Consensus Optimization with Delayed and Stochastic Gradients on Decentralized Networks

Benjamin Sirb
Xiaojing Ye
Department of Mathematics and Statistics
Georgia State University
Atlanta, GA 30309, USA
Email: bsirb1@student.gsu.edu
Email: xye@gsu.edu

***Abstract*—We analyze the convergence of a decentralized consensus algorithm with delayed gradient information across the network. The nodes in the network privately hold parts of the objective function and collaboratively solve for the consensus optimal solution of the total objective while they can only communicate with their immediate neighbors. In real-world networks, it is often difficult and sometimes impossible to synchronize the nodes, and therefore they have to use stale gradient information during computations. We show that, as long as the random delays are bounded in expectation and a proper diminishing step size policy is employed, the iterates generated by decentralized gradient descent method still converge to a consensual optimal solution. Convergence rates of both objective and consensus are derived. Numerical results on a variety of consensus optimization problems are presented to show the performance of the method.**

***Index Terms*—Decentralized consensus; delayed gradient; stochastic gradient; decentralized networks.**

I. Introduction

In this paper, we consider a decentralized consensus optimization problem arising from emerging technologies such as big data analysis [1], distributed machine learning [2], sensor networks [3], and smart grids [4]. Decentralized optimization is particularly useful due to the proliferation in recent years of very large data and parameter sizes arising from text and imaging problems [5], leading to optimization problems which are too large to process at a single centralized node.

Given a network $G(V, E)$, where $V = \{1, 2, \ldots, m\}$ is the node (also called agent, processor, or sensor) set and $E \subset V \times V$ the edge set. Two nodes i and j are called neighbors if $(i, j) \in E$. The communications between neighbor nodes are bidirectional, meaning that i and j can communicate with each other as long as $(i, j) \in E$. In a decentralized sensor network G, individual nodes can acquire, store, and process data about large-sized objects. Each node i collects data and holds objective function $F_i(x; \xi_i)$ privately where $\xi_i \in \Theta$ is random with fixed but unknown probability distribution in domain Θ to model environmental fluctuations such as noise in data acquisition. Here $x \in X$ is the unknown to be solved, where the domain $X \subset \mathbb{R}^n$ is compact and convex. Furthermore, we assume that $F_i(\cdot; \xi_i)$ is convex for all $\xi_i \in \Theta$ and $i \in V$, and define $f_i(x) = \mathbb{E}_{\xi_i}[F_i(x; \xi_i)]$ which is convex with respect to $x \in X$. Now the goal of decentralized consensus optimization is to solve the minimization problem

$$\underset{x \in X}{\text{minimize}}\, f(x), \quad \text{where } f(x) := \sum_{i=1}^{m} f_i(x) \tag{1}$$

with the restrictions that $F_i(x; \xi_i)$, and hence $f_i(x)$, are accessible by node i only, and that nodes i and j can communicate only if $(i, j) \in E$ during the entire computation.

There are a number of practical issues that need to be taken into consideration in solving real-world decentralized consensus optimization problem (1):

- The partial objective F_i (and f_i) is held privately by node i, and transferring F_i to a data fusion center is either infeasible or cost-ineffective due to data privacy, the large size of F_i, and/or limited bandwidth and communication power overhead of sensors. Therefore, the nodes can only communicate their own estimates of $x \in \mathbb{R}^n$ with their neighbors in each iteration of a decentralized consensus algorithm.
- Since it is often difficult and sometimes impossible for the nodes to be fully synchronized, they may not have access to the most up-to-date (stochastic) gradient information during computations. In this case, the node i has to use out-of-date (stochastic) gradient $\nabla F_i(x_i(t - \tau_i(t)); \xi_i(t - \tau_i(t)))$ where $x_i(t)$ is the estimate of x obtained by node i at iteration t, and $\tau_i(t)$ is the level of (possibly random) delay of the gradient information at t.
- The estimates $\{x_i(t)\}$ by the nodes should tend to be consensual as t increases, and the consensual value is a solution of problem (1). In this case, there is a guarantee of retrieving a good estimate of x from any surviving node in the network in the event that some nodes are sabotaged, lost, or run out of power during the computation process.

In this paper, we analyze a decentralized consensus algorithm which takes all the factors above into considerations in solving (1). We provide comprehensive convergence analysis of the algorithm, including the decay rates of objective function and disagreements between nodes, in terms of iteration number, level of delays, and network structure etc.

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

A. Related work

Distributed computing on networks is an emerging technology with extensive applications in big data analysis [1] and modern machine learning [6]. There are two types of scenarios in distributed computing: centralized and decentralized. In centralized scenario, computations are carried out locally by worker (slave) nodes while computations of certain global variable must eventually be processed by a designated master node, or at a center of shared memory during each (outer) iteration. A major effort in this scenario has been devoted to update the global variable more effectively using an asynchronous setting in, for example, distributed centralized alternating direction method of multipliers (ADMM) [7], [8]. In the decentralized scenario considered in this paper, the nodes privately hold parts of objective functions and can only communicate with neighbor nodes during computations. In many real-world applications, decentralized computing is particularly useful when a master-worker network setting is either infeasible or not economical, or the data acquisition and computation have to be carried out by individual nodes which then need to collaboratively solve the optimization problem. A decentralized network is also more robust to node failure and can better address privacy concerns. For more discussions about the motivations and advantages of decentralized computing, see, e.g., [9]–[11] and the references therein.

Decentralized consensus algorithms take the data distribution and communication restriction into consideration, so that they can be implemented at individual nodes in the network. A number of developments have been made in the *ideal synchronous case* of decentralized consensus, where all the nodes are coordinated to finish computation and then start to exchange information with neighbors in each iteration. A class of methods is to rewrite the consensus constraints for minimization problem (1) by introducing auxiliary variables between neighbor nodes (i.e., edges), and apply ADMM (possibly with linearization or preconditioning techniques) to derive an implementable decentralized consensus algorithm [12]. Most of these methods require each node to solve a local optimization problem at every iteration before communication, and reach a convergence rate of $O(1/T)$ in terms of outer iteration (communication) number T for general convex and smooth (continuously differentiable) objective functions $\{f_i\}$. First-order methods based on decentralized gradient descent require less computational costs at individual nodes such that between two communications they only perform one step of a gradient descent-type update at the weighted average of previous iterates obtained from neighbors. In particular, Nesterov's optimal gradient scheme is employed in decentralized gradient descent with diminishing step sizes to achieve rate of $O(1/T)$ in [9], where an alternative gradient method that requires excessive communications in each inner iteration is also developed and can reach a theoretical convergence rate of $O(\log T/T^2)$, despite the fact that it seems to work less efficiently in terms of communications than the former in practice. A correction technique is developed for decentralized gradient descent with convergence rate of $O(1/T)$ with constant step size in [11], which results in a saddle-point algorithm as pointed out in [13]. In [14], the authors combine Nesterov's gradient scheme and a multiplier-type auxiliary variable to obtain a fast optimality convergence rate of $O(1/T^2)$. Other first-order decentralized methods have also been developed recently, such as dual averaging [15]. Additional constraints for primal variables in decentralized consensus optimization (1) are considered in [16].

In real-world decentralized computing, it is often difficult and sometimes impossible to coordinate all the nodes in the network such that their computation and communication are perfectly synchronized. One practical approach for such *asynchronous consensus* is using a broadcast scenario where in each (outer) iteration one node in the network is assumed to wake up at random and broadcasts its value to neighbors (but does not hear them back). A number of algorithms for broadcast consensus are developed, for instance, in [17]. Another important issue in the asynchronous setting is that the nodes may have to use out-of-date (stale) gradient information during updates [10]. This delayed scenario in gradient descent is considered in distributed but not decentralized setting in [18]. In addition, analysis of stochastic gradients in distributed computing is also carried out in [18]. In [19], a linear convergence rate of optimality is derived for strongly convex objective functions with delays. Extending [18], a *fixed* delay at all nodes is considered in dual averaging [20] and gradient descent [21] in a decentralized setting, but they did not consider more practical and useful *random* delays, and there are no convergence rates on node consensus provided in these papers.

