex with constant $m > 0$. That is, given any $\tilde{x}, \tilde{y} \in \mathcal{R}^p$, $\langle \nabla f_i(\tilde{x}) - \nabla f_i(\tilde{y}), \tilde{x} - \tilde{y} \rangle \geq m\|\tilde{x} - \tilde{y}\|^2$ for any i.

Assumptions 1, 2, 3 and 5 are standard in analysis of decentralized algorithms. The initial condition in Assumption 4 can be easily satisfied, with the simplest choice $\mu^0 = 0$.

COLA involves a primal sequence $\{x^k\}$ and a dual sequence $\{\mu^k\}$. In the theoretical analysis, we shall construct a triple (x^k, z^k, ϕ^k) from the pair (x^k, μ^k), and prove its convergence to (x^*, z^*, ϕ^*), which is optimal to (3). Here $z^k, z^*, \phi^k, \phi^* \in \mathcal{R}^{mp}$. The next lemma gives the properties of (x^*, z^*, ϕ^*).

Lemma 1 (Lemma 1, [24]): Given a primal optimal solution x^* of (3) and $z^* := \frac{1}{2} G_u x^*$, there exist multiple optimal dual variables $[\phi^*; -\phi^*]$ such that every $(x^*, z^*, [\phi^*; -\phi^*])$ is a primal-dual optimal solution of (3). Among all these optimal dual variables, there exists a unique $[\phi^*; -\phi^*]$ in which ϕ^* lies in the column space of G_o. Moreover, any primal-dual optimal solution $(x^*, z^*, [\phi^*; -\phi^*])$ satisfies the KKT conditions

$$\nabla f(x^*) + G_o^T \phi^* = 0, \tag{14}$$

$$G_o x^* = 0, \tag{15}$$

$$\frac{1}{2} G_u x^* = z^*. \tag{16}$$

According to Lemma 1, it is natural to construct $z^k := \frac{1}{2}G_u x^k \in \mathcal{R}^{mp}$. To construct ϕ^k, note that under Assumption 4, μ^0 lies in the column space of G_o^T, and by the definition of $L_o = \frac{1}{2}G_o^T G_o$, every μ^{k+1} in the dual update (12) also lies in the column space of G_o^T. Thus, there exists a vector $\phi^k \in \mathcal{R}^{mp}$ satisfying $\mu^k = G_o^T \phi^k$ for any $k \geq 0$, such that the recursion of COLA can be rewritten as

$$x^{k+1} = x^k - (2cD + \rho I)^{-1}\left(\nabla f(x^k) + cL_o\hat{x}^k + G_o^T\phi^k\right), \tag{17}$$

$$\phi^{k+1} = \phi^k + \frac{c}{2}G_o\hat{x}^{k+1}. \tag{18}$$

Combining (17) and (18) with the KKT conditions (14)–(16), the next lemma gives two equations that are cornerstones of the theoretical analysis. To emphasize the error caused by the communication-censoring strategy, we define an error term $E^k := x^k - \hat{x}^k$ therein.

Lemma 2: Let x^* and ϕ^* be a primal-dual optimal pair of (3), with ϕ^* lying in the column space of G_o. Then, for all $k \geq 0$, the recursion of COLA satisfies

$$\nabla f(x^k) - \nabla f(x^*) = (cL_u + \rho I)(x^k - x^{k+1}) - G_o^T(\phi^{k+1} - \phi^*) + cL_o(E^k - E^{k+1}), \tag{19}$$

$$\frac{c}{2}G_o(x^{k+1} - x^*) = \phi^{k+1} - \phi^k + \frac{c}{2}G_o E^{k+1}. \tag{20}$$

Proof: See Appendix A. ■

The convergence analysis of COLA relies on the following energy function

$$V^k := \frac{\rho}{2}\|x^k - x^*\|^2 + c\|z^k - z^*\|^2 + \frac{1}{c}\|\phi^k - \phi^*\|^2, \tag{21}$$

where the auxiliary variables z^k and ϕ^k as well as their optimal values z^* and ϕ^* are defined above. This energy function also appears in the analysis of DLM, the uncensored version of COLA [24]. However, due to the existence of the communication-censoring strategy which introduces an error term in the recursion, the analysis of COLA is significantly different to that of DLM.

B. Convergence

The convergence of COLA is established as follows.

Theorem 1: Under Assumptions 1–4, in COLA we choose the penalty parameter $c > 0$ and the linearization parameter $\rho > 0$ such that $c\lambda_{\min}(L_u) + \rho > \frac{M}{2}$, and set the censoring threshold $\{\tau^k\}$ as a non-increasing non-negative summable sequence such that $\sum_{k=0}^{\infty}\tau^k < \infty$. Then the primal variable x^k converges to an optimal solution x^* of (3).

Proof: See Appendix B. ■

Theorem 1 asserts that COLA converges to an optimal solution of (1) under mild conditions and provides guidelines for setting parameters. It is interesting to see that the requirement $c\lambda_{\min}(L_u) + \rho > \frac{M}{2}$ is the same as that in DLM [24]. Fixing ρ, a network with better connectedness (namely, larger $\lambda_{\min}(L_u)$) allows us to choose a smaller penalty constant c. Fixing c and $\lambda_{\min}(L_u)$, the linearization parameter ρ must be large enough to guarantee convergence. Note that ρI_p approximates the Hessians of the local cost functions $f_i(x_i)$. A large ρ over-approximates the curvature and forces x_i^{k+1} to be close to x_i^k, which stabilizes the recursion. On the contrary, a small ρ under-approximates the curvature and allows the local variables to change quickly, at the cost of possible divergence. Fig. 2 illustrates the impact of ρ. Regarding the censoring threshold τ^k, we require it to be summable. Intuitively, τ^k determines the maximal error introduced to the primal update. When this error is controllable, the convergence of COLA is guaranteed

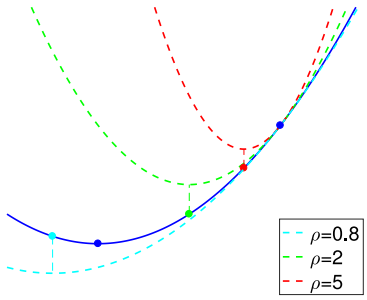

Fig. 2. An illustration of choosing different approximation parameters ρ. In this situation, at the top-right point, we approximate the original cost function (in blue) by the dashed lines (in cyan, green, and red). When $\rho = 2$ and $\rho = 5$ that are both larger than the accurate second derivative, the updates are conservative and go to the green and red points, respectively. When $\rho = 0.8$ that is smaller than the accurate second derivative, the update is aggressive and goes to the cyan point.

C. Rates of Convergence

In Section III-B, we have shown that COLA requires $\{\tau^k\}$ to be summable so as to guarantee convergence. Below, we shall prove that convergence rate of COLA also depends on convergence rate of $\{\tau^k\}$. In addition to Assumptions 1–4, we need the local cost functions to be strongly convex, as stated in Assumption 5. In this circumstance, COLA converges to the unique optimal solution of (1) at a linear (sublinear) rate when $\{\tau^k\}$ is linearly (sublinearly) decaying.

Theorem 2: Under Assumptions 1–5, in COLA we choose the penalty parameter $c > 0$ and the linearization parameter $\rho > \frac{M^2}{2m}$, and set the censoring threshold $\tau^k = \alpha \cdot (\beta)^k$ with $\alpha > 0$ and $\beta \in (0, 1)$. Then there exists a positive constant $\delta > 0$ such that the primal variable x^k converges to the unique optimal solution x^* of (3) at a global linear rate $\mathcal{O}((1+\delta)^{-\frac{k}{2}})$.

Proof: See Appendix C. ■

As shown in (60) in the proof of Theorem 2, the constant δ depends on the algorithm parameters c, ρ and β, the network topology parameterized by $\tilde{\sigma}_{\min}(G_o)$ and $\sigma_{\max}(G_u)$, and the properties of the local cost functions parameterized by M and m. Define the condition numbers of cost functions and graph as $\kappa_f = \frac{M}{m}$ and $\kappa_G = \frac{\sigma_{\max}(G_u)}{\tilde{\sigma}_{\min}(G_o)}$, respectively. The following corollary shows clearer that, by properly setting c and ρ, how the constant δ is determined by κ_f, κ_G and β.

Corollary 1: Under Assumptions 1–5, in COLA we choose $c = \frac{8M}{\sigma_{\max}(G_u)\tilde{\sigma}_{\min}(G_o)}$ and $\rho = M\kappa_f$. Then the global linear rate

$\mathcal{O}((1+\delta)^{-\frac{k}{2}})$ satisfies

$$\delta \leq \min\left\{\frac{1}{8\kappa_G^2}, \frac{1}{2\kappa_f^2+16\kappa_f\kappa_G}, \frac{\kappa_f}{12\kappa_G+6\kappa_f^2\kappa_G}, \frac{1}{\beta^2}-1\right\}. \tag{22}$$

In (22), the terms $\frac{1}{8\kappa_G^2}$, $\frac{1}{2\kappa_f^2+16\kappa_f\kappa_G}$ and $\frac{\kappa_f}{12\kappa_G+6\kappa_f^2\kappa_G}$ are monotonically decreasing when either κ_f or κ_G increases, meaning that the convergence is slower when the cost functions are worse-conditioned and/or the network is less-connected. In addition, δ is bounded by $\frac{1}{\beta^2}-1$. Therefore, (22) shows that among all β that do not affect the convergence rate, the one satisfying $\min\{\frac{1}{8\kappa_G^2}, \frac{1}{2\kappa_f^2+16\kappa_f\kappa_G}, \frac{\kappa_f}{12\kappa_G+6\kappa_f^2\kappa_G}\} = \frac{1}{\beta^2}-1$ achieves largest communication reduction per iteration.

Theorem 3: Under Assumptions 1–5, in COLA we choose the penalty parameter $c>0$ and the linearization parameter $\rho > \frac{M^2}{2m}$, and set the censoring threshold $\tau^k = \alpha\cdot(k)^{-r}$ with $\alpha>0$ and $r>1$. Then there exists a finite time index k_0 such that the distance between the primal variable x^k and the unique optimal solution x^* of (3) is upper-bounded by a sequence decaying sublinearly to 0 at a rate of $\mathcal{O}((k)^{-\frac{q}{2}})$, where $q\in(0,2r-1)$, when $k\geq k_0$.

