Abstract—This paper studies decentralized convex optimization problems defined over networks, where the objective is to minimize a sum of local smooth convex functions while respecting a common constraint. Two new algorithms based on dual averaging and decentralized consensus-seeking are proposed. The first one accelerates the standard convergence rate $O(\frac{1}{t^2})$ in existing decentralized dual averaging (DDA) algorithms to $O(\frac{1}{t^1})$, where $t$ is the time counter. This is made possible by a second-order consensus scheme that assists each agent to locally track the global dual variable more accurately and a new analysis of the descent property for the mean variable. We remark that, in contrast to its primal counterparts, this method decouples the synchronization step from nonlinear projection, leading to a rather concise analysis and a natural extension to stochastic networks. In the second one, two local sequences of primal variables are constructed in a decentralized manner to achieve acceleration, where only one of them is exchanged between agents. In addition to this, another consensus round is performed for local dual variables. The convergence rate is proved to be $O(1)(\frac{1}{t^2} + \frac{1}{t^1})$, where the magnitude of error bound is showed to be inversely proportional to the algebraic connectivity of the graph. However, the condition for stepsize does not rely on the weight matrix associated with the graph, making it easier to satisfy in practice than other accelerated methods. Finally, comparisons between the proposed methods and several recent algorithms are performed using a large-scale LASSO problem.

Index Terms—Decentralized optimization, constrained optimization, acceleration, stochastic network, dual averaging.

I. INTRODUCTION

Decentralized optimization has received increasing attention recently due to its fundamental role in engineering design, especially of modern cyber-physical systems such as federated learning and multi-agent control systems. The problem therein usually can be cast as a group of spatially separated nodes cooperatively optimizing the sum of their local objective functions with only peer-to-peer communication. For a detailed overview of decentralized optimization and its various applications, please refer to [1], [2].

Depending on whether or not constraints can be handled, the algorithms in the literature can be classified into two categories. For unconstrained problems, the authors in [3], [4] developed the decentralized gradient descent (DGD) with constant stepsizes, where local decision variables are guided by local gradients and a consensus protocol based on doubly stochastic matrices. However, since local gradients evaluated at the global consensual optimum are not necessarily zero, two forces driven by consensus and local gradient flows will conflict with each other, therefore preventing exact optimization, that is, there exists a gap between the accumulation point and the global optimum. Several efforts have been made to overcome this drawback. For example, the authors in [5] proposed the EXTRA algorithm that adds a cumulative correction term in the iteration of DGD. Alternatively, an additional gradient-tracking process based on the dynamic average consensus scheme [6] can be used. It is shown in [7], [8] that for unconstrained smooth optimization the algorithm steered by the approximated gradient achieves an exact solution with an $O(\frac{1}{t})$ rate, where $t$ is the time counter. Based on this idea, the accelerated decentralized Nesterov gradient descent was further reported in [9] to accelerate the convergence rate to $O(\frac{1}{t^{1-c}})$ for some $c \in (0, 1.4)$ at the expense of another communication round. By modeling the decentralized optimization problem as a linearly constrained one, centralized primal-dual paradigms such as the augmented Lagrangian method (ALM) [10], the alternating direction method of multipliers (ADMM) [11] and the dual ascent [12] can also be used to design decentralized algorithms. In this framework, an accelerated primal-dual method was presented in [13], where the rate is improved to $O(1)(\frac{1}{t^2} + \frac{1}{t^1})$.

The design of a decentralized algorithm for constrained optimization is more challenging. Early method in [14] is based on the projected subgradient method and peer-to-peer diffusion, where the stepsize was made decaying for convergence. To use a constant stepsize and improve the performance, a variant of EXTRA (PG-EXTRA) was developed in [15], where the constraint is generalized as a nonsmooth indicator function and handled via the proximal operator. An $O(\frac{1}{t})$ convergence rate is stated in terms of the successive difference of variables. Recently the authors in [16] proposed an accelerated decentralized penalty method (APM) with an $O(1)(\frac{1}{t^2} + \frac{1}{t^1})$ convergence rate, where the constraint can be also treated as the nonsmooth part of the objective. It is worth to mention that in [14]–[16] the local estimates about the optimum are directly generated in the constraint set that is contained in the primal vector space of variables. There are also some schemes available in the literature where the minimizer seeking process imitates dual methods such as decentralized mirror descent [17]–[19] and decentralized dual averaging (DDA) [20]–[23]. The concept of dual methods was coined in [24], where a recursively updated dual model of the objective in conunction with a prox-function establishes the mapping from the dual space to the primal in order to substantially shrink the error bound in non-Euclidean geometries. For example, the authors in [20] developed a DDA algorithm where the global dual variable is gradually learned by a consensus scheme, and demonstrated that minimizing the approximate linear model of the global objective helps bypass the difficulty caused by projection in decentralized primal methods. Recent work in [22] introduced another averaging step to standard DDA to...
reap a non-ergodic convergence property, which helps deal with decentralized optimization with coupled constraints. For problems defined over time-varying and unbalanced networks, a DDA method with the push-sum technique was reported in [23]. Note that other centralized methods for constrained optimization such as the Frank-Wolfe method [25] and primal-dual methods have also been used to develop decentralized algorithms [26]–[28], and the best convergence rate achieved so far for these methods is $O(\frac{1}{t^2})$.

Although centralized dual methods in the literature have demonstrated advantages over their primal counterparts in terms of handling constraints and time-varying communication networks, and analysis complexity, all the results reported so far focused only on nonsmooth problems and have a convergence rate of $O(\frac{1}{t^2})$. Considering this, a question naturally arises: If the objective functions exhibit some desired properties, e.g., smoothness, is it possible to accelerate the convergence rate of DDA to $O(\frac{1}{t^2})$ or even faster? We provide a affirmative answer to this question in this work. The main results and contributions are summarized in the following:

- To achieve an $O(\frac{1}{t^2})$ convergence rate, we propose a second-order consensus scheme that assists each agent to locally track the global dual variable more accurately than that in [20]. With the new dual estimate, the accumulation of error over time between local primal variables and their mean is proved to admit an upper bound in terms of the successive difference of mean variables. This together with a rigorous investigation of the descent property of the mean variable yields an $O(\frac{1}{t^2})$ convergence rate. We then show that the proposed method naturally lends itself to the case with stochastic communication primarily because the synchronization is sought purely in the dual vector space and is not coupled with the projection operation, a feature that existing decentralized constrained optimization methods do not have.

- To further accelerate the convergence rate, we consider increasing weights for new gradients entering the linear model of the objective [29]. For decentralized implementation, a first-order consensus protocol is used to track the global dual variable. Then two local sequences of primal variables are recursively generated using another consensus round and a prox-mapping for the tracked dual variable. The convergence rate is proved to be $O(1)(\frac{1}{t^2} + \frac{1}{t})$, where the convergence constant is related to the second largest singular value of the mixing matrix. In contrast to existing methods [16], the stabilizing step-size does not rely on the mixing matrix, therefore making it much easier to choose in practice and speeding up the convergence. Numerical comparison results based on a large-scale LASSO problem illustrate the advantage of the proposed method.

**Notation:** Let $\mathbb{R}$ and $\mathbb{R}^n$ represent the set of reals and the $n$-dimensional Euclidean space, respectively. Notation $\|\cdot\|_p$ denotes the $l_p$-norm operator in this space for some $p \geq 1$. We denote by $1$ a column vector of all ones, where the dimension shall be understood from the context. Given a matrix $P \in \mathbb{R}^{n \times n}$, its spectral radius is denoted by $\rho(P)$, and its eigenvalues and singular values are denoted by $\lambda_1(P) \geq \lambda_2(P) \geq \cdots \geq \lambda_n(P)$ and $\sigma_1(P) \geq \sigma_2(P) \geq \cdots \geq \sigma_n(P)$, respectively.

**II. Problem Statement and Preliminaries**

**A. Problem Statement**

We consider the multi-agent optimization problem given by

$$\min_{x \in X} \sum_{i=1}^{n} f_i(x)$$

where $f_i : \mathbb{R}^m \rightarrow \mathbb{R}, i \in \mathbb{N}_{[1,n]}$ represents the local objective function privately known by agent $i$, $x \in \mathbb{R}^m$ stands for the common decision variable, and $X \subseteq \mathbb{R}^m$ denotes the constraint set that is assumed to be convex and compact. Throughout this paper, we denote one of the minimizers by $x^*$. For [1], the following standard assumption is made.

**Assumption 1.** Each $f_i(x), i \in \mathbb{N}_{[1,n]}$ is convex and has Lipschitz continuous gradients with parameter $L$, i.e.,

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in X,$$

where $\nabla f_i$ denotes the gradient of $f_i$.