B. Contributions

The contribution of this paper is in three phases.

First, we consider a general decentralized consensus algorithm with randomly delayed and stochastic gradients (Section II). In this case, the nodes do not need to be synchronized and they may only have access to stale gradient information. This renders stochastic gradients with random delays at different nodes in their gradient updates, which is suitable for many real-world decentralized computing applications such as machine learning and big data analysis.

Second, we provide a comprehensive convergence analysis of the proposed algorithm (Section III). More precisely, we derive convergence rates for both objective function (optimality) and disagreement (feasibility constraint of consensus), and show their dependency on the characteristics of the problem, such as Lipschitz constants of (stochastic) gradients and spectral gaps of the underlying network.

Third, we conduct a number of numerical experiments on various types of datasets to validate the performance of the proposed algorithm (Section IV). In particular, we examine the convergence on synthetic decentralized least squares, robust least squares (Huber loss), and logistic loss functions.

C. Notations and assumptions

In this paper, all vectors are column vectors unless otherwise noted. We denote by $x_i(t) \in \mathbb{R}^n$ the estimate of node i at

iteration t, and $x(t) = (x_1(t), \ldots, x_m(t))^T \in \mathbb{R}^{m\times n}$. We denote $\|x\| \equiv \|x\|_2$ if x is a vector and $\|x\| \equiv \|x\|_F$ if x is a matrix, which should be clear in the context. For any two vectors of same dimension, $\langle x, y\rangle$ denotes their inner product, and $\langle x, y\rangle_Q := \langle x, Qy\rangle$ for symmetric nonnegative definite matrix Q. For notation simplicity, we use $\langle x, y\rangle = \sum_{i=1}^m \langle x_i, y_i\rangle$ where x_i and y_i are the i-th row of the $m \times n$ matrices x and y respectively. Such matrix inner product is also generalized to $\langle x, y\rangle_Q$ for matrices x and y. In this paper, we set the domain $X := \{x \in \mathbb{R}^n : \|x\|_\infty \le R\}$ for some $R > 0$,. We further denote $\mathcal{X} := X^m \subset \mathbb{R}^{m\times n}$.

For each node i, we define $f_i(x) := \mathbb{E}_{\xi_i}[F_i(x; \xi_i)]$ as the expectation of objective function, and $g_i(t) := \nabla F_i(x(t); \xi_i(t))$ (here gradient ∇ is taken with respect to x) is the stochastic gradient at $x_i(t)$ for node i. We let $\tau_i(t)$ be the delay of gradient at node i in iteration t, and $\tau(t) = (\tau_1(t), \ldots, \tau_m(t))^T$. We write $f(x(t))$ in short for $\sum_{i=1}^m f_i(x_i(t)) \in \mathbb{R}$, $x(t - \tau(t))$ for $(x_1(t - \tau_1(t)), \ldots, x_m(t - \tau_m(t)))^T \in \mathbb{R}^{m\times n}$. and $g(t - \tau(t))$ for $(g_1(t - \tau_1(t)), \ldots, g_m(t - \tau_m(t)))^T \in \mathbb{R}^{m\times n}$. We assume f_i is continuously differentiable and ∇f_i has Lipschitz constant L_i, and denote $L := \max_{1\le i\le m} L_i$. Let $x^* \in \mathbb{R}^n$ be a solution of (1). Since x^* is consensual, we denote $\mathbf{1}(x^*)^T$ simply by x^* in this paper which is clear in the context, for instance $f(x^*) = f(\mathbf{1}(x^*)^T) = \sum_{i=1}^m f_i(x^*)$. Furthermore, we let $y(T) := (1/T)\sum_{t=1}^T x(t + 1)$ be the running average of $\{x(t + 1) : 1 \le t \le T\}$, and $z(T) := (1/m)\sum_{i=1}^m y(T)$ be the consensus average of $y(T)$. We denote $J = (1/m)\mathbf{1}\mathbf{1}^T$, so $z(T) = Jy(T)$. Note that for all T, $z(T)$ is always consensual but $x(T)$ and $y(T)$ may not be.

Suppose $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function, then for any $x, y \in \mathbb{R}^n$ we denote the Bregman distance (divergence) between x and y (order matters) by

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y\rangle. \tag{2}$$

If in addition ∇h is L_h-Lipschitz continuous, then we can verify that for any $x, y, z, w \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\langle \nabla h(z) - \nabla h(w), x - y\rangle \\
&= D_h(y, z) - D_h(x, z) - D_h(y, w) + D_h(x, w) \qquad (3)\\
&\le D_h(y, z) - D_h(x, z) + \frac{L_h}{2}\|x - w\|^2 \qquad (4)
\end{aligned}$$

where we used the facts that $D_h(y, w) \ge 0$ and $D_h(x, w) \le \frac{L_h}{2}\|x - w\|^2$.

An important ingredient in decentralized gradient descent is the mixing matrix $W = [w_{ij}]$ in (5). For the algorithm to be implementable in practice, $w_{ij} > 0$ if and only if $(i, j) \in E$. In this paper, we assume that W is symmetric and $\sum_{j=1}^m w_{ij} = 1$ for all i, and hence W is doubly stochastic, namely $W\mathbf{1} = \mathbf{1}W = \mathbf{1}$ where $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^m$. With the assumption that the network G is simple and connected, we know $\|W\|_2 = 1$ and eigenvalue 1 of W has multiplicity 1 by the Perron-Frobenius theorem. As a consequence, $Wx = x$ if and only if x is consensual, i.e., $x = c\mathbf{1}$ for some $c \in \mathbb{R}$. We further assume $W \succeq 0$ (otherwise use $\frac{1}{2}(I + W) \succeq 0$ since stochastic matrix W has spectrum radius 1). Given a network G, there are different ways to design the mixing matrix W. For some optimal choices of W, see, e.g., [22], [23].

Now we make several mild assumptions that are necessary in our convergence analysis.

1) The network $G(V, E)$ is undirected, simple, and connected.
2) The stochastic gradient satisfies $\mathbb{E}_{\xi_i}[\nabla F_i(x; \xi_i)] = \nabla f_i(x)$ for all i and x. Moreover, for all i, ξ, and x, $\|\nabla f_i\|, \mathbb{E}_{\xi_i}[\|\nabla F_i(x; \xi_i)\|^2] \le G^2$ and for some $G > 0$, and $\mathbb{E}_{\xi_i}[\|\nabla F_i(x; \xi_i) - \nabla f_i(x)\|^2] \le \sigma^2$ for some $\sigma > 0$.
3) The delays $\tau_i(t)$ may follow different distributions at different nodes, but their second moments are assumed to be uniformly bounded, i.e., there exists $B > 0$ such that $\mathbb{E}[|\tau_i(t)|^2] \le B^2$ for all $i = 1, \ldots, m$ and iterations t. For each node i, we assume each update happens once, i.e., $t \mapsto t - \tau_i(t)$ is strictly increasing as t increases.

It is worth pointing out that these assumptions are rather standard and easy to satisfy in practice. For instance, the boundedness of ∇f_i is a consequence of the compactness of domain X and the Lipschitz continuity of ∇f_i. The assumption on random delays in a distributed system is also used in [18]. We further assume that the the stochastic error ξ_i and the random delay τ_i are independent.