Proof: See Appendix D. ∎

Theorems 2 and 3 indicate that, to achieve linear (sublinear) convergence, we have to impose stronger requirements on the parameters. The sequence of censoring threshold should be not only summable, but also linearly (sublinearly) decaying. The parameters c and ρ should be larger, too. Note that because $M\geq m$, $\rho > \frac{M^2}{2m} \geq \frac{M}{2}$ and consequently $c\lambda_{\min}(L_u)+\rho > \frac{M}{2}$, which is required in Theorem 1.

According to the upper bound of δ given in (60), the linear rate of x^k reaching x^* (namely, $\mathcal{O}((1+\delta)^{-k/2})$) must be slower than the linear rate of τ^k decaying to 0 (namely, $\mathcal{O}(\beta^k)$). From Theorem 3, one can also see that the sublinear rate of x^k reaching x^* (namely, $\mathcal{O}((k)^{-\frac{q}{2}})$ where $q\in(0,2r-1)$) must be slower than the sublinear rate of τ^k decaying to 0 (namely, $\mathcal{O}((k)^{-r})$). Therefore, in both the linear and the sublinear cases, the sequence of censoring threshold τ^k bounds the convergence rate of x^k to x^*. This makes sense because τ^k means the maximal error allowed to enter the recursion of x^k due to communication censoring.

Remark 2: Though COLA is devised from DLM, the error caused by the communication-censoring strategy makes its analysis different to that of DLM. The analysis of COLA is also different to that of COCA, the censored version of ADMM. The reason is that COLA updates x^k by gradient descent steps, while COCA updates x^k by solving optimization subproblems. This is analogous to the difference in the proofs of DLM and ADMM. In addition to the difference in the proof techniques, we also establish the sublinear convergence of COLA, which is absent in the analysis of DLM and COCA.

Remark 3: When the censoring threshold τ^k is set to 0, COLA degenerates to DLM. Intuitively, the convergence rate of COLA is no faster than that of DLM due to the introduction of the communication-censoring strategy. This is also observed from, for example, the linear convergence constant δ in Corollary 1. Nevertheless, the slower convergence in terms of the number of iterations is acceptable, since COLA effectively reduces the iteration-wise communication cost. We shall demonstrate with numerical experiments that COLA can reduce the overall communication cost comparing to DLM.

IV. Numerical Experiments

This section provides numerical experiments to demonstrate the satisfactory communication-computation tradeoff of COLA. In particular, we shall show that COLA inherits the advantage of cheap computation from its uncensored counterpart DLM [24], [25], but significantly reduces the overall communication cost. Beyond DLM, we compare COLA with the classical ADMM [23] and its censored version COCA [50], both of which do not use the linearization technique and are not computation-efficient. We also compare with the event-triggered sub-gradient descent (ETSD) algorithm [47], which is a primal domain first-order method but much slower than COLA in terms of convergence speed. We consider two decentralized consensus optimization problems, least squares in IV-A and logistic regression in IV-B. The cost functions are both smooth, but the latter is not strongly convex. For ADMM and COCA, subproblems in least squares have explicit solutions, while those in logistic regression needs computationally demanding inner loops. We use the accuracy of the primal variable as the performance metric, defined by $\|x^k-x^*\|^2/\|x^0-x^*\|^2$. Logistic regression may have multiple optimal solutions, among which we choose the one closest to the limit of iterate as x^*. The computation cost is evaluated by time spent to reach a target accuracy, and the communication cost is defined as the accumulated number of broadcast messages. The simulations are carried out on a laptop with an Intel I7 processor and 8 GB memory, programmed with Matlab R2017a in macOS Sierra.

A. Decentralized Least Squares

The local cost function in the decentralized least squares problem is $f_i(\tilde{x}) = \frac{1}{2}\|A_{(i)}\tilde{x}-y_{(i)}\|_2^2$, with $A_{(i)}\in\mathcal{R}^{p\times p}$ and $y_{(i)}\in\mathcal{R}^p$ being private for node i. Thus, the primal update of node i at time k in COLA is

$$x_i^{k+1} = x_i^k - (2cd_{ii}+\rho)^{-1}\Big[A_{(i)}^T(A_{(i)}x_i^k - y_{(i)}) + c\sum_{j\in\mathcal{N}_i}(\hat{x}_i^k-\hat{x}_j^k)+\mu_i^k\Big].$$

Note that node i can compute $(2cd_{ii}+\rho)^{-1}$ in advance to accelerate the computation. In the experiments, entries of $A_{(i)}$ and $b_{(i)}$ are independently and identically sampled from the uniform distribution within [0,1]. Then we let $y_{(i)} = A_{(i)}b_{(i)}$. We set the network size as $n=50$ and the dimension of the local variables as $p=3$.

First, we compare four algorithms, COLA, DLM, COCA and ADMM, over four network topologies: line, random, star and complete, as shown in Figs. 3–6. In the random network, 10% of all possible bidirectional edges are randomly chosen

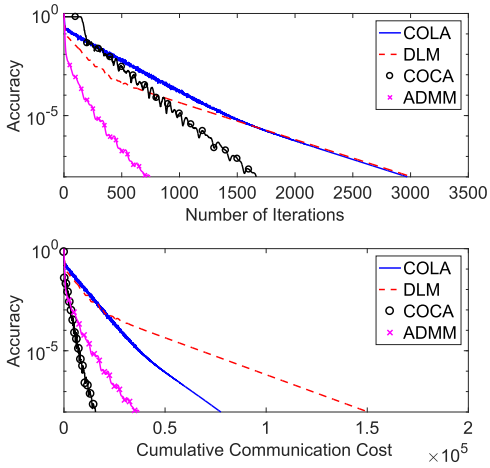

Fig. 3. Performance over line network for decentralized least squares.

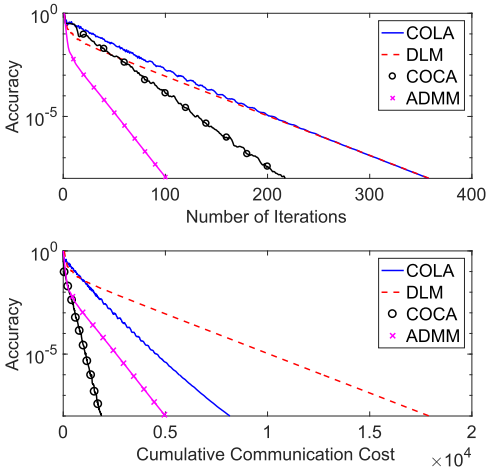

Fig. 4. Performance over random network for decentralized least squares.

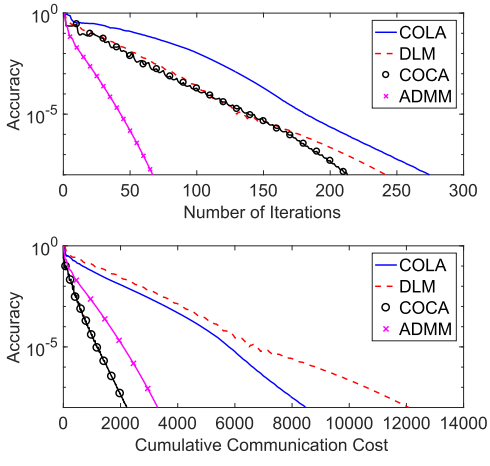

Fig. 5. Performance over star network for decentralized least squares.

to be connected. The accuracies are compared with respect to the number of iterations and the cumulative communication cost. The parameters c and ρ are tuned to be the best for the uncensored algorithms DLM and ADMM, and kept the same in their censored counterparts, respectively. We use the linear

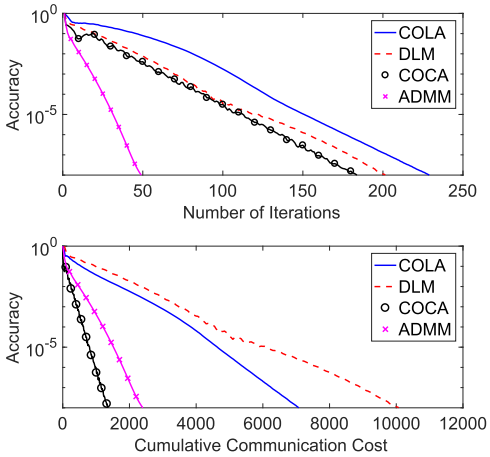

Fig. 6. Performance over complete network for decentralized least squares.

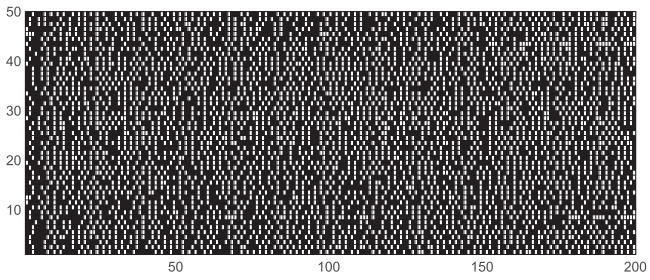

Fig. 7. Censoring pattern of the first 200 iterations of COLA over random network for decentralized least squares. The horizontal axis is the number of iterations, and the vertical axis is the index of node. A dark dot represents that the node is censored at that time.

censoring threshold in the form of $\tau^k = \alpha \cdot (\beta)^k$, where the parameters α and β are hand-tuned in COLA and COCA so as to achieve the best communication efficiency. Taking the random network as an example, we choose $c = 0.45$, $\rho = 1.1$ in DLM and $\alpha = 0.7, \beta = 0.94$ in COLA, while $c = 0.35$ in ADMM and $\alpha = 0.9$, $\beta = 0.92$ in COCA.

In all the networks, the two censored methods COLA and COCA require more iterations to reach the target accuracy than their uncensored counterparts due to the error caused by censoring, but the saving in communication is remarkable. Compared to DLM and given a target accuracy of 10^{-8}, COLA saves $\sim 1/2$ communication costs in the line and random networks, and $\sim 1/3$ in the star and complete networks. The required number of iterations in the line network is much more than those in the other networks, since the connectedness of the line network is the worst. In better connected networks such as star and complete, variable updating is often informative, such that the deterioration of convergence speed caused by skipping transmissions becomes more noticeable, yet communication per iteration is still saved by censoring.