A direct consequence of the above assumption is

$$f_i(y) \leq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{L}{2}\|x - y\|_2^2, \forall x, y \in X.$$  

We use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to describe the communication pattern between agents, where $\mathcal{V} = \{1, \cdots, n\}$ denotes the set of $n$ agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of undirected channels that connect agents, that is, the pair $(i, j) \in \mathcal{E}$ for $i, j \in \mathcal{V}$ indicates that there exists a link between node $i$ and $j$. For each communication link $(i, j) \in \mathcal{E}$, a positive weight $p_{ij}$ is assigned. Agent $j$ is said to be a neighbor of $i$ if there exists a link between them, and the set of $i$’s neighbors is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$. We make the following assumption for the graph and the weight matrix $P = [p_{ij}]$ to ensure that $\sigma_2(P) < 1$ [22].

**Assumption 2.** The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected. $P = [p_{ij}]$ is a doubly stochastic weight matrix, i.e., $P1 = 1$ and $1^T P = 1^T$, and has a strictly positive diagonal, i.e., $p_{ii} > 0$.

Given a connected network, typical rules to construct a weight matrix $P$ satisfying Assumption 2 include the constant edge weights and the Metropolis-Hastings algorithm [30].

**B. Preliminaries**

Let $\mathbb{E}$ be a finite dimensional linear vector space, and $\mathbb{E}^*$ its dual space. For some $x \in \mathbb{E}$ and $y \in \mathbb{E}^*$, denote the value of the linear function $y$ evaluated at $x$ by $\langle y, x \rangle$. If $\mathbb{E} = \mathbb{R}^m$, then $\mathbb{E}^* = \mathbb{R}^m$ and $(y, x) = y^T x$ for $x, y \in \mathbb{R}^m$. For a given (fixed) norm $\|\cdot\|$ associated with this space, the dual norm is defined as

$$\|y\|_* = \max_{\|x\| \leq 1} \langle x, y \rangle.$$  

**In this work,** we consider that the test point $x_t$ and the gradient $\nabla f(x_t)$ belong to the primal and dual vector space,
respectively. In what follows, we stay with the $l_2$-norm whose dual norm is itself for brevity.

**Definition 1.** A function $d(\cdot) : \mathcal{X} \to \mathbb{R}$ is 1-strongly convex on $\mathcal{X}$ with respect to $\|\cdot\|_2$, if
\[
d(y) \geq d(x) + \langle \nabla d(x), y - x \rangle + \frac{1}{2} \|y - x\|^2_2, \quad \forall x, y \in \mathcal{X}.
\]

**Definition 2.** The convex conjugate of function $d(\cdot) : \mathcal{X} \to \mathbb{R}$ is denoted as
\[
d^*(y) = \max_{x \in \mathcal{X}} \{\langle y, x \rangle - d(x)\}, \forall y \in \mathbb{R}^m.
\]

According to Danskin’s Theorem [31], we have the following lemma.

**Lemma 1.** If function $d(\cdot)$ is 1-strongly convex and differentiable, then
\[
\nabla d^*(y) = \arg \max_{x \in \mathcal{X}} \{\langle y, x \rangle - d(x)\}.
\]

In addition, for any $x, y \in \mathbb{R}^m$, we have
\[
\|\nabla d^*(x) - \nabla d^*(y)\|_2 \leq \|x - y\|_2. \tag{2}
\]

**Definition 3.** For $x, y \in \mathcal{X}$, the Bregman divergence induced by a differentiable function $d(\cdot)$ is defined as
\[
D_d(x, y) = d(x) - d(y) - \langle \nabla d(y), x - y \rangle.
\]

Without loss of generality, we focus on the case with $m = 1$ for ease of notation in the remaining sections, i.e., $1 \otimes I_m = 1$, $P \otimes I_m = P$.

### III. Development of Algorithms

#### A. Fast Decentralized Dual Averaging

The decentralized methods in this work are based on the dual averaging (DA) method in [32]. The DA generates sequences of estimates about the minimizer $\{x_k\}_{k \geq 0}$ and the dual variable $\{\sum_{k=0}^t \nabla f(x_k)\}_{t \geq 0}$ according to the following rule:
\[
x_{t+1} = \nabla d^*\left(-a_t \sum_{k=0}^t \nabla f(x_k)\right) \tag{3}
\]
where
\[
\nabla d^*(-y) = \arg \min_{x \in \mathcal{X}} \{d(x) + \langle y, x \rangle\},
\]

$d : \mathcal{X} \to \mathbb{R}$ is 1-strongly convex and differentiable, and $\{a_t\}_{t \geq 0}$ is a sequence of positive control parameters that directly impacts the convergence of DA. It is shown in [32] that an $O(\frac{1}{t})$ convergence rate is ensured when $a_t$ decreases at $O(\frac{1}{\sqrt{t}})$ for nonsmooth objective functions. When the objective function is smooth, an appropriate positive constant $\alpha_0 = a$ can be used to achieve an $O(\frac{1}{t})$ rate [33]. In the literature, several decentralized variants of DA have been developed [20], [22], [23]. Generally, they involve iteratively estimating the global dual variable $\sum_{k=0}^t \nabla f(x_k)$ in the following way:
\[
q_{i,t} = \sum_{j=1}^n p_{ij} q_{j,t-1} + \nabla f_i(x_{i,t}) \tag{4}
\]
where $q_{i,t}$ is an estimate of $\sum_{k=0}^t \nabla f(x_k)$ locally maintained by agent $i$, and $x_{i,t}$ represents the local estimate of the global minimizer. An equivalent form of (4) is
\[
q_{i,t} = \sum_{k=0}^t s_{i,k} \tag{5}
\]
\[
s_{i,t+1} = \sum_{j=1}^n p_{ij} s_{j,t} + \nabla f_i(x_{i,t+1}) - \nabla f_i(x_{i,t})
\]

where $s_{i,0} = \nabla f_i(x_{i,0})$. This suggests that $q_{i,t}$ consists of the estimated global gradients over time supplied by the dynamic average consensus scheme. However, it is shown in [22] that, with such an estimation scheme, the deviation between $q_{i,t}$ and the dual variable does not admit a finite upper bound. Due to this, the weight associated with $q_{i,t}$ has to be decaying in nonsmooth case for convergence. Specifically, when the weight decreases at rate $O(\frac{1}{t^\alpha})$, a slow convergence rate of $O(\frac{1}{t^{\alpha-1}})$ in terms of the error between objective function values is ensured [20], [22], [23].

To possibly accelerate convergence for smooth objectives using a constant weight sequence, the global dual variable should be more accurately estimated. Motivated by this, we use the following second-order consensus protocol to track the global dual variable:
\[
ds_{i,t+1} = \sum_{j=1}^n p_{ij} s_{j,t} + \nabla f_i(x_{i,t+1}) - \nabla f_i(x_{i,t}) \tag{5a}
\]
\[
h_{i,t+1} = \sum_{j=1}^n p_{ij} h_{j,t} + s_{i,t+1} - s_{i,t} \tag{5b}
\]

We can see from an example in Fig. 1 that for a constant ‘$\alpha$’, the second-order scheme in (5) gives a significantly smaller tracking error than (4). Note that we make the change of the reference signal in Fig. 1 square summable over time, which behaves similarly with the gradient in smooth optimization. Based on it, the estimate of the global minimizer can be determined by projecting the dual estimate into the primal space as follows
\[
x_{i,t+1} = \nabla d^*\left(-a \sum_{k=0}^t h_{i,k}\right). \tag{6}
\]

This algorithm is referred to as fast decentralized dual averaging (FDDA) and summarized in Algorithm 1. Its procedure is illustrated in Fig. 2 where the dashed lines represent the communication between agents.

#### B. Accelerated Decentralized Dual Averaging

To further accelerate the convergence rate, increasing weights for new gradients entering the linear model of the objective should be used. Following [29], we consider an increasing sequence $a_t = a(t+1), t \geq 0$ for some $a > 0$ such that the dual variable becomes $\sum_{k=0}^t a_k \nabla f(x_k)$. However, when increasing weights are considered, the second-order consensus scheme [3] still cannot track the dual variable with sufficient accuracy, and after the prox-mapping in (6), large consensus errors still exist between primal variables and
Fig. 1. An example of five agents tracking the mean of time-varying signals $\nabla f_i(x_{i,t}) = i - \frac{2}{t}, i = 1 \cdots 5$. The vertical axis represents the tracking error $\|\sum_{k=0}^{t} a_k s_{i,k} - \sum_{k=0}^{t} a_k \nabla f_i(x_{i,k})\|_2$ for (4) and $\|\sum_{k=0}^{t} a_k h_{i,k} - \sum_{k=0}^{t} a_k \nabla f_i(x_{i,k})\|_2$ for (5) with different choices of $a_k$. For different protocols, the same zero initial condition is used.