II. Algorithm

Taking the delayed stochastic gradient and the constraint that nodes can only communicate with immediate neighbors, we propose the following decentralized delayed stochastic gradient descent method for solving (1). Starting from an initial guess $\{x_i(0) : i = 1, \ldots, m\}$, each node i performs the following updates iteratively:

$$x_i(t + 1) = \Pi_X \left[\sum_{j=1}^m w_{ij}x_j(t) - \alpha(t)g_i(t - \tau_i(t))\right]. \tag{5}$$

Namely, in each iteration t, the nodes exchange their most recent $x_i(t)$ with their neighbors. Then each node takes weighted average of the received local copies using weights w_{ij}, performs a gradient descent type update using a delayed stochastic gradient $g_i(t-\tau_i(t))$ with step size $\alpha(t)$, and projects the result onto X.

Following the matrix notation in Section I-C, the iteration (5) can be written as

$$x(t + 1) = \Pi_{\mathcal{X}}[Wx(t) - \alpha(t)g(t - \tau(t))]. \tag{6}$$

Here the projection $\Pi_{\mathcal{X}}$ is accomplished by each node projecting to X due to the definition of X in Section I-C, which does not require any coordinations between nodes. Note that the update (6) is also equivalent to

$$\begin{aligned}
&x(t + 1)\\
&= \operatorname*{argmin}_{x\in\mathcal{X}} \left\{\langle g(t - \tau(t)), x\rangle + \frac{1}{2\alpha(t)}\|x - Wx(t)\|^2\right\}. \qquad (7)
\end{aligned}$$

In this paper, we may refer to the proposed decentralized delayed stochastic gradient descent algorithm by any of (5), (6), and (7) since they are equivalent.

III. Convergence analysis

In this section, we provide a comprehensive convergence analysis of the proposed algorithm (7) by employing proper step size policy. In particular, we derive convergence rates for the objective function and disagreement in order.

Lemma 1. *Let $\{x(t)\}$ be the iterates generated by Algorithm* (6)*, then the following inequality holds for all $T \geq 1$:*

$$\sum_{t=1}^{T} \langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \rangle \tag{8}$$
$$\leq 2mnLR^2(1+2B^2) + \frac{L}{2}(B+1)^2 \sum_{t=1}^{T} \|x(t+1) - x(t)\|^2.$$

Proof. We first observe that

$$\begin{aligned} &\langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \rangle \\ &= \sum_{i=1}^{m} \langle \nabla f_i(x_i(t)) - \nabla f_i(x_i(t-\tau_i(t))), x_i(t+1) - x^* \rangle \\ &\leq \sum_{i=1}^{m} \Bigg[D_{f_i}(x^*, x_i(t)) - D_{f_i}(x^*, x_i(t-\tau_i(t))) \\ &\quad + \frac{L}{2} \|x_i(t+1) - x_i(t-\tau_i(t))\|^2 \Bigg] \end{aligned} \tag{9}$$

where we applied (3) to get the inequality. We further note that the convexity of $\|\cdot\|^2$ implies

$$\begin{aligned} &\|x_i(t+1) - x_i(t-\tau_i(t))\|^2 \\ &\leq (\tau_i(t)+1) \sum_{s=0}^{\tau_i(t)} \|x_i(t-s+1) - x_i(t-s)\|^2. \end{aligned} \tag{10}$$

Combining (9) and (10), and taking the sum of t from 1 to T, we obtain

$$\begin{aligned} &\sum_{t=1}^{T} \langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \rangle \\ &\leq \sum_{i=1}^{m} \Bigg[\sum_{t=1}^{T} (D_{f_i}(x^*, x_i(t)) - D_{f_i}(x^*, x_i(t-\tau_i(t)))) \\ &\quad + \frac{L}{2} \sum_{t=1}^{T} (\tau_i(t)+1) \sum_{s=0}^{\tau_i(t)} \|x_i(t-s+1) - x_i(t-s)\|^2 \Bigg]. \end{aligned} \tag{11}$$

For each i, the sum of D_{f_i} terms for t from 1 to T above leaves only those not received by the gradient procedure within T iterations, namely

$$\begin{aligned} &\sum_{t=1}^{T} (D_{f_i}(x^*, x_i(t)) - D_{f_i}(x^*, x_i(t-\tau_i(t)))) \\ &= \sum_{t \in \mathcal{S}_i(T)} D_{f_i}(x^*, x_i(t)) \end{aligned} \tag{12}$$

where $\mathcal{S}_i(T) := \{1 \leq t \leq T : t > T - \tau_i(T)\}$. Then by Chebyshev's inequality, we can bound the expected cardinality of $\mathcal{S}_i(T)$ by

$$\begin{aligned} \mathbb{E}[|\mathcal{S}_i(T)|] &= \sum_{t=1}^{T} \mathrm{P}(\tau_i(T) > T - t) \\ &\leq 1 + \sum_{t=1}^{T-1} \frac{B^2}{(T-t)^2} \\ &\leq 1 + 2B^2 \end{aligned} \tag{13}$$

where we used the fact that $\sum_{t=1}^{T-1} \frac{1}{(T-t)^2} = \sum_{t=1}^{T-1} \frac{1}{t^2} \leq 2 - \frac{1}{T-1} \leq 2$. Combining (12) and (13), and using the fact that $D_{f_i}(x^*, x_i(t)) \leq 2nLR^2$ for all t and i, we obtain,

$$\begin{aligned} &\sum_{i=1}^{m} \sum_{t=1}^{T} (D_{f_i}(x^*, x_i(t)) - D_{f_i}(x^*, x_i(t-\tau_i(t)))) \\ &\leq 2mnLR^2(1+2B^2). \end{aligned} \tag{14}$$

For each i, the second sum of t from 1 to T on the right side of (11) yields

$$\begin{aligned} &\sum_{t=1}^{T} (\tau_i(t)+1) \sum_{s=0}^{\tau_i(t)} \|x_i(t-s+1) - x_i(t-s)\|^2 \\ &\leq \sum_{t=1}^{T} N_i(t,T) \|x_i(t+1) - x_i(t)\|^2 \end{aligned} \tag{15}$$

where the coefficient $N_i(t,T)$ is defined by

$$N_i(t,T) := \sum_{s \in Y} (\tau_i(s)+1) \tag{16}$$

with $Y = \{t \leq s \leq T : 0 \leq s - \tau_i(s) \leq t\}$. Therefore, we have for each i that

$$\begin{aligned} &\mathbb{E}[N_i(t,T)] = \\ &\mathbb{E}\left[\sum_{s \in Y} (\tau_i(s)+1) \right] \\ &= \sum_{s=t}^{T} \sum_{k=s-t}^{s} (k+1)\, \mathrm{P}(\tau_i(s) = k) \\ &\leq \sum_{k=0}^{T} (k+1)^2\, \mathrm{P}(\tau_i(s) = k) \\ &\leq \mathbb{E}[|\tau_i(s)+1|^2] \\ &\leq (B+1)^2 \end{aligned} \tag{17}$$

where the first inequality is obtained by listing each possible value of k in the double sum, and upper bounding its occurrence by $(k+1)$, and the last inequality is due to $\mathbb{E}[|\tau_i(s)|] \leq \sqrt{\mathbb{E}[|\tau_i(s)|^2]} = B$. Therefore, (15) can be bounded by

$$\begin{aligned} &\sum_{i=1}^{m} \sum_{t=1}^{T} (\tau_i(t)+1) \sum_{s=0}^{\tau_i(t)} \|x_i(t-s+1) - x_i(t-s)\|^2 \\ &\leq (B+1)^2 \sum_{t=1}^{T} \|x(t+1) - x(t)\|^2. \end{aligned} \tag{18}$$

Applying (14) and (18) to (11) completes the proof. □

Theorem 2. *Let $\{x(t)\}$ be the iterations generated by Algorithm* (6) *with $\alpha(t) = [2\,(L+\eta(t))]^{-1}$ where $\eta(t)$ is a nondecreasing function of t, then*

$$\begin{aligned}
&\mathbb{E}[f(y(T)) - f(x^*)] \\
&\le \frac{2mnR^2[4L + 2\eta(1) + 2\eta(T) + L(1+2B^2)]}{T} \\
&\quad + \frac{2m\sigma^2}{T}\sum_{t=1}^{T}\frac{1}{\eta(t)} \\
&\quad + \frac{L(B+1)^2}{2T}\sum_{t=1}^{T}\mathbb{E}[\|x(t+1) - x(t)\|^2] \qquad (19)
\end{aligned}$$

where $y(T) = (1/T)\sum_{t=1}^{T} x(t+1)$ is the running average of $\{x(t)\}$.