We study the influence of communication censoring over the random network. The censoring pattern of the first 200 iterations is shown in Fig. 7. The horizontal axis is the number of iterations, and the vertical axis is the node index. A white dot means that the node broadcasts at the time, while a black dot means

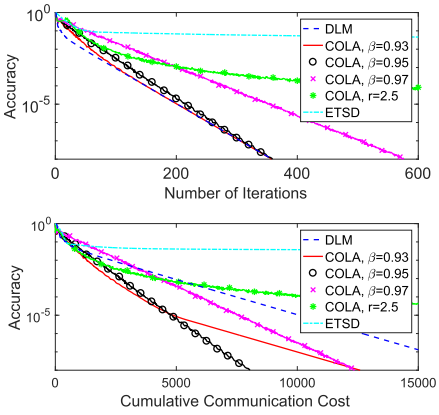

Fig. 8. Performance of DLM, ETSD, and COLA with different censoring thresholds over random network for decentralized least squares. In COLA, with the linear censoring thresholds $\tau^k = \alpha \cdot (\beta)^k$, we fix $\alpha = 0.7$ and choose different β. With the sublinear censoring threshold $\tau^k = \alpha(k)^{-r}$, we choose $\alpha = 1000$ and $r = 2.5$.

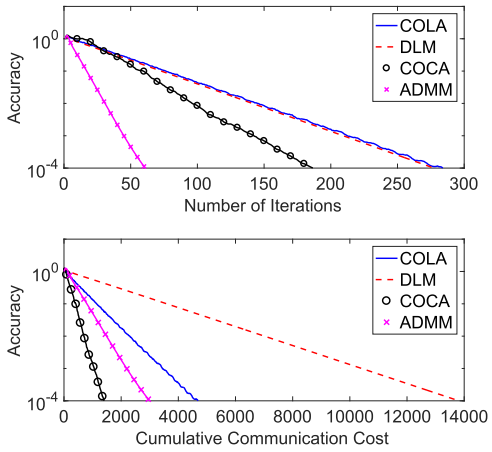

Fig. 9. Performance over random network with 50 nodes for decentralized logistic regression.

that the node is silent. Observe that communication censoring happens uniformly, namely, the frequency of communication censoring does not change too much along the optimization process. In addition, the nodes have similar communication costs eventually. On average, every node broadcasts $0.35 \sim 0.45$ message per time.

Next, we compare the choice of the censoring threshold in COLA over the random network. We compare four censoring thresholds, the linear sequences $\tau^k = \alpha \cdot (\beta)^k$ with $\alpha = 0.7$ while $\beta = 0.93, 0.95$ and 0.97, as well as the sublinear sequence $\tau^k = \alpha \cdot (k)^{-r}$ with $\alpha = 1000$ and $r = 2.5$. The parameters c and ρ remain the same. As shown in Fig. 8, the linear censoring thresholds outperforms the sublinear censoring threshold, in terms of both communication and computation. The reason is that the sublinear rate of the threshold limits the convergence rate of COLA, as we have theoretically analyzed in Section III-B. Regarding the different choices of the linear rate, we observe that a smaller β needs less number of iterations to reach a target accuracy, since it leads to faster decay of the censoring threshold, and thus less communication censoring per iteration. In contrast, with a larger β, we need more number of iterations and less communication cost per iteration. Therefore, a moderate β, such as $\beta = 0.95$ in this case, is preferred.

In Fig. 8, we also compare COLA with ETSD, a communication-censored primal domain first-order method. ETSD adopts the Metropolis-Hastings rule to design its mixing matrix. It uses a linear censoring threshold $\alpha \cdot (\beta)^k$ and a sublinear step size $O((k)^{-\frac{2}{3}})$, where the parameters are all hand-tuned to achieve the best communication efficiency. From Fig. 8, we observe that ETSD requires much more number of iterations and communication cost to reach a target accuracy comparing to COLA. The main reason of the unsatisfactory performance of ETSD is the diminishing step size, which is used to guarantee exact convergence. Similar performance gap can be observed in comparing the uncensored algorithms, sub-gradient descent and DLM.

B. Decentralized Logistic Regression

In the decentralized logistic regression problem, the local cost function of node i is

$$f_i(\tilde{x}) = \frac{1}{l_i} \sum_{l=1}^{l_i} \ln\left(1 + \exp(-y_{(i)l} q_{(i)l}^T \tilde{x})\right),$$

where $q_{(i)l} \in \mathcal{R}^p$ is the lth column of a matrix $Q_{(i)} \in \mathcal{R}^{p \times l_i}$, $y_{(i)l} \in \{-1, +1\}$ is the lth element of a binary vector $y_{(i)} \in \mathcal{R}^{l_i}$, and l_i is the number of samples held by node i. The primal update of node i at time k in COLA is

$$x_i^{k+1} = x_i^k - \frac{1}{2cd_{ii} + \rho}\left[-\frac{1}{l_i} \sum_{l=1}^{l_i} \frac{y_{(i)l} \exp(-y_{(i)l} q_{(i)l}^T x_i^k) q_{(i)l}}{1 + \exp(-y_{(i)l} q_{(i)l}^T x_i^k)} + c \sum_{j \in \mathcal{N}_i} (\hat{x}_i^k - \hat{x}_j^k) + \mu_i^k \right],$$

while the primal updates of ADMM and COCA have no explicit solutions. Therefore, we solve the subproblems therein by a gradient descent inner loop, which terminates when the ℓ_2 norm of the gradient is less than 10^{-8}.

We conduct simulations over two random networks with $n = 50$ and $n = 100$ nodes, in both of which 10% of all possible bidirectional edges are randomly chosen to be connected. The dimension of local variables is $p = 3$. The numbers of samples held by the nodes are i.i.d. and uniformly chosen from integers within $[1, 10]$. Entries of the first two rows of $Q_{(i)}$ follow the i.i.d. discrete uniform distribution on the set $\{0.1w\}$, $w = 1, \ldots, 10$, while entries of the last row are all set as 1. Entries of $y_{(i)}$ are i.i.d. and follow the uniform distribution on $\{-1, 1\}$. As we have done in Section IV-A, c in ADMM is tuned to achieve the fastest convergence, and is also used for COCA; c and ρ in DLM are tuned to achieve the fastest convergence, and are also used for COLA. The censoring threshold in both COCA and COLA is set as $\tau^k = \alpha \cdot (\beta)^k$, with parameters α and β hand-tuned to obtain the best communication efficiency.

As depicted in Figs. 9 and 10, the four algorithms behave similarly to those in the least squares problem (Figs. 3–6). COLA

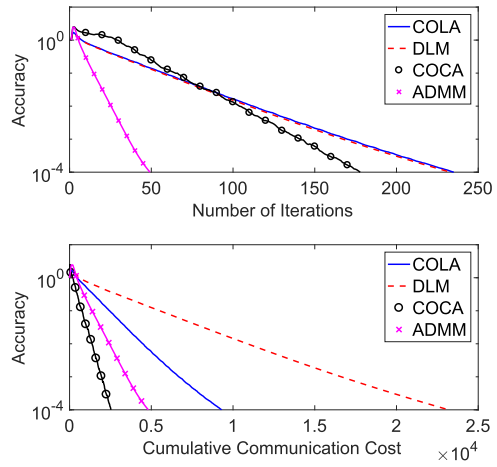

Fig. 10. Performance over random network with 100 nodes for decentralized logistic regression.

TABLE I
THE TIME SPENT OF THE FOUR ALGORITHMS IN TWO NETWORKS WITH DIFFERENT NUMBERS OF NODES n AND TARGET ACCURACIES

n	Accuracy	COLA	DLM	COCA	ADMM
50	10^{-4}	1.076s	0.971s	30.119s	9.629s
50	10^{-5}	1.126s	1.055s	37.198s	12.467s
100	10^{-4}	1.466s	1.320s	37.488s	11.175s
100	10^{-5}	1.879s	1.721s	45.302s	15.797s

saves nearly $2/3$ communication cost with few more iterations compared to DLM. To demonstrate its computation efficiency, we also show the CPU time for the four algorithms to reach target accuracies 10^{-4} and 10^{-5} in Table I. Notice that the two linearized algorithms, COLA and DLM, compute much faster than COCA and ADMM. The time spent by COLA in both networks is slightly more than that of DLM due to the communication censoring operations.

V. CONCLUSION

In this paper, we propose COLA, a communication- and computation-efficient decentralized consensus optimization algorithm. Compared to the classical ADMM, COLA uses the linearization technique to reduce the iteration-wise computation cost, and fits for networks where only light-weight computation is affordable. To compensate the sacrifice in the convergence speed, which is caused by the linearization step and results in low communication efficiency, COLA further introduces the communication-censoring strategy to prevent a node from transmitting its "less-informative" local variable to neighbors. We establish convergence and rates of convergence for COLA, and demonstrate the computation-communication tradeoff with numerical experiments. Our future work is to apply the linearization and communication censoring techniques to decentralized optimization applications in dynamic, online and stochastic environments.

APPENDIX A
PROOF OF LEMMA 2

Proof: From (18), it holds that

$$\begin{aligned}\phi^{k+1}-\phi^k &= \frac{c}{2}G_o\hat{x}^{k+1} \overset{(15)}{=} \frac{c}{2}G_o(\hat{x}^{k+1}-x^*)\\ &= \frac{c}{2}G_o(x^{k+1}-E^{k+1}-x^*),\end{aligned} \tag{23}$$

where the last equality uses the definition $E^{k+1}=x^{k+1}-\hat{x}^{k+1}$. Rearranging terms in (23) yields (19).