Algorithm 1 Fast decentralized dual averaging (FDDA)

1: Set $t = 0$, $x_{i,0} = \arg \min_{x \in X} d(x)$, $d(x_{i,0}) = 0$, $h_{i,0} = s_{i,0} = \nabla f_i(x_{i,0}), \forall i \in \mathcal{V}$. Choose a positive constant $a_t = a, t \geq 1$.
2: while Convergence is not reached do
3: for Each agent $i \in \mathcal{V}$ (in parallel) do
4: Receive $s_{j,t}, h_{j,t}, \forall j \in \mathcal{N}_i$;
5: Perform local computation in (5) and (6);
6: Broadcast $s_{i,t+1}, h_{i,t+1}, \forall j \in \mathcal{N}_i$;
7: end for
8: Set $t = t + 1$.
9: end while

their mean, which probably cannot be compensated by the descent property of the mean variable in the analysis. This argument is illustrated in Fig. [1]. Alternatively, we consider a synchronization step for primal variables

$$x_{i,t+1}^{[1]} = \frac{A_t}{A_{t+1}} \sum_{j=1}^{n} p_{ij} x_{j,t}^{[2]} + \frac{a_{t+1}}{A_{t+1}} \hat{x}_{i,t},$$  \hspace{1cm} (7a)

$$x_{i,t+1}^{[2]} = \frac{A_t}{A_{t+1}} \sum_{j=1}^{n} p_{ij} x_{j,t}^{[2]} + \frac{a_{t+1}}{A_{t+1}} \hat{x}_{i,t+1},$$  \hspace{1cm} (7b)

where $A_t = \sum_{k=1}^{t} a_k$ and

$$\hat{x}_{i,t+1} = \nabla d^* \left( - \sum_{k=0}^{t+1} a_k s_{i,k} \right).$$  \hspace{1cm} (8)

Note that $A_0 = 0$ by convention. For the dual variable, another consensus round is performed

$$s_{i,t+1} = \sum_{j=1}^{n} p_{ij} s_{j,t} + \nabla f_j(x_{j,t+1}^{[1]}) - \nabla f_j(x_{j,t}^{[1]}).$$  \hspace{1cm} (9)

By collectively considering them, the accelerated decentralized dual averaging (ADDA) algorithm is summarized in Algorithm 2. One of its round is demonstrated in Fig. [3]. It is worth to mention that although there are two sequences of primal variables $x_{i,t+1}^{[1]}$ and $x_{i,t+1}^{[2]}$, only the second one is exchanged between agents. Therefore, ADDA essentially uses the same communication cost as FDDA for acceleration.

Algorithm 2 Accelerated decentralized dual averaging (ADDA)

1: Set $t = 0$, $x_{i,0}^{[1]} = x_{i,0}^{[2]} = \hat{x}_{i,0} = \arg \min_{x \in X} d(x)$, $d(\hat{x}_{i,0}) = 0$, $s_{i,0} = \nabla f_i(x_{i,0}), \forall i \in \mathcal{V}$. Choose a positive control sequence $a_t = a(t+1), t \geq 1$.
2: while Convergence is not reached do
3: for Each agent $i \in \mathcal{V}$ (in parallel) do
4: Receive $s_{j,t}, x_{j,t}^{[2]}, \forall j \in \mathcal{N}_i$;
5: Perform local computation in (7), (8) and (9);
6: Broadcast $s_{i,t+1}, x_{i,t+1}^{[2]}, \forall j \in \mathcal{N}_i$;
7: end for
8: Set $t = t + 1$.
9: end while

C. Comparison with Existing Algorithms

First, we compare FDDA with existing decentralized algorithms that have an $O(\frac{1}{t})$ rate. Recall the PG-EXTRA method in [15]

$$x_{i,t+1} = \sum_{j=1}^{n} p_{ij} x_{j,k} + x_{i,t} - \sum_{j=1}^{n} p_{ij} x_{j,k-1}$$

$$- a \left( \nabla f_i(x_{i,t+1}) - \nabla f_i(x_{i,t}) \right)$$

$$x_{i,t+1} = \arg \min_{x \in X} \|x - x_{i,t+1}\|_2$$

where ‘$a$’ represents the constant stepsize and $[\hat{p}_{i,j}] = \hat{P} = \frac{P+I}{2}$. We can see that PG-EXTRA mixes vectors from different vector spaces, considering that $\{x_{i,t}^{[1]}\}_{t\geq0}, \{x_{i,t}^{[2]}\}_{t\geq0}$ and
Fig. 3. Illustration of ADDA.

\{\nabla f_i(x_{i,t+1}) - \nabla f_i(x_{i,t})\}_{t \geq 0}
are belong to the primal space and the dual space, respectively. This is in sharp contrast with the FDDA in [5] and [6] where two types of vectors are separately treated. We will show this essentially helps decouple the consensus-seeking procedure from the projection, and keep the iteration in [5] almost linear (considering that the Lipschitz continuity of gradients can be used to simplify \( \nabla f_i(x_{i,t+1}) - \nabla f_i(x_{i,t}) \)), and therefore bypass difficulties with the consensus-projection coupling that have been challenging for primal methods. As a consequence, the analysis of FDDA can be kept rather concise, in sharp contrast with the FDDA space, respectively. This is in sharp contrast with the FDDA algorithm [12]. However, to the best of the authors' knowledge, they cannot handle stochastic networks.

Accelerated methods for decentralized constrained optimization are rare in the literature. Recently, the authors in [10] developed the APM algorithm based on accelerated penalty methods. The iteration therein obeys

\begin{align}
\tilde{x}_{i,k+1} &= \arg \min_{x \in \mathcal{X}} \left\| x - x_{i,k} + \frac{\theta_k (1 - \theta_{k-1})}{\theta_k} (\hat{x}_{i,k} - \tilde{x}_{i,k-1}) \right\|_2^2 \quad (10a) \\
q_{i,k} &= \nabla f_i(x_{i,k}) + \beta_0 k \sum_{i=1}^n p_{i,j} (x_{i,k} - x_{j,k}) \quad (10b) \\
\hat{x}_{i,k+1} &= \arg \min_{x \in \mathcal{X}} \left\| x - x_{i,k} + \frac{q_{i,k}}{L + \beta_0 / \theta_k} \right\|_2^2 \quad (10c)
\end{align}

where \( \beta_0 = \frac{L}{\sqrt{1 - \lambda_2(F)}} \) and \( \theta_k \) is a decreasing parameter satisfying \( \theta_0 = \frac{\theta_{k-1}}{1 + \theta_{k-1}} \) with \( \theta_0 = 1 \). By letting \( q_{i,k} = \theta_k q_{i,k} \), we can equivalently rewrite (10b) and (10c) as

\begin{align}
q_{i,k} &= \theta_k \nabla f_i(x_{i,k}) + \beta_0 \sum_{i=1}^n p_{i,j} (x_{i,k} - x_{j,k}) \\
x_{i,k+1} &= \arg \min_{x \in \mathcal{X}} \left\| x - x_{i,k} + \frac{q_{i,k}}{L \theta_k + \beta_0} \right\|_2^2,
\end{align}

from which we can see that new gradients are assigned with decreasing weights, while in [8] for ADDA increasing weights are used. We will show in simulation that decreasing weights lead to slower convergence. There are also a few other accelerated decentralized methods such as [9], [13], however they do not apply to constrained problems.

IV. CONVERGENCE ANALYSIS

A. Preliminaries and Supporting Lemmas

In this subsection, we present several definitions and two useful lemmas, which play an instrumental role in analyzing convergence rates of the proposed algorithms.

For Algorithm [1] we let

\begin{align}
x_t &= \begin{bmatrix} x_{1,t}^1 & x_{2,t}^1 & \cdots & x_{n,t}^1 \end{bmatrix}^T, \quad h_t = \begin{bmatrix} h_{1,t}^1 & h_{2,t}^1 & \cdots & h_{n,t}^1 \end{bmatrix}^T, \quad s_t = \begin{bmatrix} s_{1,t}^1 & s_{2,t}^1 & \cdots & s_{n,t}^1 \end{bmatrix}^T, \quad \nabla_t = \begin{bmatrix} \nabla f_1(x_{1,t}) & \nabla f_2(x_{2,t}) & \cdots & \nabla f_n(x_{n,t}) \end{bmatrix}^T,
\end{align}

g_t = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x_{i,t}), \quad \pi_t = \frac{1}{n} x_t^1, \quad \overline{x}_t = \frac{1}{n} s_t^1.

For Algorithm [2] with a slight abuse of notation, we preserve the definitions for \( s_t \) and \( \pi_t \), and let

\begin{align}
x_t^{[1]} &= \begin{bmatrix} x_{1,t}^{[1]} & x_{2,t}^{[1]} & \cdots & x_{n,t}^{[1]} \end{bmatrix}^T, \quad x_t^{[2]} = \begin{bmatrix} x_{1,t}^{[2]} & x_{2,t}^{[2]} & \cdots & x_{n,t}^{[2]} \end{bmatrix}^T, \quad \nabla_t^{[1]} = \begin{bmatrix} \nabla f_1(x_{1,t}^{[1]}) & \nabla f_2(x_{2,t}^{[1]}) & \cdots & \nabla f_n(x_{n,t}^{[1]}) \end{bmatrix}^T,
\end{align}

and \( g_t^{[1]} = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x_{i,t}^{[1]}) \). Define \( \pi_t^{[1]} = \frac{1}{n} 1^T x_t^{[1]} \) and \( \pi_t^{[2]} = \frac{1}{n} 1^T x_t^{[2]} \). According to (11), we have

\begin{align}
\pi_{t+1}^{[1]} &= \frac{A_t}{A_{t+1}} \pi_t^{[2]} + \frac{a_{t+1} - 1}{A_{t+1}} \pi_t, \\
\pi_{t+1}^{[2]} &= \frac{A_t}{A_{t+1}} \pi_t^{[1]} + \frac{a_{t+1} - 1}{A_{t+1}} \pi_t^{[2]},
\end{align}

where \( \pi_t = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x_{i,t}) \).