Proof. We first note that there is

$$\begin{aligned}
&f(x(t+1)) - f(x^*) = \sum_{i=1}^{m}\left(f_i(x_i(t+1)) - f_i(x^*)\right) \\
&= \sum_{i=1}^{m}\left[f_i(x_i(t+1)) - f_i(x_i(t)) + f_i(x_i(t)) - f_i(x^*)\right] \\
&\le \sum_{i=1}^{m}\langle \nabla f_i(x_i(t)), x_i(t+1) - x_i(t)\rangle \\
&\quad + \sum_{i=1}^{m}\frac{L_i}{2}\|x_i(t+1) - x_i(t)\|^2 \\
&\quad + \sum_{i=1}^{m}\langle \nabla f_i(x_i(t)), x_i(t) - x^*\rangle \qquad (20) \\
&\le \sum_{i=1}^{m}\langle \nabla f_i(x_i(t)), x_i(t+1) - x^*\rangle \\
&\quad + \sum_{i=1}^{m}\frac{L_i}{2}\|x_i(t+1) - x_i(t)\|^2 \\
&\le \langle \nabla f(x(t)), x(t+1) - x^*\rangle + \frac{L}{2}\|x(t+1) - x(t)\|^2 \\
&\le \langle g(t-\tau(t)), x(t+1) - x^*\rangle \\
&\quad + \langle \nabla f(x(t)) - g(t-\tau(t)), x(t+1) - x^*\rangle \\
&\quad + \frac{L}{2}\|x(t+1) - x(t)\|^2
\end{aligned}$$

where we used the L_i-Lipschitz continuity of ∇f_i and convexity of f_i to obtain the first inequality. Note that $x(t+1)$ is obtained by (7) as

$$\begin{aligned}
&x(t+1) \\
&= \operatorname*{argmin}_{x\in\mathcal{X}}\left\{\langle g(t-\tau(t)), x\rangle + \frac{1}{2\alpha(t)}\|x - Wx(t)\|^2\right\} \qquad (21) \\
&= \operatorname*{argmin}_{x\in\mathcal{X}}\left\{\left\langle g(t-\tau(t)) + \frac{1}{\alpha(t)}(I-W)x(t), x\right\rangle \right. \\
&\qquad\qquad \left. + \frac{1}{2\alpha(t)}\|x - x(t)\|^2\right\}.
\end{aligned}$$

Therefore, the optimality of $x(t+1)$ in (7) implies that

$$\begin{aligned}
&\langle g(t-\tau(t)), x(t+1) - x^*\rangle \\
&\le -\frac{1}{\alpha(t)}\langle (I-W)x(t), x(t+1) - x^*\rangle \\
&\quad + \frac{1}{2\alpha(t)}\Big[\|x^* - x(t)\|^2 - \|x(t+1) - x(t)\|^2 \\
&\qquad\qquad - \|x^* - x(t+1)\|^2\Big]. \qquad (22)
\end{aligned}$$

Furthermore, we note that $\mathbf{1} \in \mathrm{Null}(I-W)$ and x^* is consensual, hence we have

$$\begin{aligned}
&-\frac{1}{\alpha(t)}\langle (I-W)x(t), x(t+1) - x^*\rangle \\
&= -\frac{1}{\alpha(t)}\langle (I-W)(x(t) - x^*), x(t+1) - x^*\rangle \\
&= \frac{1}{2\alpha(t)}\Big(\|x(t) - x(t+1)\|_{I-W}^2 - \|x(t) - x^*\|_{I-W}^2 \\
&\qquad - \|x(t+1) - x^*\|_{I-W}^2\Big) \qquad (23) \\
&\le \frac{1}{4\alpha(t)}\|x(t) - x(t+1)\|_{I-W}^2
\end{aligned}$$

where we have used the fact that

$$\begin{aligned}
&\|x(t) - x(t+1)\|_{I-W}^2 \\
&\le 2\Big(\|x(t) - x^*\|_{I-W}^2 + \|x(t+1) - x^*\|_{I-W}^2\Big)
\end{aligned}$$

to obtain the inequality above. We also have that

$$\|x(t) - x(t+1)\|_{I-W}^2 \le \|x(t) - x(t+1)\|^2$$

with which we can further bound (23) as

$$\begin{aligned}
&-\frac{1}{\alpha(t)}\langle (I-W)x(t), x(t+1) - x^*\rangle \\
&\le \frac{1}{4\alpha(t)}\|x(t) - x(t+1)\|^2.
\end{aligned}$$

Now applying the inequality above and (22) to (20), and summing t from 1 to T, we get

$$\begin{aligned}
&\sum_{t=1}^{T} f(x(t+1)) - Tf(x^*) \qquad (24) \\
&\le \sum_{t=1}^{T}\left[\frac{1}{2\alpha(t)}\Big(\|x(t) - x^*\|^2 - \|x(t+1) - x^*\|^2\Big)\right. \\
&\qquad \left. + \Big(\frac{L}{2} - \frac{1}{4\alpha(t)}\Big)\|x(t) - x(t+1)\|^2\right] \\
&\qquad + \sum_{t=1}^{T}\langle \nabla f(x(t)) - g(t-\tau(t)), x(t+1) - x^*\rangle\,.
\end{aligned}$$

For the last term on the right hand side of (24), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\langle\nabla f(x(t))-g(t-\tau(t)),x(t+1)-x^*\rangle\\
&=\sum_{t=1}^{T}\langle\nabla f(x(t))-\nabla f(x(t-\tau(t))),x(t+1)-x^*\rangle\\
&\quad+\sum_{t=1}^{T}\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x^*\rangle
\end{aligned}\tag{25}$$

$$\begin{aligned}
&\le 2mnLR^2(1+2B^2)\\
&\quad+\sum_{t=1}^{T}\frac{L}{2}(B+1)^2\|x(t+1)-x(t)\|^2\\
&\quad+\sum_{t=1}^{T}\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x^*\rangle
\end{aligned}$$

where we applied the Lemma 1 to obtain the inequality.

Note that the running average $y(T)=(1/T)\sum_{t=1}^{T}x(t+1)$ satisfies $f(y(T))\le\sum_{t=1}^{T}f(x(t+1))$ due to the convexity of all f_i. Therefore, together with (24) and (25) and the definition of $\alpha(t)$, we have

$$\begin{aligned}
&T(f(y(T))-f(x^*))\\
&\le\sum_{t=1}^{T}\left[\frac{1}{2\alpha(t)}\Big(\|x(t)-x^*\|^2-\|x(t+1)-x^*\|^2\Big)\right.\\
&\quad\left.+\frac{L(B+1)^2-\eta(t)}{2}\|x(t)-x(t+1)\|^2\right]\\
&\quad+2mnLR^2(1+2B^2)\\
&\quad+\sum_{t=1}^{T}\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x^*\rangle.
\end{aligned}\tag{26}$$

Now, by taking expectation on both sides of (26), we obtain

$$\begin{aligned}
&T\,\mathbb{E}[f(y(T))-f(x^*)]\\
&\le\sum_{t=1}^{T}\left[\frac{1}{2\alpha(t)}\Big(e(t)-e(t+1)\Big)\right.\\
&\quad\left.+\frac{L(B+1)^2-\eta(t)}{2}\,\mathbb{E}[\|x(t)-x(t+1)\|^2]\right]\\
&\quad+2mnLR^2(1+2B^2)\\
&\quad+\sum_{t=1}^{T}\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x^*\rangle
\end{aligned}\tag{27}$$

where we denoted $e(t):=\mathbb{E}[\|x(t)-x^*\|^2]$ for simplicity of notation.