Also rearranging terms in (17) to place $\nabla f(x^k)$ at the left side, we have

$$\nabla f(x^k)=(2cD+\rho I)(x^k-x^{k+1})-cL_o\hat{x}^k-G_o^T\phi^k. \tag{24}$$

Subtracting (24) with (14) and noticing the definitions of $D=\frac{1}{2}(L_o+L_u)$ and $L_o=\frac{1}{2}G_o^TG_o$, we have

$$\begin{aligned}&\nabla f(x^k)-\nabla f(x^*)\\ &=(2cD+\rho I)(x^k-x^{k+1})-cL_o\hat{x}^k-G_o^T(\phi^k-\phi^*)\\ &=(cL_u+\rho I)(x^k-x^{k+1})+cL_o(x^k-x^{k+1})-cL_o\hat{x}^k\\ &\quad-G_o^T(\phi^{k+1}-\phi^*)+G_o^T(\phi^{k+1}-\phi^k)\\ &\overset{(23)}{=}(cL_u+\rho I)(x^k-x^{k+1})+cL_o(x^k-x^{k+1})-cL_o\hat{x}^k\\ &\quad-G_o^T(\phi^{k+1}-\phi^*)+cL_o(x^{k+1}-E^{k+1}-x^*)\\ &\overset{(15)}{=}(cL_u+\rho I)(x^k-x^{k+1})-G_o^T(\phi^{k+1}-\phi^*)\\ &\quad+cL_o(E^k-E^{k+1}),\end{aligned}$$

which completes the proof. ■

APPENDIX B
PROOF OF THEOREM 1

Proof: Throughout the proof, we assume $\tau^0>0$. When $\tau^0=0$ such that all $\tau^k=0$ and COLA degenerates to DLM, since the value of τ^0 does not affect the operation of COLA, we can simply set τ^0 as any positive constant.

Step 1: The proof in this step is analogous to the proof of Lemma 3 in [24], but more complicated due to the existence of censoring error. From Assumptions 2 and 3, the gradients of the convex local cost functions ∇f_i are Lipschitz continuous with constant $M>0$. Thus, we have

$$\begin{aligned}&\frac{1}{M}\|\nabla f(x^k)-\nabla f(x^*)\|^2\\ &\le\langle\nabla f(x^k)-\nabla f(x^*),x^k-x^*\rangle\\ &=\langle\nabla f(x^k)-\nabla f(x^*),x^{k+1}-x^*\rangle\\ &\quad+\langle\nabla f(x^k)-\nabla f(x^*),x^k-x^{k+1}\rangle.\end{aligned} \tag{25}$$

For the second term at the right-hand side of (25), we choose an upper bound

$$\begin{aligned}&\langle\nabla f(x^k)-\nabla f(x^*),x^k-x^{k+1}\rangle\\ &\le\frac{1}{M}\|\nabla f(x^k)-\nabla f(x^*)\|^2+\frac{M}{4}\|x^k-x^{k+1}\|^2.\end{aligned} \tag{26}$$

To establish an upper bound for the first term at the right-hand side of (25), we use (19) in Lemma 2 to rewrite it as

$$\begin{aligned}&\langle\nabla f(x^k)-\nabla f(x^*),x^{k+1}-x^*\rangle\\&=\langle(cL_u+\rho I)(x^k-x^{k+1}),x^{k+1}-x^*\rangle\\&\quad-\langle G_o^T(\phi^{k+1}-\phi^*),x^{k+1}-x^*\rangle\\&\quad+\langle cL_o(E^k-E^{k+1}),x^{k+1}-x^*\rangle.\end{aligned}\tag{27}$$

We shall handle the terms at the right-hand side of (27) one by one. The first one satisfies

$$\begin{aligned}&\langle(cL_u+\rho I)(x^k-x^{k+1}),x^{k+1}-x^*\rangle\\&=\frac{c}{2}\langle G_u(x^k-x^{k+1}),G_u(x^{k+1}-x^*)\rangle\\&\quad+\rho\langle x^k-x^{k+1},x^{k+1}-x^*\rangle\\&=2c\langle z^k-z^{k+1},z^{k+1}-z^*\rangle\\&\quad+\rho\langle x^k-x^{k+1},x^{k+1}-x^*\rangle,\end{aligned}\tag{28}$$

which uses the definitions $L_u=\frac{1}{2}G_u^TG_u$, $z^k=\frac{1}{2}G_ux^k$ and $z^*=\frac{1}{2}G_ux^*$. The second one satisfies

$$\begin{aligned}&-\langle G_o^T(\phi^{k+1}-\phi^*),x^{k+1}-x^*\rangle\\&=-\langle\phi^{k+1}-\phi^*,G_o(x^{k+1}-x^*)\rangle\\&\overset{(20)}{=}\frac{2}{c}\langle\phi^{k+1}-\phi^*,\phi^k-\phi^{k+1}\rangle-\langle\phi^{k+1}-\phi^*,G_oE^{k+1}\rangle.\end{aligned}\tag{29}$$

The third one satisfies

$$\begin{aligned}&\langle cL_o(E^k-E^{k+1}),x^{k+1}-x^*\rangle\\&=\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_o(x^{k+1}-x^*)\rangle\\&\overset{(20)}{=}\langle G_o(E^k-E^{k+1}),\phi^{k+1}-\phi^k\rangle\\&\quad+\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_oE^{k+1}\rangle,\end{aligned}\tag{30}$$

which uses the definition $L_o=\frac{1}{2}G_o^TG_o$. Summing up (28), (29) and (30), applying the equality $2\langle v_a-v_b,v_b-v_c\rangle=\|v_a-v_c\|^2-\|v_a-v_b\|^2-\|v_b-v_c\|^2$ that holds for any vectors v_a, v_b and v_c to $\langle z^k-z^{k+1},z^{k+1}-z^*\rangle$, $\langle x^k-x^{k+1},x^{k+1}-x^*\rangle$ and $\langle\phi^{k+1}-\phi^*,\phi^{k+1}-\phi^k\rangle$, and then reorganizing terms, we can rewrite (27) as

$$\begin{aligned}&\langle\nabla f(x^k)-\nabla f(x^*),x^{k+1}-x^*\rangle\\&=c(\|z^k-z^*\|^2-\|z^k-z^{k+1}\|^2-\|z^{k+1}-z^*\|^2)\\&\quad+\frac{\rho}{2}(\|x^k-x^*\|^2-\|x^k-x^{k+1}\|^2-\|x^{k+1}-x^*\|^2)\\&\quad+\frac{1}{c}(\|\phi^k-\phi^*\|^2-\|\phi^k-\phi^{k+1}\|^2-\|\phi^{k+1}-\phi^*\|^2)\\&\quad-\langle\phi^{k+1}-\phi^*,G_oE^{k+1}\rangle+\langle G_o(E^k-E^{k+1}),\phi^{k+1}-\phi^k\rangle\\&\quad+\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_oE^{k+1}\rangle\\&\overset{(21)}{=}V^k-V^{k+1}\\&\quad-c\|z^k-z^{k+1}\|^2-\frac{\rho}{2}\|x^k-x^{k+1}\|^2-\frac{1}{c}\|\phi^k-\phi^{k+1}\|^2\\&\quad-\langle G_oE^{k+1},\phi^k-\phi^*\rangle+\langle G_o(E^k-2E^{k+1}),\phi^{k+1}-\phi^k\rangle\\&\quad+\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_oE^{k+1}\rangle.\end{aligned}\tag{31}$$

For the term $-\langle G_oE^{k+1},\phi^k-\phi^*\rangle$, we observe that

$$\begin{aligned}&-\langle G_oE^{k+1},\phi^k-\phi^*\rangle\\&\le\frac{c_1}{2}\|G_oE^{k+1}\|\|\phi^k-\phi^*\|^2+\frac{1}{2c_1}\|G_oE^{k+1}\|\\&\le\frac{c_1\sigma_{\max}(G_o)\|E^{k+1}\|}{2}\|\phi^k-\phi^*\|^2+\frac{\sigma_{\max}(G_o)}{2c_1}\|E^{k+1}\|,\end{aligned}\tag{32}$$

where $c_1>0$ is any positive constant. Similarly, for $\langle G_o(E^k-2E^{k+1}),\phi^{k+1}-\phi^k\rangle$, it holds

$$\begin{aligned}&\langle G_o(E^k-2E^{k+1}),\phi^{k+1}-\phi^k\rangle\\&\le\frac{c_2}{2}\|G_o(E^k-2E^{k+1})\|\|\phi^{k+1}-\phi^k\|^2\\&\quad+\frac{1}{2c_2}\|G_o(E^k-2E^{k+1})\|\\&\le\frac{c_2\sigma_{\max}(G_o)(\|E^k\|+2\|E^{k+1}\|)}{2}\|\phi^{k+1}-\phi^k\|^2\\&\quad+\frac{\sigma_{\max}(G_o)}{2c_2}(\|E^k\|+2\|E^{k+1}\|),\end{aligned}\tag{33}$$

where $c_2>0$ is any positive constant. For $\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_oE^{k+1}\rangle$, we have

$$\begin{aligned}&\frac{c}{2}\langle G_o(E^k-E^{k+1}),G_oE^{k+1}\rangle\\&\le\frac{c}{2}\langle G_oE^k,G_oE^{k+1}\rangle\\&\le\frac{c}{4}\|G_oE^k\|^2+\|G_oE^{k+1}\|^2\\&\le\frac{c}{4}\sigma_{\max}^2(G_o)\|E^k\|^2+\frac{c}{4}\sigma_{\max}^2(G_o)\|E^{k+1}\|^2.\end{aligned}\tag{34}$$

Using (32), (33) and (34) to rewrite (31) followed by substituting the result and (26) into (25), we obtain

$$\begin{aligned}&V^k-V^{k+1}-c\|z^k-z^{k+1}\|^2\\&\quad-\left(\frac{\rho}{2}-\frac{M}{4}\right)\|x^k-x^{k+1}\|^2-\frac{1}{c}\|\phi^k-\phi^{k+1}\|^2\\&\quad+\frac{c_1\sigma_{\max}(G_o)\|E^{k+1}\|}{2}\|\phi^k-\phi^*\|^2+\frac{\sigma_{\max}(G_o)}{2c_1}\|E^{k+1}\|\\&\quad+\frac{c_2\sigma_{\max}(G_o)(\|E^k\|+2\|E^{k+1}\|)}{2}\|\phi^{k+1}-\phi^k\|^2\\&\quad+\frac{\sigma_{\max}(G_o)}{2c_2}(\|E^k\|+2\|E^{k+1}\|)\\&\quad+\frac{c}{4}\sigma_{\max}^2(G_o)\|E^k\|^2+\frac{c}{4}\sigma_{\max}^2(G_o)\|E^{k+1}\|^2\ge0.\end{aligned}\tag{35}$$