The following conservation property holds true for both FDDA and ADDA.

Lemma 2. For Algorithms [1] and [2] we have \( \overline{x}_t = \pi_t = g_t^{[1]} \), respectively.

Proof. By projecting (11) into the average space, the proof is completed. \( \square \)

Then, a lemma is stated for the prox-mapping. It will be used to establish descent properties for operations in (6) and (8).

Lemma 3. Given an arbitrary sequence of variables \( \{v_t\}_{t \geq 0} \) in the dual space and a positive sequence \( \{a_t\}_{t \geq 0} \) generated by

\begin{align}
w_t = \nabla^* \left( - \sum_{k=0}^t a_k v_k \right),
\end{align}

it holds

\begin{align}
\sum_{k=1}^t a_k \langle v_k, w_k - x^* \rangle \leq d(x^*) - \sum_{k=1}^t \frac{1}{2} \|w_k - w_{k-1}\|^2. \quad (12)
\end{align}

Proof. The proof is postponed to Appendix A. \( \square \)
B. Convergence Analysis for FDDA

For the iteration rule in Algorithm [1] it may be difficult to directly analyze the convergence of \( x_{i,t} \). To facilitate the analysis, we construct an auxiliary sequence \( \{y_t\}_{t \geq 0} \) whose update exploits the global information \( y_t \) and obeys the following rule

\[
y_{t+1} = \nabla d^*(\sum_{k=0}^t y_k),
\]

where the initial vector \( y_0 = \arg\min_{x \in X} d(x) \) satisfying \( d(y_0) = 0 \). The sequence \( \{y_t\}_{t \geq 0} \) will function as a reference that converges to the global minimizer for agent \( i \) to track.

The following lemma establishes the relation between sequences \( \{x_{i,t}\}_{t \geq 0} \) and \( \{y_t\}_{t \geq 0} \); the deviation between them represents the consensus error to be compensated in convexity analysis.

**Lemma 4.** For Algorithm [1] if Assumptions [2] are satisfied, \( \rho(M(a)) < 1 \), where

\[
M(a) = \begin{bmatrix}
\beta & a \\
L(\beta + 1) & \beta + La
\end{bmatrix},
\]

and \( \beta = \sigma_2(P) \), it holds that

\[
\sum_{k=1}^t \|x_k - 1y_k\|^2 \leq \frac{n}{(1 - \rho(M(a)))^2} \sum_{k=1}^t \|y_k - y_{k-1}\|^2.
\]

**Proof.** The proof is deferred to Appendix B. \( \square \)

**Remark 1.** Lemma 4 requires the spectral radius of \( M(a) \) to be smaller than 1. The eigenvalues of \( M(a) \) are identified as

\[
2\beta + aL \pm \sqrt{a^2 L^2 + 4(\beta + 1)aL}.
\]

Denote them by \( \lambda_1 > \lambda_2 \). Note that when \( a \) approaches 0, \( \rho(M(a)) \) converges to \( \beta \), which is smaller than 1 by Assumption [2] Indeed, it can be verified that \( \lambda_1 \) and \( \lambda_2 \) are monotonically increasing and decreasing over \( a \), respectively. By further letting \( \lambda_1 = 1 \) and \( \lambda_2 = -1 \), we obtain that as long as

\[
a < \frac{(1 - \beta)^2}{2L}
\]

is satisfied, \( \rho(M(a)) < 1 \) holds true.

We are now in a position to present the convergence result for FDDA.

**Theorem 1.** For Algorithm [1] if Assumptions [2] and the condition in Lemma 4 are satisfied, and

\[
1 + \tau + \frac{1 + \frac{1}{\tau}}{(1 - \rho(M(a)))^2} \leq \frac{1}{aL}
\]

for some \( \tau > 0 \), then

\[
f(\bar{y}_t) - f(x^*) \leq \frac{nd(x^*)}{at},
\]

where \( \bar{y}_t = \frac{1}{t} \sum_{k=1}^t y_k \).

**Proof.** Consider

\[
\sum_{j=1}^n a\left(f_j(y_k) - f_j(x^*)\right) \leq \sum_{j=1}^n a\left(\frac{L}{2}\|y_k - x_{j,k-1}\|^2 + f_j(x_{j,k-1}) - 4\sigma \|y_k - x_{j,k-1}\| + \beta \|y_k - x_{j,k-1}\|\right),
\]

where \( \sigma = \frac{\gamma}{\alpha} \).

This together with convexity of \( f_j \)

\[
f(\bar{y}_t) - f(x^*) \leq \frac{\tau}{2} \sum_{k=1}^t \|y_k - y_{k-1}\|^2.
\]

By plugging (17) and the consensus error derived in Lemma 4 into (16), we arrive at

\[
\sum_{k=1}^t \left(\langle a g_{k-1}, y_k - x^*\rangle\right) \leq \frac{\tau}{2} \sum_{k=1}^t \|y_k - y_{k-1}\|^2.
\]

This gives us the desired inequality in (15), thereby concluding the proof. \( \square \)

Next, we demonstrate that FDDA accommodates stochastic communication. In particular, we model each communication link \( (i, j) \) in the supergraph \( G \) as a Beroulli process with a certain probability; and the Bernoulli processes associated with different links are statistically independent [3]. Under this model, the topology at each time \( t \) is a random undirected graph \( G_t = \{V, E_t\} \). For each communication link active at time \( t \), a positive weight \( p_{ij,t} \) is assigned. We make the following assumption for the weight matrix \( P_t = [p_{ij,t}] \).

\[
\sum_{j=1}^n a\left(f_j(y_k) - f_j(x^*)\right) \leq \sum_{k=1}^n \left(\langle a p_{ij,t}, f(y_k) - f(x^*)\rangle\right),
\]

with \( a = \frac{1}{t} \sum_{k=1}^t a(y_k) \).

Proof. Consider
Assumption 3. The weight matrix \( P_t = [p_{i,t}] \) at each time instant \( t \) is doubly stochastic and statistically independent with each other. In addition, it holds that \( \eta = \sqrt{\rho(\mathbf{E}(P_t^2) - \frac{11L}{2})} < 1 \).

Corollary 1. For Algorithm 1 if Assumptions 1 and 3 are satisfied, \( \rho(N(a)) < 1 \) where

\[
N(a) = \begin{bmatrix} \eta & a \\ L(\eta + 1) & \eta + La \end{bmatrix},
\]

and

\[
1 + \tau + \frac{1 + \frac{1}{2}}{(1 - \rho(N(a)))^2} \leq \frac{1}{\alpha L} \quad (19)
\]

for some \( \tau > 0 \), then

\[
\mathbf{E}[f(\tilde{y}_t) - f(x^*)] \leq \frac{nd(x^*)}{at},
\]

where \( \tilde{y}_t = \frac{1}{T} \sum_{k=1}^t y_k \).

Proof. Firstly, we bound the expectation of the consensus error \( \mathbf{E}[\|x_k - 1y_k\|_2^2] \). By invoking Lemma 4 with replacements of \( \|x\|_2 \) by its expectation and \( \sigma_2(P) \) by \( \eta \), we have

\[
\sum_{k=1}^t \mathbf{E}[\|x_k - 1y_k\|_2^2] \leq \frac{n}{(1 - \rho(N(a)))^2} \sum_{k=1}^t \mathbf{E}[\|y_k - y_{k-1}\|_2^2]
\]

when \( \rho(N(a)) < 1 \). Since \( \{y_k\}_{k=0}^\infty \) is arbitrary in Lemma 3, we also have that (14) holds valid. Then, by a similar argument as in (18), we are able to bound the expectation of objective error by

\[
\sum_{k=1}^t a \mathbf{E}[f(y_k) - f(x^*)] \leq nd(x^*) + \frac{naL}{2} \left( 1 + \tau + \frac{1 + \frac{1}{2}}{(1 - \rho(N(a)))^2} - \frac{1}{\alpha L} \right) \sum_{k=1}^t \mathbf{E}[\|y_k - y_{k-1}\|_2^2].
\]

(21)

It is easy to verify that when ‘a’ further satisfies (19), the inequality in (20) holds true.