Now we work on the last sum of inner products on the right side of (27). First we observe that

$$\begin{aligned}
&\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x^*\rangle\\
&=\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t)-x^*\rangle\\
&\quad+\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x(t)\rangle.
\end{aligned}\tag{28}$$

Note that $g(t-\tau(t))$ is used to calculate $x(t+1)$, and hence its stochastic error $g(t-\tau(t))-\nabla f(x(t-\tau(t)))$ is independent of $x(t)$. Therefore, we have

$$\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t)-x^*\rangle=0.\tag{29}$$

Furthermore, by Young's inequality, we have

$$\begin{aligned}
&\mathbb{E}\,\langle\nabla f(x(t-\tau(t)))-g(t-\tau(t)),x(t+1)-x(t)\rangle\\
&\le\frac{2}{\eta(t)}\,\mathbb{E}[\|\nabla f(x(t-\tau(t)))-g(t-\tau(t))\|^2]\\
&\quad+\frac{\eta(t)}{2}\,\mathbb{E}[\|x(t+1)-x(t)\|^2]\\
&\le\frac{2m\sigma^2}{\eta(t)}+\frac{\eta(t)}{2}\,\mathbb{E}[\|x(t+1)-x(t)\|^2]
\end{aligned}\tag{30}$$

where we used the fact that $\mathbb{E}[\|\nabla f(x(t-\tau(t)))-g(t-\tau(t))\|^2]\le m\sigma^2$ for all t. Now applying (28), (29) and (30) in (27), we have

$$\begin{aligned}
&T\,\mathbb{E}\,[f(y(T))-f(x^*)]\\
&\le\sum_{t=1}^{T}\frac{1}{2\alpha(t)}\,(e(t)-e(t+1))\\
&\quad+2mnLR^2(1+2B^2)+\sum_{t=1}^{T}\frac{2m\sigma^2}{\eta(t)}\\
&\quad+\frac{L(B+1)^2}{2}\sum_{t=1}^{T}\mathbb{E}[\|x(t+1)-x(t)\|^2]\\
&\le\frac{e(1)}{2\alpha(1)}+\sum_{t=2}^{T}\frac{e(t)}{2}\left(\frac{1}{\alpha(t)}-\frac{1}{\alpha(t-1)}\right)\\
&\quad+2mnLR^2(1+2B^2)+\sum_{t=1}^{T}\frac{2m\sigma^2}{\eta(t)}\\
&\quad+\frac{L(B+1)^2}{2}\sum_{t=1}^{T}\mathbb{E}[\|x(t+1)-x(t)\|^2].
\end{aligned}\tag{31}$$

Note that $\alpha(t)$ is nonincreasing, therefore $\frac{1}{\alpha(t)}-\frac{1}{\alpha(t-1)}\ge 0$ and hence

$$\begin{aligned}
&\sum_{t=2}^{T}\frac{e(t)}{2}\left(\frac{1}{\alpha(t)}-\frac{1}{\alpha(t-1)}\right)\\
&\le 2mnR^2\sum_{t=2}^{T}\left(\frac{1}{\alpha(t)}-\frac{1}{\alpha(t-1)}\right)\\
&\le\frac{2mnR^2}{\alpha(T)}
\end{aligned}\tag{32}$$

where we used the fact that $e(t)=\mathbb{E}[\|x(t)-x^*\|^2]\le 4mnR^2$ for all t. Applying (32) to (31) yields (19). □

We have shown that the running average $y(T)$ makes the objective function decay as in (19). However, an important feature in decentralized computing is that $x_i(t)$ tends to be consensual. Now we prove that consensus can be achieved by the proposed algorithm (6), and we derive the convergence rate for the employed step size policy.

Next we observe that projection is a non-expansive operation as it bounds the individual components of a vector so their disagreement does not increase. By disagreement we mean

$$\|(I-J)x\|^2 = \sum_{i=1}^{m}(x_i - \overline{x})^2$$

where $\overline{x}$ is the arithmetic mean of the components x_i of x. Due to space limitations, we state the lemma without proof.

Lemma 3. *For any $x \in \mathbb{R}^{m\times n}$, its projection onto $\mathcal{X}$ yields nonincreasing disagreement. That is*

$$\|(I-J)\,\Pi_{\mathcal{X}}(x)\|^2 \le \|(I-J)x\|^2 \tag{33}$$

Lemma 4. *Let $c_1 \ge 0$ and $c_2 > 0$, and define $\alpha(t) = 1/(c_1 + c_2\sqrt{t})$. Then for any $\lambda \in (0,1)$ there is*

$$\sum_{s=0}^{t-1}\alpha(s)\lambda^{t-s-1} \le \frac{\sqrt{\pi}\lambda^{-2}}{c_2\sqrt{t}\log(\lambda^{-1})} = O\left(\frac{1}{\sqrt{t}}\right) \tag{34}$$

for all $t = 1, 2, \ldots$.

Proof. First, we note that

$$\sum_{s=0}^{t-1}\alpha(s)\lambda^{t-1-s} = \alpha(0)\lambda^{t-1} + \alpha(1)\lambda^{t-2} + \sum_{s=2}^{t-1}\alpha(s)\lambda^{t-1-s} \tag{35}$$

which means that the rate is bound from above by the last sum on the right side since the first two tend to 0 at a linear rate $\lambda \in (0,1)$.

Next, note that for all $w \in [s-1, s]$ we have $\frac{1}{\sqrt{s}} \le \frac{1}{\sqrt{w}}$ and $\lambda^{-s} \le \lambda^{-(w+1)}$ since $\lambda \in (0,1)$, and therefore

$$\begin{aligned}\alpha(s)\lambda^{t-1-s} &= \frac{\lambda^{t-1-s}}{c_1 + c_2\sqrt{s}} \le \frac{\lambda^{t-1}\lambda^{-s}}{c_2\sqrt{s}} \\ &\le \frac{\lambda^{t-1}\lambda^{-(w+1)}}{c_2\sqrt{w}} = \frac{\lambda^{t-2-w}}{c_2\sqrt{w}}.\end{aligned} \tag{36}$$

This inequality allows us to bound the last term on right hand side of (35) by

$$\begin{aligned}\sum_{s=2}^{t-1}\alpha(s)\lambda^{t-1-s} &\le \sum_{s=2}^{t-1}\int_{s-1}^{s}\frac{\lambda^{t-2-w}}{c_2\sqrt{w}}dw \\ &= \int_{1}^{t-1}\frac{\lambda^{t-2-w}}{c_2\sqrt{w}}dw \\ &= \frac{\lambda^{t-2}}{c_2}\int_{1}^{t-1}\frac{\lambda^{-w}}{\sqrt{w}}dw.\end{aligned} \tag{37}$$