Step 2: Now we characterize the upper bound of $\|E^k\|$. According to the censoring strategy, $\hat{x}_i^k-x_i^k$, the ith block

of E^k, becomes 0 if $\|\hat{x}_i^{k-1} - x_i^k\| \geq \tau^k$ or equals $\hat{x}_i^{k-1} - x_i^k$ otherwise. In both cases, it holds $\|\hat{x}_i^k - x_i^k\| \leq \tau^k$. Therefore, we know $\|E^k\| \leq \sqrt{n}\tau^k$. Since τ^k is non-increasing, it also holds $\|E^{k+1}\| \leq \sqrt{n}\tau^{k+1} \leq \sqrt{n}\tau^k$. Thus, (35) becomes

$$\begin{aligned} &c\|z^k - z^{k+1}\|^2 + \left(\frac{\rho}{2} - \frac{M}{4}\right)\|x^k - x^{k+1}\|^2 \\ &\quad + \left(\frac{1}{c} - \frac{3c_2\sigma_{\max}(G_o)\sqrt{n}\tau^k}{2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &\leq V^k - V^{k+1} + \frac{c_1\sigma_{\max}(G_o)\sqrt{n}\tau^k}{2}\|\phi^k - \phi^*\|^2 \\ &\quad + \left(\frac{1}{2c_1} + \frac{3}{2c_2}\right)\sigma_{\max}(G_o)\sqrt{n}\tau^k + \frac{cn\sigma_{\max}^2(G_o)}{2}(\tau^k)^2. \end{aligned} \tag{36}$$

Setting the constants c_1 and c_2 in (36) as

$$c_1 = 3c_2 = \frac{1}{c\sigma_{\max}(G_o)\sqrt{n}\tau^0},$$

we rewrite (36) to

$$\begin{aligned} &c\|z^k - z^{k+1}\|^2 + \left(\frac{\rho}{2} - \frac{M}{4}\right)\|x^k - x^{k+1}\|^2 \\ &\quad + \left(\frac{1}{c} - \frac{\tau^k}{2c\tau^0}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &\leq V^k - V^{k+1} + \frac{\tau^k}{2c\tau^0}\|\phi^k - \phi^*\|^2 \\ &\quad + 5cn\sigma_{\max}^2(G_o)\tau^0\tau^k + \frac{cn\sigma_{\max}^2(G_o)(\tau^k)^2}{2}. \end{aligned} \tag{37}$$

Since τ^k is non-decreasing, $\frac{1}{c} - \frac{\tau^k}{2c\tau^0} \geq \frac{1}{2c}$. Meanwhile, by the definition of the energy function, $V^k \geq \frac{1}{c}\|\phi^k - \phi^*\|^2$. By the definitions of $z^k = \frac{1}{2}G_u x^k$ and $L_u = \frac{1}{2}G_u^T G_u$, $\|z^k - z^{k+1}\|^2 = \frac{1}{4}\|G_u(x^k - x^{k+1})\|^2 \geq \frac{1}{2}\lambda_{\min}(L_u)\|x^k - x^{k+1}\|^2$. Applying these three facts to (37) yields

$$\begin{aligned} &\frac{1}{2}\left(c\lambda_{\min}(L_u) + \rho - \frac{M}{2}\right)\|x^k - x^{k+1}\|^2 + \frac{1}{2c}\|\phi^k - \phi^{k+1}\|^2 \\ &\leq \left(1 + \frac{\tau^k}{2\tau^0}\right)V^k - V^{k+1} \\ &\quad + 5cn\sigma_{\max}^2(G_o)\tau^0\tau^k + \frac{cn\sigma_{\max}^2(G_o)(\tau^k)^2}{2}. \end{aligned} \tag{38}$$

Step 3: Define

$$\theta^k := 5cn\sigma_{\max}^2(G_o)\tau^0\tau^k + \frac{cn\sigma_{\max}^2(G_o)(\tau^k)^2}{2},$$

which is a non-increasing non-negative summable summable sequence as τ^k is. The left-hand side of (38) is non-negative because $c\lambda_{\min}(L_u) + \rho \geq \frac{M}{2}$. Thus, (38) leads to

$$\left(1 + \frac{\tau^k}{2\tau^0}\right)V^k - V^{k+1} + \theta^k \geq 0. \tag{39}$$

We use this inequality to show that V^k has a finite upper bound. From (39) we have

$$\begin{aligned} V^{k+1} &\leq \left(1 + \frac{\tau^k}{2\tau^0}\right)V^k + \theta^k \\ &\leq \left(1 + \frac{\tau^k}{2\tau^0}\right)\left(\left(1 + \frac{\tau^{k-1}}{2\tau^0}\right)V^{k-1} + \theta^{k-1}\right) + \theta^k \\ &\leq \quad \cdots \\ &\leq V^0\prod_{k'=0}^{k}\left(1 + \frac{\tau^{k'}}{2\tau^0}\right) + \sum_{k''=0}^{k-1}\left(\theta^{k''}\prod_{k'=k''+1}^{k}\left(1 + \frac{\tau^{k'}}{2\tau^0}\right)\right) + \theta^k \\ &\leq V^0\prod_{k'=0}^{k}\left(1 + \frac{\tau^{k'}}{2\tau^0}\right) + \sum_{k''=0}^{k}\theta^{k''}\prod_{k'=0}^{k}\left(1 + \frac{\tau^{k'}}{2\tau^0}\right) \\ &\leq \left(V^0 + \sum_{k''=0}^{\infty}\theta^{k''}\right)\prod_{k'=0}^{\infty}\left(1 + \frac{\tau^{k'}}{2\tau^0}\right) \\ &\leq \left(V^0 + \sum_{k''=0}^{\infty}\theta^{k''}\right)\exp\left\{\sum_{k'=0}^{\infty}\frac{\tau^{k'}}{2\tau^0}\right\} < \infty, \end{aligned} \tag{40}$$

where we use the inequality $1 + a \leq \exp\{a\}$ that holds for all $a \in \mathcal{R}$, and the fact that τ^k and θ^k are both non-negative and summable. Thus, we conclude that V^k has a finite upper bound, denoted as $\bar{V}$.

Step 4: Now we begin to prove the convergence. Summing up (38) from $k = 0$ to $k = \infty$ yields

$$\begin{aligned} &\sum_{k=0}^{\infty}\left[\frac{1}{2}\left(c\lambda_{\min}(L_u) + \rho - \frac{M}{2}\right)\|x^k - x^{k+1}\|^2 \right.\\ &\qquad \left. + \frac{1}{2c}\|\phi^k - \phi^{k+1}\|^2\right] \\ &\leq V^0 + \sum_{k=0}^{\infty}\frac{\tau^k}{2\tau^0}V^k + \sum_{k=0}^{\infty}\theta^k \\ &\leq V^0 + \frac{\bar{V}}{2\tau^0}\sum_{k=0}^{\infty}\tau^k + \sum_{k=0}^{\infty}\theta^k < \infty. \end{aligned} \tag{41}$$

Thus, we conclude that $\lim_{k\to\infty}(x^k - x^{k+1}) = 0$ and $\lim_{k\to\infty}(\phi^k - \phi^{k+1}) = 0$. Following these limiting properties, when $k \to \infty$, the dual update (18) leads to $G_o\hat{x}^k \to 0$, which implies that

$$G_o x^k = G_o\hat{x}^k + G_o E^k \to 0. \tag{42}$$

Also, we have $L_o\hat{x}^k \to 0$ as $L_o = \frac{1}{2}G_o^T G_o$. Consequently, in the limit (17) becomes

$$\nabla f(x^k) + G_o^T\phi^k \to 0. \tag{43}$$

Meanwhile, by definition

$$\frac{1}{2}G_u x^k - z^k = 0. \tag{44}$$

Comparing (43), (42) and (44) with the KKT conditions (14), (15) and (16), we conclude that the triple (x^k, z^k, ϕ^k) satisfies the KKT conditions when k goes to infinity.

Next, we show that $\{(x^k, z^k, \phi^k)\}$ converges when $k \to \infty$. Since the sequence V^k is bounded, $\|x^k - x^*\|$ and $\|\phi^k - \phi^*\|$ are also bounded. Thus, there exists a subsequence $\{(x^{k_t}, \phi^{k_t})\}$ which converges to a cluster point (x^∞, ϕ^∞) of $\{(x^k, \phi^k)\}$ and (x^∞, ϕ^∞) is optimal to (3).

Construct another energy function $V_\infty^k := \frac{\rho}{2}\|x^k - x^\infty\|^2 + c\|z^k - z^\infty\|^2 + \frac{1}{c}\|\phi^k - \phi^\infty\|^2$, where $z^\infty := \frac{1}{2}G_u x^\infty$. The analysis for V^k can be applied to V_∞^k. In particular, analogous to (40), given any fixed k_t, we have

$$\begin{aligned} V_\infty^k &\le \left(V_\infty^{k_t} + \sum_{k''=k_t}^{\infty} \theta^{k''}\right) \exp\left\{\sum_{k'=k_t}^{\infty} \frac{\tau^{k'}}{\tau^{k_t}}\right\} \\ &\le \left(V_\infty^{k_t} + \sum_{k''=k_t}^{\infty} \theta^{k''}\right) \exp\left\{\sum_{k'=k_t}^{\infty} \frac{\tau^{k'}}{\tau^{0}}\right\}, \end{aligned} \tag{45}$$

for any $k \ge k_t$. Observe that $(x^{k_t}, \phi^{k_t}) \to (x^\infty, \phi^\infty)$ leads to $V_\infty^{k_t} \to 0$. In addition, the sequences θ^k and τ^k $\sum_{k''=0}^{\infty} \theta^{k''} < \infty$ and $\sum_{k'=0}^{\infty} \tau^{k'} < \infty$, respectively. Therefore, for any $\epsilon > 0$ there exists an integer t_0 such that

$$V_\infty^{k_{t_0}} < \frac{\epsilon}{4}, \sum_{k''=k_{t_0}}^{\infty} \theta^{k''} < \frac{\epsilon}{4}, \text{and} \sum_{k'=k_{t_0}}^{\infty} \tau^{k'} < \tau^0 \log 2.$$