In Theorem 1 and Corollary 1, the parameter ‘a’ should satisfy two different conditions. The first one \( \rho(M(a)) < 1 \) (\( \rho(N(a)) < 1 \) for Corollary 1) ensures that the deviation between local primal variables and their average can be bounded from above, while the second one ensures that the consensus error can be compensated by the descent property of the mean variable. In addition, Theorem 1 and Corollary 1 state that \( \tilde{y}_t \) converges to the global minimizer at an \( O(\frac{1}{t}) \) rate (in expectation for Corollary 1). By (14) and convexity of the operator \( \|\cdot\|_2^2 \), one has

\[
t\|x_t - 1\tilde{y}_t\|_2^2 \leq \sum_{k=1}^t \|x_k - 1y_k\|_2^2 \leq \frac{n}{1 - \rho(M(a))} \sum_{k=1}^t \|y_k - y_{k-1}\|_2^2,
\]

where \( \tilde{x}_t = \frac{1}{T} \sum_{k=1}^t x_k \). Moreover, from (18), we know that the right-hand side of (22) remains finite as \( t \) approaches infinity. Therefore, \( \|\tilde{x}_t - 1\tilde{y}_t\|_2^2 \) converges at an \( O(\frac{1}{t}) \) rate, where \( \tilde{x}_t = \frac{1}{T} \sum_{k=1}^t x_{i,k} \). This implies that \( \tilde{x}_t \) shares a similar convergence guarantee with \( \tilde{y}_t \). In what follows, we consider an unconstrained version of (1), i.e., \( X = \mathbb{R}^m \), where a convergence result can be directly stated for \( f(\tilde{x}_{i,k}) - f(x^*) \).

Corollary 2. Suppose \( X = \mathbb{R}^m \) in (1) and \( d(x) = \frac{1}{2}\|x\|_2^2 \) in (6). For Algorithm 1 if Assumptions 1, 2, and the condition in Lemma 4 are satisfied, and

\[
1 + \tau + \frac{5 + \frac{1}{2}}{(1 - \rho(M(a)))} \leq \frac{1}{\alpha L},
\]

for some \( \tau > 0 \), then

\[
f(\tilde{x}_{i,t}) - f(x^*) \leq \frac{n^2\|x^\ast\|_2^2}{2at}.
\]

Proof. The proof is given in Appendix C.

C. Convergence Analysis for ADDA

Assumption 4. For the problem in (1), the constraint set \( X \) is bounded with the following diameter:

\[
D = \max_{x,y \in X} \|x - y\|_2.
\]

Lemma 5. For Algorithm 2 if Assumptions 1, 2, and 4 are satisfied, then

\[
\|x_t^{[2]} - 1\tilde{x}_t^{[2]}\|_2 \leq \frac{\alpha \sqrt{\beta}}{A_t} C_p
\]

(25)

\[
\|x_t^{[1]} - 1\tilde{x}_t^{[1]}\|_2 \leq \frac{\alpha \sqrt{\beta}}{A_t} C_p
\]

(26)

for all \( t > 0 \), where \( C_p = \sqrt{\frac{1}{t}} \sqrt{nD} \), and \( \beta = \sigma_2(P) \).

Proof. Please see Appendix D for the proof.

Lemma 6. For Algorithm 2 if Assumptions 1, 2, and 4 are satisfied, then

\[
\|s_t - 1\tilde{y}_t^{[1]}\|_2 \leq \frac{\alpha \sqrt{\beta}}{A_t} C_g
\]

(27)

for all \( t \geq 0 \), where \( C_g = \sqrt{\frac{1}{\eta^2}} \sqrt{2\gamma M + 2C_p} \) and \( \beta = \sigma_2(P) \).

Proof. The proof is given in Appendix E.

Lemma 5 establishes decreasing upper bounds for consensus errors between variables \( x_t^{[1]}, x_t^{[2]} \) and their averages, and Lemma 6 proves the upper bound for the deviation between locally tracked gradients and their mean. In particular, the errors are guaranteed to converge at an \( O(\frac{1}{t}) \) rate, which will be exploited in the following rate analysis of ADDA.

Theorem 2. For Algorithm 7 if Assumptions 1, 2, and 4 are satisfied, and

\[
a < \frac{1}{2L(1 + 2\gamma)}
\]

(28)
for some \( \gamma > 0 \), then
\[
\begin{align*}
  f(\bar{x}_t^2) - f(x^*) \\
  \leq \frac{nd(x^*)}{A_t} + \left( \frac{D(1+1/\gamma) + (2 + \gamma + \frac{3}{\gamma})C_p^2}{2(1 + 2\gamma)} \right) t A_t,
\end{align*}
\]
where constants \( C_p \) and \( C_g \) are defined in Lemma 5 and Lemma 6 respectively.

**Proof.** We consider
\[
A_t \left( f(\bar{x}_t^2) - f(x^*) \right)
\]
\[
= \sum_{k=1}^{t} \left( A_k f(\bar{x}_k^2) - A_k (f(\bar{x}_k^2) - f(x^*) - \sum_{i=1}^{n} a_k f(x^*) \right)
\]
\[
= \sum_{k=1}^{t} \left( A_k f(\bar{x}_k^2) - A_k f(\bar{x}_k^2) + A_k (f(\bar{x}_k^2) - f(x^*) - \sum_{i=1}^{n} a_k f(x^*) \right)
\]
\[
= \sum_{k=1}^{t} a_k f(x_k^1) - \sum_{k=1}^{t} a_k f(x^*)
\]

Using the convexity of the objective over (I) and (II) gives
\[
A_t \left( f(\bar{x}_t^2) - f(x^*) \right)
\]
\[
\leq \sum_{k=1}^{t} \left( A_k f(\bar{x}_k^2) - A_k f(x_k^1) \right)
\]
\[
+ A_{k-1} \left( \nabla f(\bar{x}_k^1), x_k^1 - x_k^{2} \right) + a_k \left( \nabla f(\bar{x}_k^1), x_k^1 - x^* \right)
\]

Recall in (11) that
\[
A_t \bar{x}_t^2 = A_{t-1} \bar{x}_{t-1}^2 + a_t x_t.
\]

Therefore we are able to obtain
\[
A_t \left( f(\bar{x}_t^2) - f(x^*) \right)
\]
\[
\leq \sum_{k=1}^{t} A_k \left( f(\bar{x}_k^2) - f(x_k^1) + \left( \nabla f(\bar{x}_k^1), x_k^1 - x_k^{2} \right) \right)
\]
\[
+ \sum_{k=1}^{t} a_k \left( \nabla f(\bar{x}_k^1), x_k^1 - x^* \right)
\]
\[
(III) \leq \sum_{k=1}^{t} \frac{A_k L}{2} n \|x_k^1 - x_k^{2}\|^2 + \sum_{i=1}^{n} \sum_{k=1}^{t} a_k \left( s_{i,k}^1, x_k + x_{i,k}^1 - x^* \right)
\]
\[
(V) + \sum_{k=1}^{t} \sum_{i=1}^{n} a_k \left( \frac{1}{n} \nabla f(\bar{x}_k^1), s_{i,k}^1, x_k + x_{i,k}^1 - x^* \right)
\]
\[
(VI) \leq \left( \frac{D(1+1/\gamma) + (2 + \gamma + \frac{3}{\gamma})C_p^2}{2(1 + 2\gamma)} \right) \frac{t}{A_k} \sum_{k=1}^{t} a_k^2
\]

To bound (VI), we consider
\[
\sum_{i=1}^{n} a_k \left( \frac{1}{n} \nabla f(\bar{x}_k^1) - g_k + g_k - s_{i,k} \right)/2
\]

Further using Lemma 5 and Lemma 6 yields
\[
\| \frac{1}{n} \nabla f(\bar{x}_k^1) - g_k \|_2 \leq \frac{1}{n} \sum_{i=1}^{n} \| \nabla f_i(\bar{x}_k^1) - \nabla f_i(x_k^1) \|_2
\]
\[
\leq L \sum_{i=1}^{n} \| x_k^1 - x_k^2 \|_2 \leq L \frac{a_k C_p}{\sqrt{n}}
\]

and \( \sum_{i=1}^{n} \| g_k - s_{i,k} \|_2 \leq \sqrt{n} C_g \), and therefore
\[
\sum_{i=1}^{n} a_k \left( \frac{1}{n} \nabla f(\bar{x}_k^1) - s_{i,k}, x_k + x_{i,k}^1 - x^* \right) \leq D \frac{a_k^2}{A_k} \frac{LC_p + nC_g}{\sqrt{n}}.
\]

Finally, by collectively substituting (31), (32), and (33) into (30), we get
\[
A_t \left( f(\bar{x}_t^2) - f(x^*) \right)
\]
\[
\leq \left( \frac{D(1+1/\gamma) + (2 + \gamma + \frac{3}{\gamma})C_p^2}{2(1 + 2\gamma)} \right) \frac{t}{A_k} \sum_{k=1}^{t} a_k^2
\]
\[
+ \frac{t}{2} \sum_{k=1}^{t} a_k \left( \frac{1}{n} \nabla f(\bar{x}_k^1), s_{i,k}^1, x_k + x_{i,k}^1 - x^* \right)
\]
\[
\|x^*\| + \sum_{i=1}^{n} \sum_{k=1}^{t} \left( \frac{a_k^2}{A_k} \frac{L(1+2\gamma)}{2} - 1 \right) \| x_k^1 - x_{i,k} - x_k + x_{i,k} - x^* \|_2.
\]

Based on the condition in (28) and the fact that \( \frac{a_k^2}{A_k} \leq 2a \), the result in (29) follows. This completes the proof. \( \square \)

**Remark 2.** Theorem 2 proves that an \( O(1/\sqrt{n} + 1/\gamma) \) rate can be guaranteed if the parameter \( \gamma \) satisfies the condition in (28). The condition does not rely on the mixing matrix and therefore makes itself easy to satisfy in practice. In the error
bound [29], the first term matches the $O\left(\frac{1}{\sqrt{t}}\right)$ rate in centralized accelerated methods, and the second term represents the penalty due to the heterogeneity of local objective functions, which converges at a slower $O\left(\frac{1}{t}\right)$ rate. Note that this overall rate matches the result in [13] for decentralized unconstrained optimization. Regarding the rate constant, the one for the first term in the right-hand side of [29] is also observed in centralized accelerated methods [29]. By definitions of $C_p$ and $C_g$, we note that the rate constant in the second term is inversely proportional to $\sigma_2(P)$, whose value usually depends on the algebraic connectivity of the graph [21], implying that there is a tradeoff between convergence speed and communication cost in ADDA.