Now we focus on the value of integral

$$I_t := \frac{1}{2}\int_{1}^{t-1}\frac{\lambda^{-w}}{\sqrt{w}}dw = \int_{1}^{\sqrt{t-1}}\lambda^{-u^2}du \tag{38}$$

where we applied a change of variables $w = u^2$. Note that we have

$$\begin{aligned}I_t^2 &= \int_1^{\sqrt{t-1}}\int_1^{\sqrt{t-1}}\lambda^{-(u^2+v^2)}dudv \\ &\le \int_0^{\sqrt{t}}\int_0^{\sqrt{t}}e^{-(u^2+v^2)\log\lambda}dudv \\ &= 2\int_0^{\pi/4}\int_0^{\sqrt{t}/\cos\theta}e^{-\rho^2\log\lambda}\rho d\rho d\theta \\ &= -\frac{1}{\log\lambda}\int_0^{\pi/4}(e^{-t\log\lambda/\cos^2(\theta)} - 1)d\theta \\ &< -\frac{1}{\log\lambda}\int_0^{\pi/4}e^{-t\log\lambda/\cos^2(\theta)}d\theta\end{aligned} \tag{39}$$

where the third equality comes from changing to a polar system with the substitutions $u = \rho\cos\theta$ and $v = \rho\sin\theta$. Note that $\cos^{-2}(\theta) - (1 + 4\theta/\pi) \le 0$ for all $\theta \in [0, \pi/4]$ since $\cos^{-2}(\theta) - 1 - 4\theta/\pi$ is convex with respect to θ and vanishes at $\theta = 0$ and $\theta = \pi/4$. Therefore

$$I_t^2 \le -\frac{1}{\log\lambda}\int_0^{\pi/4}e^{-t\log\lambda(1+4\theta/\pi)}d\theta \le \frac{\pi\lambda^{-2t}}{4t(\log\lambda)^2}. \tag{40}$$

Hence the sum in (37) is bounded by

$$\begin{aligned}\sum_{s=2}^{t-1}\alpha(s)\lambda^{t-1-s} &\le \frac{2\lambda^{t-2}}{c_2}I_t \le \frac{2\lambda^{t-2}}{c_2}\frac{\sqrt{\pi}\lambda^{-t}}{2\sqrt{t}\log(\lambda^{-1})} \\ &= \frac{\sqrt{\pi}\lambda^{-2}}{c_2\sqrt{t}\log(\lambda^{-1})}\end{aligned} \tag{41}$$

which completes the proof. □

Theorem 5. *Let $\{x(t)\}$ be the iterates generated by Algorithm (7) with $\alpha(t) = [2(L + \eta\sqrt{t})]^{-1}$ for $\eta > 0$, and $\lambda = \|W - J\|$. Then λ is the second largest eigenvalue of W and hence $\lambda \in (0,1)$. Moreover, the disagreement of $x(t)$ is bounded above by*

$$\begin{aligned}&\|(I-J)x(t)\| \\ &\le \sqrt{m}G\sum_{s=0}^{t-1}\alpha(s)\lambda^{t-s-1} \le \frac{\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{t}\log(\lambda^{-1})}\end{aligned} \tag{42}$$

and the disagreement of running average $y(T) = (1/m)\sum_{t=1}^{T}x(t+1)$ is bounded above by

$$\|(I-J)y(T)\| \le \frac{2\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{T}\log(\lambda^{-1})} = O\left(\frac{1}{\sqrt{T}}\right). \tag{43}$$

Proof. We prove this bound by induction. It is trivial to show the bound for $t = 1$. Assume (42) is true for t, then we have

$$\begin{aligned}&\|(I-J)x(t+1)\| \\ &= \|(I-J)\,\Pi_{\mathcal{X}}(Wx(t) - \alpha(t)g(t-\tau(t)))\| &(44)\\ &\le \|(I-J)(Wx(t) - \alpha(t)g(t-\tau(t)))\| &(45)\\ &\le \|(I-J)Wx(t)\| + \alpha(t)\|(I-J)g(t-\tau(t))\| \\ &\le \|(I-J)Wx(t)\| + \alpha(t)\sqrt{m}G\end{aligned}$$

where we used Lemma 3 in the first inequality, and $\|I-J\| \leq 1$ and $\|g_i(t-\tau_i(t))\| \leq G$ in the last inequality. Noting that $J^2 = J$ and $JW = WJ = J$, we have

$$(W-J)(I-J) = (I-J)W.$$

Therefore, we obtain

$$\begin{aligned}
&\|(I-J)x(t+1)\| \\
&\leq \|(I-J)Wx(t)\| + \alpha(t)\sqrt{m}G \\
&= \|(W-J)(I-J)x(t)\| + \alpha(t)\sqrt{m}G \\
&\leq \|(W-J)\|\|(I-J)x(t)\| + \alpha(t)\sqrt{m}G \qquad (46)\\
&\leq \lambda\sqrt{m}G\sum_{s=0}^{t-1}\alpha(s)\lambda^{t-s-1} + \alpha(t)\sqrt{m}G \\
&= \sqrt{m}G\sum_{s=0}^{t}\alpha(s)\lambda^{t-s}
\end{aligned}$$

where we used the induction assumption for t in the last inequality. Applying Lemma 4 yields the second inequality in (42). By convexity of $\|\cdot\|$ and definition of $y(T)$, we have

$$\begin{aligned}
&\|(I-J)y(T)\| \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\|(I-J)x(t+1)\| \\
&\leq \sum_{t=1}^{T}\frac{\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{t}\log(\lambda^{-1})} \leq \frac{2\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{T}\log(\lambda^{-1})} \qquad (47)
\end{aligned}$$

by applying (42) and using $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$. □

Corollary 6. *Let $\{x(t)\}$ be the iterates generated by Algorithm (7) with the settings of $\alpha(t)$, λ, and η same as in Theorem 5. Then there is*

$$\|x(t+1)-x(t)\|^2 \leq \frac{2G^2}{\eta^2 t}\left[\frac{\pi m\lambda^{-4}}{\log^2(\lambda^{-1})}+\frac{1}{4}\right]. \qquad (48)$$

Proof. According to the update (7) or equivalently (6), we have

$$\begin{aligned}
&\|x(t+1)-x(t)\|^2 \\
&= \|\,\Pi_{\mathcal{X}}(Wx(t)-\alpha(t)g(t-\tau(t))) - x(t)\|^2 \\
&\leq \|(I-W)x(t)+\alpha(t)g(t-\tau(t)))\|^2 \qquad (49)\\
&\leq 2\left(\|(I-W)x(t)\|^2 + \|\alpha(t)g(t-\tau(t)))\|^2\right)
\end{aligned}$$

where we used the facts that $x(t) \in \mathcal{X}$ and that projection $\Pi_{\mathcal{X}}$ is nonexpansive. Note that $WJ = J$ and hence $I-W = (I-W)(I-J)$, we have

$$\begin{aligned}
\|(I-W)x(t)\|^2 &= \|(I-W)(I-J)x(t)\|^2 \\
&\leq \|(I-J)x(t)\|^2 \\
&\leq \frac{\pi m G^2\lambda^{-4}}{\eta^2 t\log^2(\lambda^{-1})} \qquad (50)
\end{aligned}$$

where we used the fact that $\|I-W\| \leq 1$ in the first inequality and applied Theorem 5 to obtain the second inequality.