Then according to (45) we have $V_\infty^k < \epsilon$ for all $k \ge k_{t_0}$. Therefore, $V_\infty^k \to 0$ as $k \to \infty$. From the definition of V_∞^k, we conclude that $\{(x^k, z^k, \phi^k)\}$ converges to $(x^\infty, z^\infty, \phi^\infty)$, which is optimal to (3). ■

Appendix C
Proof of Theorem 2

Proof. Step 1: From Assumption 5, the local cost functions f_i are strongly convex with constant $m > 0$. Thus, we have

$$\begin{aligned} &m\|x^{k+1} - x^*\|^2 \\ &\le \langle \nabla f(x^{k+1}) - \nabla f(x^*), x^{k+1} - x^* \rangle \\ &= \langle \nabla f(x^k) - \nabla f(x^*), x^{k+1} - x^* \rangle \\ &\quad + \langle \nabla f(x^{k+1}) - \nabla f(x^k), x^{k+1} - x^* \rangle. \end{aligned} \tag{46}$$

Observe that $\langle \nabla f(x^k) - \nabla f(x^*), x^{k+1} - x^* \rangle$, the first term at the right-hand side of (46), also appears in (25) in the proof of Theorem 1. We follow the derivation to obtain (31), but then look for new upper bounds of $\langle G_o E^{k+1}, \phi^k - \phi^* \rangle$ and $\langle G_o E^k, \phi^{k+1} - \phi^k \rangle$, which are different to those in (32) and (33). For the term $\langle G_o E^{k+1}, \phi^k - \phi^* \rangle$, we observe that

$$\begin{aligned} &\langle G_o E^{k+1}, \phi^k - \phi^* \rangle \\ &= \langle G_o E^{k+1}, \phi^{k+1} - \phi^* \rangle + \langle G_o E^{k+1}, \phi^k - \phi^{k+1} \rangle \\ &\le \frac{c_1}{2}\|G_o E^{k+1}\|^2 + \frac{1}{2c_1}\|\phi^{k+1} - \phi^*\|^2 \\ &\quad + \frac{c_1}{2}\|G_o E^{k+1}\|^2 + \frac{1}{2c_1}\|\phi^k - \phi^{k+1}\|^2 \\ &\le c_1 \sigma_{\max}^2(G_o)\|E^{k+1}\|^2 \\ &\quad + \frac{1}{2c_1}\|\phi^{k+1} - \phi^*\|^2 + \frac{1}{2c_1}\|\phi^k - \phi^{k+1}\|^2, \end{aligned} \tag{47}$$

where $c_1 > 0$ is any positive constant. For $\langle G_o E^k, \phi^{k+1} - \phi^k \rangle$, it holds

$$\begin{aligned} &\langle G_o E^k, \phi^{k+1} - \phi^k \rangle \\ &\le \frac{c_2}{2}\|G_o E^k\|^2 + \frac{1}{2c_2}\|\phi^{k+1} - \phi^k\|^2 \\ &\le \frac{c_2 \sigma_{\max}^2(G_o)\|E^k\|^2}{2} + \frac{1}{2c_2}\|\phi^{k+1} - \phi^k\|^2, \end{aligned} \tag{48}$$

where $c_2 > 0$ is any positive constant.

For the second term at the right-hand side of (46), we have

$$\begin{aligned} &\langle \nabla f(x^{k+1}) - \nabla f(x^k), x^{k+1} - x^* \rangle \\ &\le \frac{c_3}{2}\|\nabla f(x^{k+1}) - \nabla f(x^k)\|^2 + \frac{1}{2c_3}\|x^{k+1} - x^*\|^2 \\ &\le \frac{c_3 M^2}{2}\|x^{k+1} - x^k\|^2 + \frac{1}{2c_3}\|x^{k+1} - x^*\|^2, \end{aligned} \tag{49}$$

where $c_3 > 0$ is any positive constant. The last inequality uses the fact that the gradients of the local cost functions ∇f_i are Lipschitz continuous with constant $M > 0$ according to Assumption 3.

Using (47), (48) and (34) to rewrite (31) followed by substituting the result and (49) into (46), we obtain

$$\begin{aligned} V^{k+1} \le\ & V^k - c\|z^k - z^{k+1}\|^2 \\ & - \left(\frac{\rho}{2} - \frac{c_3 M^2}{2}\right)\|x^k - x^{k+1}\|^2 \\ & - \left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ & - \left(m - \frac{1}{2c_3}\right)\|x^{k+1} - x^*\|^2 + \frac{1}{2c_1}\|\phi^{k+1} - \phi^*\|^2 \\ & + \left(\frac{c_2}{2} + \frac{c}{4}\right)\sigma_{\max}^2(G_o)\|E^k\|^2 \\ & + \left(c_1 + \frac{c}{4}\right)\sigma_{\max}^2(G_o)\|E^{k+1}\|^2. \end{aligned} \tag{50}$$

By the same reasoning in the proof of Theorem 1, $\|E^k\| \le \sqrt{n}\tau^k$ and $\|E^{k+1}\| \le \sqrt{n}\tau^k$. Thus, (50) becomes

$$\begin{aligned} V^{k+1} \le\ & V^k - c\|z^k - z^{k+1}\|^2 \\ & - \left(\frac{\rho}{2} - \frac{c_3 M^2}{2}\right)\|x^k - x^{k+1}\|^2 \\ & - \left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ & - \left(m - \frac{1}{2c_3}\right)\|x^{k+1} - x^*\|^2 + \frac{1}{2c_1}\|\phi^{k+1} - \phi^*\|^2 \\ & + sn(\tau^k)^2. \end{aligned} \tag{51}$$

where

$$s := \left(c_1 + \frac{c_2}{2} + \frac{c}{2}\right)\sigma_{\max}^2(G_o) > 0.$$

Step 2: Now we are going to find constants $\delta > 0$ and $\gamma \ge 0$ such that

$$(1 + \delta)V^{k+1} \le V^k + \gamma n(\tau^k)^2. \tag{52}$$

Given any δ, using the definition of the energy function V^{k+1} to rewrite (51) as

$$\begin{aligned}(1+\delta)V^{k+1} &\le V^k - c\|z^k - z^{k+1}\|^2 \\ &- \left(\frac{\rho}{2} - \frac{c_3\, M^2}{2}\right)\|x^k - x^{k+1}\|^2 \\ &- \left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &- \left(m - \frac{1}{2c_3} - \frac{\rho\delta}{2}\right)\|x^{k+1} - x^*\|^2 \\ &+ \left(\frac{1}{2c_1} + \frac{\delta}{c}\right)\|\phi^{k+1} - \phi^*\|^2 + c\delta\|z^{k+1} - z^*\|^2 \\ &+ sn(\tau^k)^2. \end{aligned} \tag{53}$$

We shall replace the terms $\|z^{k+1} - z^*\|^2$ and $\|\phi^{k+1} - \phi^*\|^2$ in (53) with terms $\|z^k - z^{k+1}\|^2$, $\|z^{k+1} - z^*\|^2$, $\|x^k - x^{k+1}\|^2$, $\|x^{k+1} - x^*\|^2$ and $(\tau^k)^2$.

For $\|z^{k+1} - z^*\|^2$, because $z^{k+1} - z^* = \frac{G_u}{2}(x^{k+1} - x^*)$, we have

$$\|z^{k+1} - z^*\|^2 \le \frac{\sigma_{\max}^2(G_u)}{4}\|x^{k+1} - x^*\|^2. \tag{54}$$

To handle $\|\phi^{k+1} - \phi^*\|^2$, use the fact that $L_u = \frac{1}{2}G_u^T G_u$ and reorganize (19) to obtain

$$\begin{aligned}&G_o^T(\phi^{k+1} - \phi^*) \\ &= -\big(\nabla f(x^k) - \nabla f(x^*)\big) + cG_u^T(z^k - z^{k+1}) \\ &\quad + \rho(x^k - x^{k+1}) + cL_o(E^k - E^{k+1}). \end{aligned} \tag{55}$$

Since both ϕ^{k+1} and ϕ^* are in the column space of G_o, the left-hand side of (55) is lower-bounded by

$$\tilde{\sigma}_{\min}^2(G_o)\|\phi^{k+1} - \phi^*\|^2 \le \|G_o^T(\phi^{k+1} - \phi^*)\|^2. \tag{56}$$

The right-hand side of (55) is upper-bounded by

$$\begin{aligned}&\| -\big(\nabla f(x^k) - \nabla f(x^*)\big) + cG_u^T(z^k - z^{k+1}) \\ &\quad + \rho(x^k - x^{k+1}) + cL_o(E^k - E^{k+1})\|^2 \\ &\le 4\|\nabla f(x^k) - \nabla f(x^*)\|^2 + 4\|cG_u^T(z^k - z^{k+1})\|^2 \\ &\quad + 4\|\rho(x^k - x^{k+1})\|^2 + 4\|cL_o(E^k - E^{k+1})\|^2 \\ &\le 8\|\nabla f(x^{k+1}) - \nabla f(x^*)\|^2 + 8\|\nabla f(x^k) - \nabla f(x^{k+1})\|^2 \\ &\quad + 4\|cG_u^T(z^k - z^{k+1})\|^2 + 4\|\rho(x^k - x^{k+1})\|^2 \\ &\quad + 8\|cL_oE^k\|^2 + 8\|cL_oE^{k+1}\|^2 \\ &\le 8\,M^2\|x^{k+1} - x^*\|^2 + (8\,M^2 + 4\rho^2)\|x^k - x^{k+1}\|^2 \\ &\quad + 4c^2\sigma_{\max}^2(G_u)\|z^k - z^{k+1}\|^2 + 4c^2\sigma_{\max}^4(G_o)n(\tau^k)^2. \end{aligned} \tag{57}$$

The last inequality uses the fact that ∇f is Lipschitz continuous with constant $M > 0$ such that

$$\begin{aligned}\|\nabla f(x^{k+1}) - \nabla f(x^*)\|^2 &\le M^2\|x^{k+1} - x^*\|^2, \\ \|\nabla f(x^k) - \nabla f(x^{k+1})\|^2 &\le M^2\|x^k - x^{k+1}\|^2,\end{aligned}$$

and the definition of $L_o = \frac{1}{2}G_o^T G_o$ such that

$$\begin{aligned}\|cL_oE^k\|^2 &\le \frac{c^2}{4}\sigma_{\max}^2(G_o)\|E^k\|^2 \le \frac{c^2}{4}\sigma_{\max}^4(G_o)n(\tau^k)^2, \\ \|cL_oE^{k+1}\|^2 &\le \frac{c^2}{4}\sigma_{\max}^4(G_o)n(\tau^k)^2.\end{aligned}$$