To conclude this section, an illustration of the theoretical results and how they relate to each other is presented in Fig. 4.

V. SIMULATION

In this section, we verify the proposed methods by applying them to solve a large-scale LASSO problem in a decentralized setting. In the simulation, each agent $i \in \mathcal{V}$ has access to a local data tuple $(y_i, A_i)$ satisfying the following equation

$$y_i = A_i x^* + b_i,$$

where $A_i \in \mathbb{R}^{n \times m}$, $y_i \in \mathbb{R}^n$, $x^* \in \mathbb{R}^m$ is a sparse vector, and $b_i \in \mathbb{R}^m$ is the additive Gaussian noise with zero mean and certain variance. To recover $x^*$, the following decentralized optimization problem is constructed

$$\min_{x \in \mathbb{R}^m} \sum_{i=1}^{n} \frac{1}{2} \|y_i - A_i x\|_2^2, \quad \text{s.t.} \|x\|_1 \leq R.$$

In the simulation, the parameters are set as: $n = 50$, $m = 10000$, $p_i = 20, \forall i \in \mathcal{V}$. Following [25], the data is generated in the following way. First, the matrix $A_i$ is produced randomly with $\mathcal{N}(0, 1)$ elements. The minimizer $x^*$ is a sparse vector that only has 50 non-zero $\mathcal{N}(0, 1)$ entries. The variance for noise $b_i, \forall i \in \mathcal{V}$ is set as 0.01. Set $R = 1.1 \cdot \|x^*\|_1$. The communication network is characterized by an Erdos-Renyi graph, where the probability for a connection between any two agents is set as 0.1. The doubly stochastic matrix $P$ associated with the graph is determined based on the Metropolis-Hastings rule [30]. For comparison, the PG-EXTRA method in [15] and the APM method in [21] are simulated. The free parameters for them are manually tuned to achieve the best performance. In particular, the stepsize for PG-EXTRA is set as $10^{-4}$; for APM, $L = 250$ and another parameter is calculated following $\beta_0 = \frac{L}{\sqrt{1 - \lambda_2(P)}}$. For FDDA and ADDA in this work, control sequences $a = 5 \cdot 10^{-4}$ and $a_t = (t + 1) \cdot 10^{-4}$ are used, respectively. In addition to that, a simple prox-function $\frac{1}{2} \|x\|_2^2$ is employed for them. We further consider a stochastic setting to demonstrate the efficiency of FDDA, where all the communication links in the original graph stay active with probability 0.4. The $l_1$ projection is performed by using the algorithm in [35]. The initial primal variables for all methods are set as $x_{i,0} = 0, \forall i \in \mathcal{V}$.

The convergence behaviors of four algorithms are displayed in Figs. 5 and 6. In Fig. 5, the convergence is evaluated in terms of the objective function value, while the primal residual $\|x_t - 1 \otimes x^*\|_2$ is considered in Fig. 6. The results suggest that ADDA converges faster than APM, primarily because the weight for the gradient information gradually decreases in APM. For methods with constant stepsizes, FDDA in both deterministic and stochastic settings outperforms PG-EXTRA.

To compare performances of ADDA in different network configurations, we consider networks with connectivity ratio $r = 0.2, 0.4$ and 0.6, where $r$ is defined as the actual number of links divided the number of all possible links. Note that the same control sequence $a_t$ is considered. The results are plotted in Figs. 7 and 8. It can be observed from them that with a higher connectivity, ADDA converges slightly faster. This is in line with Theorem 2 that although the sufficient condition for stepsize does not rely on the graph, a sparser network can give rise to a larger consensus error and therefore a larger error bound.
works include extensions of ADDA to unconstrained problems and is applicable to problems defined over stochastic networks, provided that a sufficient condition for the stepsize, the weight matrix is satisfied. The second method ADDA has an $O(\frac{1}{k^2})$ convergence rate of $O(\frac{1}{k})$ and is applicable to problems defined over stochastic networks, provided that a sufficient condition for the stepsize, Lipschitz modulus of the gradient, and the spectral radius of the weight matrix is satisfied. The second method ADDA has an $O(\frac{1}{k^2})$ convergence rate, where the condition on the stepsize only depends on the Lipschitz constant of the gradient. In contrast to FDDA, the dependence on network connectivity is reflected by the rate constant in ADDA. Future works include extensions of ADDA to unconstrained problems and the case with stochastic communication.

**VI. Conclusion**

In this work, we have designed two algorithms for decentralized constrained optimization, each of which involves two rounds of communication and a projection step. The first method, abbreviated as FDDA, has a convergence rate of $O(\frac{1}{k})$ and is applicable to problems defined over stochastic networks, provided that a sufficient condition for the stepsize, Lipschitz modulus of the gradient, and the spectral radius of the weight matrix is satisfied. The second method ADDA has an $O(\frac{1}{k^2})$ convergence rate, where the condition on the stepsize only depends on the Lipschitz constant of the gradient. In contrast to FDDA, the dependence on network connectivity is reflected by the rate constant in ADDA. Future works include extensions of ADDA to unconstrained problems and the case with stochastic communication.

**APPENDIX A**

**PROOF OF LEMMA**

Define

$$m_k(x) = \sum_{i=0}^{k} a_i \langle v_i, x \rangle + d(x).$$

We then have

$$m_k(x) = m_{k-1}(x) + a_k \langle v_k, x \rangle$$

According to the definition of Bregman divergence, we have

$$D_{m_{k-1}}(w_k - w_{k-1}) = m_{k-1}(w_k) - m_{k-1}(w_{k-1}) - \langle \nabla m_{k-1}(w_{k-1}), w_k - w_{k-1} \rangle$$

which is equivalent to

$$D_d(w_k - w_{k-1}) = m_{k-1}(w_k) - m_{k-1}(w_{k-1}) - \langle \nabla m_{k-1}(w_{k-1}), w_k - w_{k-1} \rangle.$$ 

Since

$$w_{k-1} = \arg \min_{x \in X} m_{k-1}(x),$$

by the optimality condition we have

$$\langle \nabla m_{k-1}(w_{k-1}), w_k - w_{k-1} \rangle \geq 0.$$ 

Coupled with the above inequality, using the fact that

$$D_d(w_k - w_{k-1}) \geq \frac{1}{2} \|w_k - w_{k-1}\|^2$$

gives rise to

$$0 \leq m_{k-1}(w_k) - m_{k-1}(w_{k-1}) - \frac{1}{2} \|w_k - w_{k-1}\|^2$$

$$= m_k(w_k) - a_k \langle v_k, w_k \rangle - m_{k-1}(w_{k-1}) - \frac{1}{2} \|w_k - w_{k-1}\|^2$$

which is equivalent to

$$a_k \langle v_k, w_k \rangle \leq m_k(w_k) - m_{k-1}(w_{k-1}) - \frac{1}{2} \|w_k - w_{k-1}\|^2.$$
Summing the above equation over $k$ from 1 to $t$ yields

$$\sum_{k=1}^{t} \langle a_k v_k, w_k \rangle \leq m_t(w_t) - m_0(w_0) - \sum_{k=1}^{t} \frac{1}{2} \|w_k - w_{k-1}\|^2$$

(34)

We turn to consider

$$\sum_{k=1}^{t} \langle a_k v_{k*}, -x^* \rangle \leq \max_{x \in X} \left\{ \sum_{k=1}^{t} a_k \langle v_{k*}, -x \rangle - d(x) \right\} + d(x^*)$$

$$= -\min_{x \in X} \left\{ \sum_{k=1}^{t} a_k \langle v_{k*}, x \rangle + d(x) \right\} + d(x^*)$$

$$= -m_t(w_t) + d(x^*),$$

which in conjunction with (34) gives rise to the inequality in (12), thereby concluding the proof.