On the other hand, we have by the definition of $\alpha(t)$ that

$$\|\alpha(t)g(t-\tau(t))\|^2 = (\alpha(t))^2G^2 = \frac{G^2}{4(L+\eta\sqrt{t})^2} \leq \frac{G^2}{4\eta^2 t}. \qquad (51)$$

Applying (50) and (51) to (49) yields (48). □

Theorem 7. *Let $x(t)$ be generated by Algorithm (5) with $\alpha(t) = [2(L+\eta\sqrt{t})]^{-1}$ for some $\eta > 0$. Let $y(T) = (1/T)\sum_{t=1}^{T}x(t+1)$ be the running average of $x(t)$ and $z(T) = Jy(T) = (1/m)\sum_{i=1}^{m}y_i(T)$ be the consensus average of $y(T)$, then*

$$\begin{aligned}
&\mathbb{E}[f(z(T))] - f(x^*) \\
&\leq \frac{2\sqrt{\pi}mG^2}{\eta\lambda^2\sqrt{T}\log(\lambda^{-1})} + \frac{2mnR^2(4L+2\eta+L(1+2B^2))}{T} \\
&\quad + \frac{2mnR^2\eta}{\sqrt{T}} + \frac{4m\sigma^2}{\eta\sqrt{T}} \\
&\quad + \frac{2L(B+1)^2G^2(1+\log T)}{\eta^2 T}\left[\frac{\pi m\lambda^{-4}}{\log^2(\lambda^{-1})}+\frac{1}{4}\right]. \qquad (52)
\end{aligned}$$

Proof. We first bound the difference between the function values at the running average $y(T)$ and the consensus average $z(T) = Jy(T)$:

$$\begin{aligned}
&|f(y(T)) - f(z(T))| = \left|\sum_{i=1}^{m}(f_i(y_i(T)) - f_i(z(T)))\right| \\
&\leq \sum_{i=1}^{m}|\langle\nabla f_i(z(T)), y_i(T)-z(T)\rangle| \\
&\leq G\sum_{i=1}^{m}\|y_i(T)-z(T)\| \\
&\leq \sqrt{m}G\|(I-J)y(t)\| \\
&\leq \frac{\sqrt{m}G}{T}\sum_{t=1}^{T}\|(I-J)x(t+1)\| \qquad (53)\\
&\leq \frac{\sqrt{\pi}mG^2}{\eta\lambda^2 T\log(\lambda^{-1})}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq \frac{2\sqrt{\pi}mG^2}{\eta\lambda^2\sqrt{T}\log(\lambda^{-1})}
\end{aligned}$$

where we used convexity of f_i in the first inequality, $\|\nabla f_i\| \leq G$ in the second inequality, the convexity of $\|\cdot\|$ in the fourth inequality, and Theorem 5 to get the fifth inequality. Note that Theorem 2 implies

$$\begin{aligned}
&\mathbb{E}[f(y(T)) - f(x^*)] \\
&\leq \frac{2mnR^2(2L+\eta+L(1+2B^2))}{T} + \frac{2mnR^2\eta}{\sqrt{T}} + \frac{4m\sigma^2}{\eta\sqrt{T}} \\
&\quad + \frac{L(B+1)^2}{2T}\sum_{t=1}^{T}\mathbb{E}[\|x(t+1)-x(t)\|^2], \qquad (54)
\end{aligned}$$

and the last term on the right hand side can be bounded by using Corollary 6:

$$\begin{aligned}
&\frac{L(B+1)^2}{2T}\sum_{t=1}^{T}\mathbb{E}[\|x(t+1)-x(t)\|^2]\\
&\le \frac{2L(B+1)^2G^2}{\eta^2 T}\left[\frac{\pi m\lambda^{-4}}{\log^2(\lambda^{-1})}+\frac{1}{4}\right]\sum_{t=1}^{T}\frac{1}{t}\\
&\le \frac{2L(B+1)^2G^2(1+\log T)}{\eta^2 T}\left[\frac{\pi m\lambda^{-4}}{\log^2(\lambda^{-1})}+\frac{1}{4}\right] \qquad (55)
\end{aligned}$$

where we used the fact that $\sum_{t=1}^{T}(1/t)\le 1+\log T$. Therefore, we obtain

$$\begin{aligned}
&\mathbb{E}[f(z(T))]-f(x^*)\\
&\le \mathbb{E}[|f(z(T))-f(y(T))|]+\mathbb{E}[f(y(T))-f(x^*)]\\
&\le \frac{2\sqrt{\pi}mG^2}{\eta\lambda^2\sqrt{T}\log(\lambda^{-1})}+\frac{2mnR^2(2L+\eta+L(1+2B^2))}{T}\\
&+\frac{2mnR^2\eta}{\sqrt{T}}+\frac{4m\sigma^2}{\eta\sqrt{T}}\\
&+\frac{2L(B+1)^2G^2(1+\log T)}{\eta^2 T}\left[\frac{\pi m\lambda^{-4}}{\log^2(\lambda^{-1})}+\frac{1}{4}\right]. \qquad (56)
\end{aligned}$$

On the other hand, $z(T)$ is consensus, so $f(z(T))\ge f(x^*)$ since x^* is a solution of (1). This completes the proof. □

IV. Numerical experiments

In this section, we test algorithm (5) on decentralized consensus optimization problem (1) with delayed stochastic gradients using a number of synthetic datasets. In particular, we apply algorithm (5) to decentralized least squares, decentralized robust least squares, and decentralized logistic regression problems. The structure of network $G(V,E)$ and objective function in (1) are explained for each dataset, followed by performance evaluation shown in plots of objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m}\|y_i(T)-z(T)\|^2$ versus the iteration number T, where $y_i(T)=(1/T)\sum_{t=1}^{T}x_i(t+1)$ is the running average of $x_i(t)$ in algorithm (5) over t from 1 to T at each node i, and $z(T)=(1/m)\sum_{i=1}^{m}y_i(T)$ is the consensus average of $y_i(T)$ over all nodes at iteration T. For reference, we also show $f^*:=f(x^*)$ in the plots of objective functions, where x^* is the optimal solution computed by MATLAB built-in linear system solvers for the synthetic decentralized least squares dataset and the real seismic datasets, and by a regular centralized gradient descent method for the synthetic robust least squares and logistic regression datasets.

In decentralized least squares, we set the number of nodes to $m=5$ and dimension of unknown x to $n=5$. The radius specified for X is set to $R=10$. For the given nodes, we generate a network by randomly turning on each of $\binom{m}{2}$ possible edges with probability 0.5 independently. For each node i, we generate a matrix A_i with $p_i=15$ using MATLAB built-in function `randn`. We also generate a random vector $\hat{x}\in\mathbb{R}^n$ using `randn` with mean 0 and standard deviation 2. Then we simulate $b_i=A_i\hat{x}+\epsilon_i$ where ϵ_i is generated by `randn` with mean 0 and standard deviation 0.001. We set the objective function to $f_i(x)=(1/2)\|A_ix-b_i\|^2$ at node i. Therefore the Lipschitz constant of ∇f_i is $L_i=\|A_i^TA_i\|_2$, and we further set $L=\max_{1\le i\le m}\{L_i\}$. The initial guess $x_i(0)$ is set to 0 for all i. For each iteration t, the delay $\tau_i(t)$ at each node i is uniformly drawn from integers 1 to B with $B=5$, 10 and 20. For given t, the stochastic gradient is simulated by setting $\nabla F_i(x_i(t);\xi_i(t))=A_i^T(A_ix_i(t)-b_i)+\xi_i(t)$ where $\xi_i(t)$ is generated by `randn` with mean 0 and standard deviation σ set to 0.1 and 0.5. We run our algorithm using step size $\alpha(t)=1/(2L+2\eta\sqrt{t})$ with $\eta=\sqrt{[2\sigma^2+\sqrt{\pi}G^2/\lambda^2\log(\lambda^{-1})]/nR^2}$ which minimizes the $O(1/\sqrt{T})$ terms in the right side of (52). The objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m}\|y_i(T)-z(T)\|^2$ versus the iteration number T are plotted in the top row of Figure 1. In the two plots, we observe that $f(z(T))$ decays to the optimal value $f^*:=f(x^*)$ and disagreement $\sum_{i=1}^{m}\|y_i(T)-z(T)\|^2$ decays to 0 as justified by our theoretical analysis in Section III. In general, we observe that delays with larger bound B and/or larger standard deviation σ in stochastic gradient yield slower convergence, as expected.