Combining (55), (56) and (57), we obtain

$$\begin{aligned}\|\phi^{k+1} - \phi^*\|^2 &\le \frac{1}{\tilde{\sigma}_{\min}^2(G_o)}\big(8\,M^2\|x^{k+1} - x^*\|^2 \\ &+ (8\,M^2 + 4\rho^2)\|x^k - x^{k+1}\|^2 + 4c^2\sigma_{\max}^2(G_u)\|z^k - z^{k+1}\|^2 \\ &+ 4c^2\sigma_{\max}^4(G_o)n(\tau^k)^2\big). \end{aligned} \tag{58}$$

Thus, we can use (54) and (58) to rewrite (53) as

$$\begin{aligned}&(1+\delta)V^{k+1} \\ &\le V^k - c\left(1 - \left(\frac{1}{2c_1} + \frac{\delta}{c}\right)\frac{4c\sigma_{\max}^2(G_u)}{\tilde{\sigma}_{\min}^2(G_o)}\right)\|z^k - z^{k+1}\|^2 \\ &\quad - \left(\frac{\rho}{2} - \frac{c_3\, M^2}{2} - \left(\frac{1}{2c_1} + \frac{\delta}{c}\right)\frac{(8\,M^2 + 4\rho^2)}{\tilde{\sigma}_{\min}^2(G_o)}\right) \\ &\quad \|x^k - x^{k+1}\|^2 - \left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &\quad - \Bigg(m - \frac{1}{2c_3} - \frac{\rho\delta}{2} - \frac{c\delta\sigma_{\max}^2(G_u)}{4} \\ &\qquad - \left(\frac{1}{2c_1} + \frac{\delta}{c}\right)\frac{8\,M^2}{\tilde{\sigma}_{\min}^2(G_o)}\Bigg). \\ &\quad \|x^{k+1} - x^*\|^2 + \left(s + \left(\frac{1}{2c_1} + \frac{\delta}{c}\right)\frac{4c^2\sigma_{\max}^4(G_o)}{\tilde{\sigma}_{\min}^2(G_o)}\right)n(\tau^k)^2. \end{aligned} \tag{59}$$

For convenience, set the constants as

$$\begin{aligned}c_1 &= \frac{c}{2} + \frac{1}{m(2m\rho - M^2)\tilde{\sigma}_{\min}^2(G_o)}\Big(4\,M^2(2m\rho + M^2) \\ &\qquad + 16\,m^2(2\,M^2 + \rho^2) + 2c\sigma_{\max}^2(G_u)m(2m\rho - M^2)\Big), \\ c_2 &= \frac{c_1 c}{2c_1 - c}, \\ c_3 &= \frac{1/2m + \rho/M^2}{2} = \frac{2m\rho + M^2}{4\,mM^2} \in \left(\frac{1}{2\,m}, \frac{\rho}{M^2}\right),\end{aligned}$$

where the range of c_3 is from the hypothesis that $\rho > \frac{M^2}{2m}$. Then (52) is achieved with constants

$$\begin{aligned}\delta &\le \min\left\{\frac{\tilde{\sigma}_{\min}^2(G_o)}{4\sigma_{\max}^2(G_u)} - \frac{c}{2c_1}, \frac{c\tilde{\sigma}_{\min}^2(G_o)(2m\rho - M^2)}{32m(\rho^2 + 2\,M^2)} - \frac{c}{2c_1},\right. \\ &\qquad \left.\frac{\frac{m(2m\rho - M^2)}{2m\rho + M^2} - \frac{4\,M^2}{c_1\tilde{\sigma}_{\min}^2(G_o)}}{\left(\frac{\rho}{2} + \frac{c\sigma_{\max}^2(G_u)}{4} + \frac{8\,M^2}{c\tilde{\sigma}_{\min}^2(G_o)}\right)}\right\}, \\ \gamma &= s + \left(\frac{\delta}{c} + \frac{1}{2c_1}\right)\frac{4c^2\sigma_{\max}^4(G_o)}{\tilde{\sigma}_{\min}^2(G_o)}.\end{aligned}$$

Note that $\delta > 0$ and $\gamma > 0$.

Step 3: Now we prove the linear convergence of V^k to 0, which implies the linear convergence of x^k to x^*. Using the censoring threshold rule $\tau^k = \alpha \cdot (\beta)^k$, we further rewrite (52) as

$$(1+\delta)V^{k+1} \leq V^k + \gamma n \alpha^2 \cdot (\beta^2)^k.$$

Analogous to the technique used in handling (40), it holds

$$\begin{aligned} V^{k+1} &\leq (1+\delta)^{-1}\left(V^k + \gamma n\alpha^2\cdot(\beta^2)^k\right) \\ &\leq (1+\delta)^{-1}[(1+\delta)^{-1}\left(V^{k-1} + \gamma n\alpha^2\cdot(\beta^2)^{k-1}\right) \\ &\qquad + \gamma n\alpha^2\cdot(\beta^2)^k] \\ &\leq \qquad \cdots \\ &\leq (1+\delta)^{-(k+1)}V^0 + Cn\alpha^2\sum_{k'=0}^{k} \\ &\qquad \left((1+\delta)^{-(k+1-k')}(\beta^2)^{k'}\right) \\ &= (1+\delta)^{-(k+1)}\left[V^0 + \gamma n\alpha^2\sum_{k'=0}^{k}\left((1+\delta)\beta^2\right)^{k'}\right] \\ &\leq (1+\delta)^{-(k+1)}\left(V^0 + \frac{\gamma n\alpha^2}{1-(1+\delta)\beta^2}\right), \end{aligned}$$

where the last inequality holds when $(1+\delta)\beta^2 < 1$.

In summary, for any positive $\delta > 0$ that satisfies

$$\delta \leq \min\left\{\frac{\tilde{\sigma}^2_{\min}(G_o)}{4\sigma^2_{\max}(G_u)} - \frac{c}{2c_1}, \frac{c\tilde{\sigma}^2_{\min}(G_o)(2m\rho - M^2)}{32m(\rho^2 + 2M^2)} - \frac{c}{2c_1}, \right.$$
$$\left. \frac{\frac{m(2m\rho - M^2)}{2m\rho + M^2} - \frac{4M^2}{c_1\tilde{\sigma}^2_{\min}(G_o)}}{\left(\frac{\rho}{2} + \frac{c\sigma^2_{\max}(G_u)}{4} + \frac{8M^2}{c\tilde{\sigma}^2_{\min}(G_o)}\right)}, \frac{1}{\beta^2} - 1\right\}, \qquad (60)$$

the energy function V^k converges to 0 at a linear rate of $\mathcal{O}((1+\delta)^{-k})$. Moreover, by the definition of V^k, it holds that $V^k \geq \frac{\rho}{2}\|x^k - x^*\|^2$. Thus, the primal variable x^k converges to the unique optimal solution x^* at $\mathcal{O}((1+\delta)^{-\frac{k}{2}})$. ■

Remark 4: If we set $c = \frac{8M}{\sigma_{\max}(G_u)\tilde{\sigma}_{\min}(G_o)}$ and $\rho = M\kappa_f$ and further set $\frac{1}{2c_1} = \frac{\delta}{c}$, $\frac{1}{2c_2} = \frac{1}{c} - \frac{1}{2c_1}$ and $c_3 = \frac{2}{3m}$ in (59), then following the proof of Theorem 2, we can derive another upper bound of δ as shown in (22) of Corollary 1.

Appendix D
Proof of Theorem 3

Proof. Step 1: As in Step 1 of the proof of Theorem 2, we obtain the inequality (51).

Step 2: Now our aim is different to that in Step 2 of the proof of Theorem 2, as we are going to find constants $q > 0$ and $\eta^k \geq 0$ as well as a time index k_0 such that

$$(k+1)^q\left(V^{k+1} + \eta^{k+1}\right) \leq (k)^q\left(V^k + \eta^k\right), \qquad (61)$$

for all $k \geq k_0$.

From (61), in which k_0, q and η^k will be determined later, we have

$$\begin{aligned} &(k+1)^q\left(V^{k+1} + \eta^{k+1}\right) \\ &\quad = (k)^q V^{k+1} + [(k+1)^q - (k)^q]\left(\frac{\rho}{2}\|x^{k+1} - x^*\|^2\right. \\ &\qquad \left. + c\|z^{k+1} - z^*\|^2 + \frac{1}{c}\|\phi^{k+1} - \phi^*\|^2\right) + (k+1)^q\eta^{k+1} \\ &\overset{(51)}{\leq} (k)^q V^k - c(k)^q\|z^k - z^{k+1}\|^2 \\ &\qquad - (k)^q\left(\frac{\rho}{2} - \frac{c_3 M^2}{2}\right)\|x^k - x^{k+1}\|^2 \\ &\qquad - (k)^q\left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &\qquad - \left\{(k)^q\left(m - \frac{1}{2c_3}\right) - \left[(k+1)^q - (k)^q\right]\frac{\rho}{2}\right\}\cdot \\ &\qquad \|x^{k+1} - x^*\|^2 \\ &\qquad + c\left[(k+1)^q - (k)^q\right]\|z^{k+1} - z^*\|^2 \\ &\qquad + \left[\frac{(k)^q}{2c_1} + \frac{(k+1)^q - (k)^q}{c}\right]\|\phi^{k+1} - \phi^*\|^2 \\ &\qquad + (k)^q sn(\tau^k)^2 + (k+1)^q\eta^{k+1} \\ &\overset{(54),(58)}{\leq} (k)^q V^k - (k)^q\left[c - \frac{1}{2c_1} - \frac{(k+1)^q - (k)^q}{(k)^q}\cdot\right. \\ &\qquad \left.\frac{4c\sigma^2_{\max}(G_u)}{\tilde{\sigma}^2_{\min}(G_o)}\right]\|z^k - z^{k+1}\|^2 \\ &\qquad - (k)^q\left\{\frac{\rho}{2} - \frac{c_3 M^2}{2} - \left(\frac{c}{2c_1} + \frac{(k+1)^q - (k)^q}{(k)^q}\right)\cdot\right. \\ &\qquad \left.\frac{8M^2 + 4\rho^2}{c\tilde{\sigma}^2_{\min}(G_o)}\right\}\|x^k - x^{k+1}\|^2 \\ &\qquad - (k)^q\left[m - \frac{1}{2c_3} - \frac{4M^2}{c_1\tilde{\sigma}^2_{\min}(G_o)} - \frac{(k+1)^q - (k)^q}{(k)^q}\cdot\right. \\ &\qquad \left.\left(\frac{\rho}{2} + \frac{c\sigma^2_{\max}(G_u)}{4} + \frac{8M^2}{c\tilde{\sigma}^2_{\min}(G_o)}\right)\right] \\ &\qquad \|x^{k+1} - x^*\|^2 - (k)^q\left(\frac{1}{c} - \frac{1}{2c_1} - \frac{1}{2c_2}\right)\|\phi^k - \phi^{k+1}\|^2 \\ &\qquad + (k)^q t^k n(\tau^k)^2 + (k+1)^q\eta^{k+1}, \end{aligned}$$

where

$$t^k := s + \frac{2c^2\sigma^4_{\max}(G_o)}{c_1\tilde{\sigma}^2_{\min}(G_o)} + \frac{(k+1)^q - (k)^q}{(k)^q}\frac{4c\sigma^4_{\max}(G_o)}{\tilde{\sigma}^2_{\min}(G_o)} > 0.$$