**Appendix B**

**Proof of Lemma 1**

Let $z_{k+1} = \sum_{t=0}^{k} a_t h_t$. Since $s_0 = h_0 = v_0$, from (9) we have

$$z_k = Pz_{k-1} + as_{k-1}.$$  

By subtracting $1 \sum_{t=0}^{k-1} a_t g_t$ on both sides and the triangle inequality, we get

$$\|z_k - 1 \sum_{t=0}^{k-1} a_t g_t\| \leq \|Pz_{k-1} - 1 \sum_{t=0}^{k-2} a_t g_t\| + \|s_{k-1} - 1 g_{k-1}\|$$

(35)

Similarly, it holds that

$$\|s_{k} - 1 g_{k}\|$$

$$= \|Ps_{k-1} - 1 g_{k-1} + \nabla_k - \nabla_{k-1} - 1 g_{k} + 1 g_{k-1}\|$$

(36)

where the fact

$$\|\nabla_k - \nabla_{k-1} - 1 g_{k} + 1 g_{k-1}\| \leq \|\nabla_k - \nabla_{k-1}\|$$

and the Lipschitz continuity of the gradient are used to get the last inequality. Using Lemma 1 over $\|x_k - 1 y_k\|$ and $\|x_{k-1} - 1 y_{k-1}\|$, and (35) allows us to further get

$$\|s_{k} - 1 g_{k}\| \leq (\beta + L) \|s_{k-1} - 1 g_{k-1}\| + \sqrt{\|x_k - 1 y_k\|^2 + \sqrt{\|x_{k-1} - 1 y_{k-1}\|^2}}$$

(37)

From (35) and (37), the following linear system inequality can be established:

$$\left\|z_k - 1 \sum_{t=0}^{k-1} a_t g_t\right\| \leq M(a) \left\|z_{k-1} - 1 \sum_{t=0}^{k-2} a_t g_t\right\| + \sqrt{n} \left\|y_k - y_{k-1}\right\|.$$  

(38)

Since $h_0 = s_0 = v_0 = 1 g_0$ by initialization, it holds that

$$\left\|z_k - 1 \sum_{t=0}^{k-1} a_t g_t\right\| \leq \sqrt{n} \sum_{j=1}^{k} \rho(M(a))^{k-j} \left\|y_j - y_{j-1}\right\|.$$  

(39)

It is easy to check that the eigenvalues of $M(a)$ are

$$\frac{2 \beta + aL}{\sqrt{a^2 L^2 + 4(\beta + 1) aL}}.$$  

Since $\rho(M(a)) < 1$, one readily has $aL < \beta + 1$. Then, according to (36),

$$\|z_k - 1 \sum_{t=0}^{k-1} a_t g_t\| \leq \sqrt{n} \sum_{j=1}^{k} \rho(M(a))^{k-j} \left\|y_j - y_{j-1}\right\|.$$  

(40)

where $\lambda_1 > \lambda_2$ are eigenvalues of $M(a)$. Therefore, by invoking Lemma 1 we get

$$\sum_{k=1}^{t} \|x_k - 1 y_k\|^2 \leq n \sum_{k=1}^{t} \left( \sum_{j=1}^{k} \rho(M(a))^{k-j} \left\|y_j - y_{j-1}\right\|^2 \right)^2$$

$$\leq n \sum_{k=1}^{t} \left( \sum_{j=1}^{k} \rho(M(a))^{k-j} \left\|y_j - y_{j-1}\right\|^2 \right)^2$$

$$\leq n \sum_{k=1}^{t} \frac{1}{\left(1 - \rho(M(a))\right)} \sum_{j=1}^{k} \rho(M(a))^{k-j} \left\|y_j - y_{j-1}\right\|^2$$

$$\leq \frac{n}{\left(1 - \rho(M(a))\right)} \sum_{j=1}^{t} \left\|y_j - y_{j-1}\right\|^2.$$  

The proof is completed.
Appendix C

Proof of Corollary 2

Following a similar procedure as in [16], we make use of the convexity and smoothness of the objective to get (I) and (II) in the following:

\[ f(x_{i,t}) - f(y_k) \leq \sum_{j=1}^{n} \left( f_j(x_{j,k}) - \langle \nabla f_j(x_{j,k}), y_k - x_{j,k} \rangle - f_j(x_{j,k}) \right) \]

\[ \leq \sum_{j=1}^{n} \left( \langle \nabla f_j(x_{j,k}), x_{j,k} - x_{j,k} \rangle - \langle \nabla f_j(x_{j,k}), y_k - x_{j,k} \rangle \right) + \frac{L}{2} \| x_{i,k} - y_k + y_k - x_{j,k} \|^2 \]

\[ = \sum_{j=1}^{n} \left( \| \nabla f_j(x_{j,k}) \|_2^2 + \frac{L}{2} \| y_k - x_{j,k} \|^2 \right) + nL \| x_{i,k} - y_k \|^2 + L \| y_k - x_{k} \|^2. \]

(41)

According to the optimality condition, we can simplify the iteration in (6) and (13) as

\[ x_{i,t} = -a \sum_{k=1}^{t-1} s_{i,k}, \quad y_{t} = -a \sum_{k=1}^{t-1} g_{k}. \]

Using Lemma 2 allows us to get \( y_{t} = \sum_{i=1}^{n} x_{i,t} \). Therefore, summing (41) over \( i \) from \( i = 1 \) to \( n \) and then over \( k \) from \( k = 1 \) to \( t \) yields

\[ \sum_{k=1}^{t} \sum_{i=1}^{n} \left( f(x_{i,k}) - f(y_k) \right) \leq 2nL \sum_{k=1}^{t} \| 1y_k - x_k \|^2, \]

which along with Lemma 4 and (18) further gives rise to

\[ aT \left( f(x), f(x^*) \right) \leq aT \sum_{i=1}^{n} \left( f(x_{i,k}) - f(x^*) \right) \]

\[ \leq \frac{n^2 aT}{2} \left( 1 + \tau + \frac{5 + \frac{1}{a} - \rho(M(a))}{(1 - \rho(M(a)))^2} - \frac{1}{aT} \right) \sum_{k=1}^{t} \| y_k - y_{k-1} \|^2 \]

\[ + \frac{n^2 \| x \|^2}{2}. \]

For \( \tau \) satisfying (23), the inequality in (24) holds true.

Appendix D

Proof of Lemma 5

We begin by observing the definitions that

\[ \frac{a_t}{A_t} \cdot t = \frac{2(t + 1)}{t + 3} \geq 1. \]

Since both \( x^{[1]}_{t+1} \) and \( \pi^t \) are within the constraint set, the inequalities in (26) and (25) hold for

\[ 1 \leq t \leq \left[ \frac{3}{1 - \beta} \right]. \]

Next, for the sake of induction, we assume for some \( t \geq \left[ \frac{3}{1 - \beta} \right] \) that (26) and (25) are satisfied.

1) Upper bound for \( \| x_{t+1} - \pi_{t+1} \|^2 \): According to (7) and (11), one has

\[ \| x_{t+1} - \pi_{t+1} \|^2 \leq \left[ \frac{2a_t}{A_t} \right] \left( \| x_{t+1} - \pi_{t+1} \|^2 + \frac{a_{t+1}}{A_{t+1}} \| x_{t+1} - \pi_{t+1} \| \right). \]

where in (I) the hypothesis that \( \| x_{t+1} - \pi_{t+1} \|^2 \leq \frac{a_t}{A_t} C_p \) and the bound of the constraint set in Assumption 4 are used. Since by definition \( \sqrt{n}D = \frac{C_p}{\| x \|} \), we are able to further get

\[ \| x_{t+1} - \pi_{t+1} \|^2 \leq \frac{a_t}{A_t} C_p \left( 1 + \frac{1}{3 - \beta} \right). \]

It then remains to prove that

\[ \left( 1 + \frac{1}{3 - \beta} \right) \leq \frac{a_t}{A_t} \left( 1 + \frac{1}{3 - \beta} \right), \quad \forall t \geq t_0. \]

To see this, we first define

\[ t_0 = \left[ \frac{3}{1 - \beta} \right], \]

which gives us that

\[ \frac{3}{t_0} \leq 1 - \beta. \]

By algebraic manipulations, we are able to get

\[ \beta + 1 \leq t_0 - \frac{2}{t_0} \leq t_0 + \frac{2}{t_0} \leq t_0 + 4. \]

This in conjunction with the definitions of \( a_t \) and \( A_t \) yields

\[ \beta + 1 \leq \frac{t_0}{t_0 + 2} \leq \frac{t_0 + 2}{t_0 + 4} \cdot \frac{t_0 + 3}{(t_0 + 1)(t_0 + 1)} = \frac{a_t}{A_t} \left( 1 + \frac{1}{3 - \beta} \right), \]

where the fact that \( t_0(t_0 + 3) \leq 1 \) is used in (b). Note that \( \frac{a_t}{A_t} \left( 1 + \frac{1}{3 - \beta} \right) \) is monotonically increasing over \( t \). This implies that (42) holds true, and therefore

\[ \| x_{t+1} - \pi_{t+1} \|^2 \leq \frac{a_t}{A_t} C_p. \]