In the second test, we apply (5) to the decentralized robust least squares problem where the objective function is set to $f_i(x):=\sum_{j=1}^{p_i}h_i^j(x)$ with

$$h_i^j(x)=\begin{cases}\frac{1}{2}|(a_i^j)^Tx-b_i^j|^2 & \text{if } |(a_i^j)^Tx-b_i^j|\le\delta\\ \delta\left(|(a_i^j)^Tx-b_i^j|-\frac{1}{2}\delta\right) & \text{if } |(a_i^j)^Tx-b_i^j|>\delta\end{cases} \qquad (57)$$

where $(a_i^j)^T\in\mathbb{R}^n$ is the j-th row of matrix $A_i\in\mathbb{R}^{p_i\times n}$, and $b_i^j\in\mathbb{R}$ is the j-th component of $b_i\in\mathbb{R}^{p_i}$ at each node i. In this test, we simulate network $G(V,E)$ and set A_i, b_i, m, n, R, $x_i(0)$ the same way as in the decentralized least squares test above, and set $\delta=2$ for the robust least squares. The stochastic gradient is given by $\nabla F_i(x;\xi_i(t))=\sum_{j=1}^{p_i}\nabla h_i^j(x)+\xi_i(t)$ where $\xi_i(t)$ is generated as before with σ set to 0.1 and 0.5. Lipschitz constants L_i and L are determined as in the previous test. The settings of η and $\tau_i(t)$ remain the same as well. The objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m}\|y_i(T)-z(T)\|^2$ are plotted in the middle row of Figure 1. In these two plots, we observe similar convergence behavior as in the test on the decentralized least squares problem above.

The last test using a synthetic dataset is on decentralized logistic regression. In this test, we generate the network $G(V,E)$ as before but work on a slightly larger problem size where each node i possesses $A_i\in\mathbb{R}^{p_i\times n}$ with $p_i=45$ and $n=15$. We generate A_i the same way as before but then replace their first columns by 1. We also generate $b_i\in\{0,1\}^{p_i}$ where each component has a random binary value. Now the objective function f_i at node i is set to

$$f_i(x)=\sum_{j=1}^{p_i}\left(\log[1+\exp((a_i^j)^Tx)]-b_i^j(a_i^j)^Tx\right), \qquad (58)$$

where $(a_i^j)^T\in\mathbb{R}^n$ is the j-th row of matrix $A_i\in\mathbb{R}^{p_i\times n}$, and $b_i^j\in\mathbb{R}$ is the j-th component of $b_i\in\mathbb{R}^{p_i}$. Then we perform (5) to solve this problem in the network G above. Note that $L_i\le\|A_i^TA_i\|_2$ for all i, and we set $L=\max_{1\le i\le m}\{\|A_i^TA_i\|_2\}$.

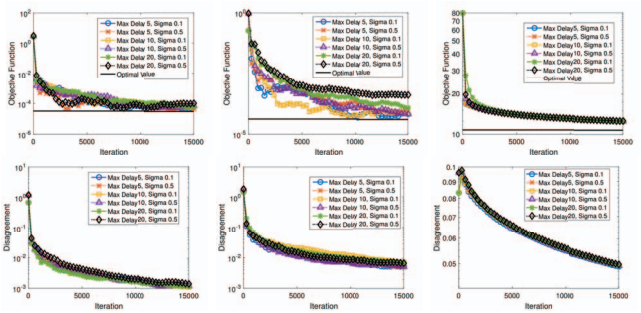

Fig. 1. Test on synthetic decentralized least-squares (left), robust least-squares (middle), and logistic regression (right) for different levels of delay $B = 5$, 10, and 20 and standard deviation in stochastic gradient $\sigma = 0.1$ and 0.5. Top: objective function $f(z(T))$ versus iteration T. Optimal value is $f^* := f(x^*)$. Bottom: disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ versus iteration number T.

The settings for the stochastic gradients, the delay $\tau_i(t)$, η, and initial value $x_i(0)$ remain the same. The objective function $f(z(T))$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ are plotted in the bottom row of Figure 1, where similar convergence behavior as in the previous two tests can be observed.

In these simulations, we notice that the largest impact to convergence arises from the delay bound used, with a smaller impact resulting from the choice of σ. Having a large maximum delay significantly impacts performance much more than using a large value for σ. In general, we expect that the best case for convergence arises from the smallest σ and delay bound, and the worst case arises when using the largest σ and largest delay bound. Supposing that one of these cannot be changed, it is best to minimize the other parameter as much as possible, noting that even for large delay, better convergence arises from smaller σ, and for large σ, reducing the delay bound ensures better performance.

V. Concluding remarks

We analyzed the convergence of method (5) for solving problem (1). As long as the random delays are bounded in expectation, using a proper diminishing step size policy, the iterates generated converge to a consensual, optimal solution.

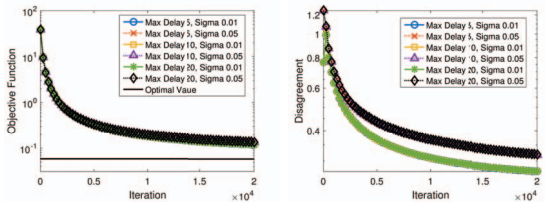

Fig. 2. Tests on decentralized regularized least-squares for different levels of delay $B = 5$, 10, and 20 and standard deviation in stochastic gradient $\sigma = 0.01$ and 0.05.

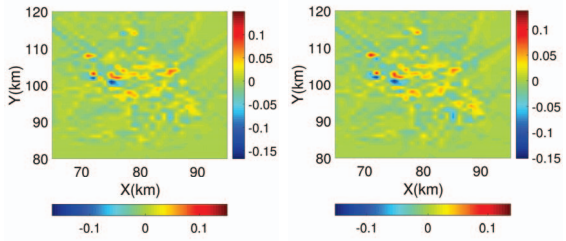

Fig. 3. Left: DDGD output image. Right: Centralized solution output image.

References

[1] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 32–43, 2014.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends{®} in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 20–27.

[4] C.-H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 66–77, 2013.

[5] G. Kamath, L. Shi, W. Z. Song, and J. Lees, "Distributed travel-time seismic tomography in large-scale sensor networks," *Journal of Parallel and Distributed Computing*, vol. 89, pp. 50–64, 2016.

[6] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Advances in Neural Information Processing Systems*, 2014, pp. 19–27.

[7] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous Distributed ADMM for Large-Scale Optimization-Part I: Algorithm and Convergence Analysis," *arXiv preprint arXiv:1509.02597*, 2015.

[8] T.-H. Chang, W.-C. Liao, M. Hong, and X. Wang, "Asynchronous Distributed ADMM for Large-Scale Optimization-Part II: Linear Convergence Analysis and Numerical Performance," *arXiv preprint arXiv:1509.02604*, 2015.

[9] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1131–1146, 2014.

[10] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, 2009.

[11] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

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

[13] A. Mokhtari and A. Ribeiro, "Decentralized double stochastic averaging gradient," *arXiv preprint arXiv:1506.04216*, 2015.

[14] L. Zhao, W.-Z. Song, and X. Ye, "Fast decentralized gradient descent method and applications to in-situ seismic tomography," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 908–917.

[15] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *Automatic control, IEEE Transactions on*, vol. 57, no. 3, pp. 592–606, 2012.

[16] D. Yuan, D. W. C. Ho, and S. Xu, "Regularized primal-dual subgradient method for distributed constrained optimization," *IEEE Transactions on Cybernetics*, 2015.

[17] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2748–2761, 2009.

[18] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[19] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "A delayed proximal gradient method with linear convergence rate," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*. IEEE, 2014, pp. 1–6.

[20] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, 2015.

[21] H. Wang, X. Liao, T. Huang, and C. Li, "Cooperative distributed optimization in multiagent networks with delays," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, no. 2, pp. 363–369, 2015.

[22] A. H. Sayed, S.-Y. Tu, and J. Chen, "Online learning and adaptation over networks: More information is not necessarily better," in *Information Theory and Applications Workshop (ITA), 2013*. IEEE, 2013, pp. 1–8.

[23] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.