Set the constants c_1, c_2 and c_3 the same values as those in the proof of Theorem 2. Notice that $\frac{(k+1)^q-(k)^q}{(k)^q} = (1+\frac{1}{k})^q - 1 \to 0$ as k goes to infinity. Then, there exists a time index k_0 such that for any $k \geq k_0$, it holds

$$\frac{(k+1)^q-(k)^q}{(k)^q} \leq \min\left\{ \frac{\tilde{\sigma}^2_{\min}(G_o)}{4\sigma^2_{\max}(G_u)}\left(c-\frac{1}{2c_1}\right), \right.$$
$$\frac{c\tilde{\sigma}^2_{\min}(G_o)}{4(\rho^2+2\,M^2)}\left(\frac{\rho}{2}-\frac{c_3\,M^2}{2}-\frac{2(\rho^2+2\,M^2)}{c_1\tilde{\sigma}^2_{\min}(G_o)}\right),$$
$$\left.\frac{m-\frac{1}{2c_3}-\frac{4\,M^2}{c_1\tilde{\sigma}^2_{\min}(G_o)}}{\frac{\rho}{2}+\frac{c\sigma^2_{\max}(G_u)}{4}+\frac{8\,M^2}{c\tilde{\sigma}^2_{\min}(G_o)}}, \frac{\tilde{\sigma}^2_{\min}(G_o)}{4c\sigma^4_{\max}(G_o)}\right\}, \tag{62}$$

where the right-hand side is larger than 0. In this situation, $t^k \leq t := s + \frac{2c^2\sigma^4_{\max}(G_o)}{c_1\tilde{\sigma}^2_{\min}(G_o)} + 1$. Further, using the censoring threshold $\tau^k = \frac{\alpha}{(k)^r}$, we have

$$(k+1)^q\left(V^{k+1}+\eta^{k+1}\right) \leq (k)^q V^k + \frac{tn\alpha^2}{(k)^{2r-q}} + (k+1)^q\eta^{k+1}. \tag{63}$$

Now we determine the values of q and η^k. Since $\sum_{k'=k}^{\infty}\frac{1}{(k')^{2r-q}} < \infty$ for any time index k when $2r-q>1$, setting $(k)^q\eta^k := \sum_{k'=k}^{\infty}\frac{tn\alpha^2}{(k')^{2r-q}}$ in (63) leads to an equivalent form

$$(k+1)^q\left(V^{k+1}+\eta^{k+1}\right) \leq (k)^q\left(V^k+\eta^k\right),$$

which is exactly what we want in (61). Therefore, for any $k \geq k_0$, it holds

$$V^k \leq V^k + \eta^k \leq \frac{(k_0)^q\left(V^{k_0}+\eta^{k_0}\right)}{(k)^q}.$$

That is, the energy function V^k converges to 0 at a sublinear rate of $\mathcal{O}((k)^{-q})$. Moreover, by the definition of V^k, it holds that $V^k \geq \frac{\rho}{2}\|x^k - x^*\|^2$. Thus, the primal variable x^k converges to the unique optimal solution x^* at a sublinear rate of $\mathcal{O}((k)^{-\frac{q}{2}})$. ■

REFERENCES

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

[2] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[3] S. Dougherty and M. Guay, "An extremum-seeking controller for distributed optimization over sensor networks," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 928–933, Feb. 2017.

[4] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multi-hop wideband cognitive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 37–48, Feb. 2011.

[5] G. Giannakis, Q. Ling, G. Mateos, I. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication and Imaging, Science and Engineering*. New York, NY, USA: Springer, 2016.

[6] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[7] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[8] G. Giannakis, V. Kekatos, N. Gatsis, S. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[9] K. Antoniadou-Plytaria, I. Kouveliotis-Lysikatos, P. Georgilakis, and N. Hatziargyriou, "Distributed and decentralized voltage control of smart distribution networks: Models, methods, and future research," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2999–3008, 2017.

[10] H. Liu, W. Shi, and H. Zhu, "Distributed voltage control in distribution networks: Online and robust implementations," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6106–6117, Nov. 2018.

[11] X. Zhao and A. Sayed, "Distributed clustering and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.

[12] A. Mokhtari, "Efficient methods for large-scale empirical risk minimization," Ph.D. dissertation, Dept. Elect. Syst. Eng., Univ. Pennsylvania, Philadelphia, PA, USA, 2017.

[13] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5336–5346.

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

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

[16] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 30, no. 5, pp. 1835–1854, 2016.

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

[18] K. Tsianos and M. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. Amer. Control Conf.*, 2012, pp. 1067–1072.

[19] S. Lee, A. Nedic, and M. Raginsky, "Stochastic dual averaging for decentralized online optimization on time-varying communication graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6407–6414, Dec. 2017.

[20] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network Newton distributed optimization methods," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, Jan. 2017.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, 2010.

[22] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[23] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[24] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, pp. 4051–4064, Apr. 2015.

[25] T. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.

[26] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[27] P. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[28] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 788–794.

[29] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[30] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[31] R. Xin and U. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 325–330, Jul. 2018.

[32] S. Pu, W. Shi, J. Xu, and A. Nedic, "A push-pull gradient method for distributed optimization in networks," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 3385–3390.

[33] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5158–5173, Oct. 2016.

[34] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 2, no. 4, pp. 507–522, Dec. 2016.
[35] M. Eisen, A. Mokhtari, and A. Ribeiro, "A primal-dual Quasi-Newton method for exact consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 5983–5997, Dec. 2019.
[36] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulie, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3027–3036.
[37] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulie, "Optimal algorithms for non-smooth distributed optimization in networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2740–2749.
[38] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5912–5928, Nov. 2019.
[39] K. Tsianos, S. Lawlor, and M. Rabbat, "Communication/computation tradeoffs in consensus-based distributed optimization," in *Proc. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1943–1951.
[40] A. Berahas, R. Bollapragada, N. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3141–3155, Aug. 2019.
[41] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," in *Math. Program.*, pp. 1–48, Dec. 2018.
[42] A. Nedic, A. Olshevsky, and M. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," in *Proc. IEEE*, vol. 106, no. 5, 2018, pp. 953–976.
[43] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7652–7662.
[44] D. Dimarogonas, E. Frazzoli, and K. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
[45] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *Int. J. Control*, vol. 86, no. 9, pp. 1479–1488, 2013.
[46] C. Nowzari and J. Cortes, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.
[47] Q. Lu and H. Li, "Event-triggered discrete-time distributed consensus optimization over time-varying graphs," *Complexity*, vol. 2017, 2017, Art. no. 5385708.
[48] K. Tsianos, S. Lawlor, J. Yu, and M. Rabbat, "Networked optimization with adaptive communication," in *Proc. Global Conf. Signal Inf. Process.*, 2013, pp. 579–582.
[49] W. Chen and W. Ren, "Event-triggered zero-gradient-sum distributed consensus optimization over directed networks," *Automatica*, vol. 65, pp. 90–97, 2016.
[50] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "COCA: Communication-censored ADMM for decentralized consensus optimization," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 33–37.

Weiyu Li is currently working toward the B.S. degree in mathematics and applied mathematics with the School of Gifted Young, University of Science and Technology of China, Hefei, China. Her research interests include optimization and high-dimensional statistics.

Yaohua Liu received the B.E. degree in automation from the South China University of Technology, Guangzhou, China, in 2015. She is currently working toward the Ph.D. degree with the Department of Automation, University of Science and Technology of China, Hefei, China. Her current research focuses on distributed large-scale optimization.

Zhi Tian received the B.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1994, the M.S. and Ph.D. degrees from George Mason University, Fairfax, VA, USA, in 1998 and 2000, respectively. From 2000 to 2014, she was on the faculty of Michigan Technological University, where she was promoted to a Full Professor in 2011. From 2011 to 2014, she was a Program Director at the U.S. National Science Foundation through an IPA assignment with Michigan Technological University. Since 2015, she has been with the Electrical and Computer Engineering Department, George Mason University, as a Professor. Her general interests lie in the areas of signal processing, wireless communications, and estimation and detection theory. Her current research focuses on compressed sensing for random processes, statistical inference of network data, cognitive radio, and distributed wireless sensor networks. She was an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON SIGNAL PROCESSING. She is a Distinguished Lecturer of the IEEE Vehicular Technology Society and the IEEE Communications Society. She received a CAREER Award from the U.S. National Science Foundation in 2003.

Qing Ling received the B.E. degree in automation and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA, from 2006 to 2009 and an Associate Professor with the Department of Automation, University of Science and Technology of China, from 2009 to 2017. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include decentralized network optimization and its applications. He was the recipient of the 2017 IEEE Signal Processing Society Young Author Best Paper Award as a Supervisor. He is an Associate Editor for the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and IEEE SIGNAL PROCESSING LETTERS.