2) Upper bound for \( \| x_{t+1} - \pi_{t+1} \|^2 \): By the iteration rule in (7) and (11), we have

\[ x_{t+1} = \frac{A_t}{A_{t+1}} \left( px_{t+1} - \pi_{t+1} \right) + \frac{a_{t+1}}{A_{t+1}} \left( x_t - \pi_t \right). \]

By following the same line of reasoning, we are able to obtain

\[ \| x_{t+1} - \pi_{t+1} \|^2 \leq \frac{a_t}{A_t} C_p. \]

By jointly considering the above bounds, the proof is completed.
According to (7), we have
\[
\|s_{t+1} - 1g_{t+1}^{[1]}\|_2^2 = \|Ps_t - 1g_{t+1}^{[1]} + \nabla_t - 1g_{t+1}^{[1]}\|_2^2 \\
\leq \|Ps_t - 1g_{t+1}^{[1]}\|_2^2 + \|\nabla_t - 1g_{t+1}^{[1]} - 1g_{t+1}^{[1]}\|_2^2
\]
By smoothness of the objective, it holds
\[
\|\nabla_t - 1g_{t+1}^{[1]} - 1g_{t+1}^{[1]}\|_2 \leq \nabla_t - 1g_{t+1}^{[1]} \|
\]
To bound \(\|x_{t+1} - x_t\|_2\), we consider
\[
\|x_{t+1} - x_t\|_2 = \|\nabla_t x_t - x_t\|_2 \\
= \|\frac{A_t}{A_{t+1}} P\nabla_t x_t^2 + \frac{\alpha_{t+1}}{A_{t+1}} x_t - x_t\|_2 \\
\leq \|\frac{A_t}{A_{t+1}} P\nabla_t x_t^2 + \frac{\alpha_{t+1}}{A_{t+1}} x_t - x_t\|_2 + \frac{\alpha_{t+1}}{A_{t+1}} \|x_t - x_t\|_2 \\
\leq \frac{\alpha_{t+1}}{A_{t+1}} \|\nabla_t x_t^2 + \frac{\alpha_{t+1}}{A_{t+1}} x_t - x_t\|_2 + \frac{\alpha_{t+1}}{A_{t+1}} \|x_t - x_t\|_2 \\
\leq \frac{\alpha_{t+1}}{A_{t+1}} (\|\nabla_t x_t^2 + \frac{\alpha_{t+1}}{A_{t+1}} x_t - x_t\|_2 + \|x_t - x_t\|_2)
\]
where (I) is due to the iteration rule in (7), and (II) is derived using the bound for \(\|x_t^2 - 1x_t^2\|_2\) in Lemma 5 and the bound of \(\nabla_t\) in Assumption 4. By collectively considering the above relations, it holds
\[
\|s_{t+1} - 1g_{t+1}^{[1]}\|_2 \leq \beta \|s_t - 1g_{t}^{[1]}\|_2 + L \frac{\alpha_{t+1}}{A_{t+1}} (2\sqrt{n}D + 2C_P).
\]
By initialization, we have \(s_0 = \nabla_{t_0}^{[1]} = 1g_{t_0}^{[1]}\) and therefore
\[
\|s_{t_0} - 1g_{t_0}^{[1]}\|_2 \leq \beta \|s_t - 1g_{t}^{[1]}\|_2 + L \frac{\alpha_{t+1}}{A_{t+1}} (2\sqrt{n}D + 2C_P)
\]
Assume for some \(t \geq \frac{3}{1-\beta}\) that (27) holds true. Then according to (44), we have
\[
\|s_{t+1} - 1g_{t+1}^{[1]}\|_2 \leq \beta \frac{\alpha_{t+1}}{A_{t+1}} C_g + L \frac{\alpha_{t+1}}{A_{t+1}} (2\sqrt{n}D + 2C_P)
\]
By using the same argument (42) in the proof of Lemma 5, we are able to get
\[
\|s_{t+1} - 1g_{t+1}^{[1]}\|_2 \leq \frac{\alpha_{t+1}}{A_{t+1}} C_g
\]
thereby concluding the proof.

REFERENCES
[1] A. Nedić, A. Olshevsky, and M. G. Rabbat, “Network topology and communication-computation tradeoffs in decentralized optimization,” Proceedings of the IEEE, vol. 106, no. 5, pp. 953–976, 2018.
[2] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” Annual Reviews in Control, 2019.
[3] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” SIAM Journal on Optimization, vol. 26, no. 3, pp. 1835–1854, 2016.
[4] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, 2009.
[5] 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.
[6] M. Zhu and S. Martínez, “Discrete-time dynamic average consensus,” Automatica, vol. 46, no. 2, pp. 322–329, 2010.
[7] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, “Newton-raphson consensus for distributed convex optimization,” IEEE Transactions on Automatic Control, vol. 61, no. 4, pp. 994–1009, 2015.
[8] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1245–1260, 2017.
[9] ——, “Accelerated distributed nesterov gradient descent,” IEEE Transactions on Automatic Control, 2019.
[10] D. Jakovetić, “A unification and generalization of exact distributed first-order methods,” IEEE Transactions on Signal and Information Processing over Networks, vol. 5, no. 1, pp. 31–46, 2018.
[11] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the admn in decentralized consensus optimization,” IEEE Transactions on Signal Processing, vol. 62, no. 7, pp. 1750–1761, 2014.
[12] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, “A dual approach for optimal algorithms in distributed optimization over networks,” Optimization Methods and Software, pp. 1–40, 2020.
[13] J. Xu, Y. Tian, Y. Sun, and G. Scutari, “Accelerated primal-dual algorithms for distributed smooth convex optimization over networks,” arXiv preprint arXiv:1910.10666, 2019.
[14] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Transactions on Automatic Control, vol. 55, no. 4, pp. 922–938, 2010.
[15] W. Shi, Q. Ling, G. Wu, and W. Yin, “A proximal gradient algorithm for decentralized composite optimization,” IEEE Transactions on Signal Processing, vol. 63, no. 22, pp. 6013–6023, 2015.
[16] H. Li, C. Fang, W. Yin, and Z. Lin, “A sharp convergence rate analysis for distributed accelerated gradient methods,” arXiv preprint arXiv:1810.01053, 2018.
[17] M. Rabbat, “Multi-agent mirror descent for decentralized stochastic optimization,” in 2015 IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP). IEEE, 2015, pp. 517–520.
[18] S. Shahrampour and A. Jadbabaie, “Distributed online optimization in dynamic environments using mirror descent,” IEEE Transactions on Automatic Control, vol. 63, no. 3, pp. 714–725, 2017.
[19] D. Yuan, Y. Hong, D. W. Ho, and G. Jiang, “Optimal distributed stochastic mirror descent for strongly convex optimization,” Automatica, vol. 90, pp. 196–203, 2018.
[20] J. C. Duchi, X. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 592–606, 2011.
[21] S. Liu, P. Y. Chen, and A. O. Hero, “Accelerated distributed dual averaging over evolving networks of growing connectivity,” IEEE Transactions on Signal Processing, vol. 66, no. 7, pp. 1845–1859, 2018.
[22] C. Liu, H. Li, and Y. Shi, “A unitary distributed subgradient method for multi-agent optimization with different coupling sources,” Automatica, vol. 114, no. 108834, 2020.
[23] S. Liang, G. Yin et al., “Dual averaging push for distributed convex optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, 2019.

[24] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, 1983.

[25] H.-T. Wai, J. Lafond, A. Scaglione, and E. Moulines, “Decentralized frank–wolfe algorithm for convex and nonconvex problems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, 2017.

[26] Z. Li and M. Yan, “A primal-dual algorithm with optimal stepsizes and its application in decentralized consensus optimization,” *arXiv preprint arXiv:1711.06785*, 2017.

[27] P. Latafat, L. Stella, and P. Patrinos, “New primal-dual proximal algorithm for distributed optimization,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 1959–1964.

[28] C. Liu, H. Li, and Y. Shi, “Resource-aware exact decentralized optimization using event-triggered broadcasting,” *arXiv preprint arXiv:1907.10179*, 2019.

[29] M. Cohen, J. Diakonikolas, and L. Orecchia, “On acceleration with noise-corrupted gradients,” in *International Conference on Machine Learning*, 2018, pp. 1019–1028.

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

[31] D. P. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[32] Y. Nesterov, “Primal-dual subgradient methods for convex problems,” *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2009.

[33] H. Lu, R. M. Freund, and Y. Nesterov, “Relatively smooth convex optimization by first-order methods, and applications,” *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 333–354, 2018.

[34] S. Kar and J. M. Moura, “Sensor networks with random links: Topology design for distributed consensus,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, 2008.

[35] L. Condat, “Fast projection onto the simplex and the l1 ball,” *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, 2016.

[36] K. S. Williams, “The nth power of a 2×2 matrix,” *Mathematics Magazine*, vol. 65, no. 5, pp. 336–336, 1992.