A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization

Wotao Yin[†], Xianghui Mao[◇], Kun Yuan[*‡], Yuantao Gu[◇], and Ali H. Sayed[‡]

***Abstract*— This paper addresses consensus optimization problems in a multi-agent network, where all the agents collaboratively find a common minimizer to the sum of their private functions. Our goal is to develop a decentralized algorithm in which there is no center agent and each agent only communicates with its neighbors.**

State-of-the-art decentralized algorithms for consensus optimization with convex objectives use fixed step sizes but involve communications among either *all*, or a *random subset*, of the agents at each iteration. Another approach is to employ a *random walk incremental* strategy, which activates a succession of nodes and their links, only one node and one link each time; since the existing algorithms in this approach require diminishing step sizes to converge to the optimal solution, its convergence is relatively slow.

In this work, we propose a random walk algorithm that uses a fixed step size and converges faster than the existing random walk incremental algorithms. It is also communication efficient. We derive our algorithm by modifying ADMM and analyze its convergence. We establish linear convergence for least squares problems, along with a state-of-the-art communication complexity. Numerical experiments verify our analyses.

I. Introduction

Consider a directed graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the set of agents and E is the set of m edges. We aim to solve the following optimization problem:

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where each f_i is locally held by agent i. An algorithm is decentralized if it relies only on communications between immediate (one-hop) neighbors; there is no central node that collects or distributes information to the agents. Decentralize consensus optimization finds applications in various areas including wireless sensor networks [1], [2], multi-vehicle and multi-robot control systems [3], [4], smart grid implementations, distributed adaptation and estimation [5], [6], distributed statistical learning [7]–[9] and clustering [10].

This work was supported in part by NSF grants CCF-1524250, DMS-1720237, ECCS-1407712, NSFC grants 61531166005 and 61571263, and ONR grant N000141712162.

†W. Yin is with the Mathematics Department, University of California, Los Angeles, CA 90095. Email: wotaoyin@math.ucla.edu

◇X. Mao and Y. Gu are with the Electrical Engineering Department, Tsinghua University, Beijing, China. Email: maoxh14@mails.tsinghua.edu.cn, gyt@tsinghua.edu.cn.

*K. Yuan is with the Electrical and Computer Engineering Department, University of California, Los Angeles, CA 90095. Email:kunyuan@ucla.edu.

‡A. H. Sayed is with the Ecole Polytechnique Federale de Lausanne (EPFL), School of Engineering, CH-1015 Lausanne, Switzerland. Email: ali.sayed@epfl.ch. K. Yuan is also with EPFL as a visiting student.

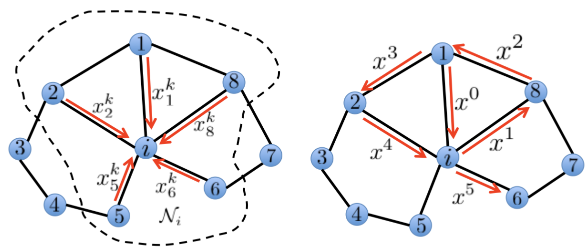

Fig. 1: Left: each agent communicates with all neighbors. Right: variable x is updated along the random walk $(1, i, 8, 1, 2, i, 6, \cdots)$.

A. The literature

There are several decentralized numerical approaches to solve problem (1). One well-known approach lets *every* agent exchange information with all, or a random subset, of its direct neighbors per iteration. This is illustrated in the left plot of Fig. 1, where agent i is collecting information from all its neighbors (to update its local variables). This approach includes well-known algorithms such as diffusion [5], [6] and consensus [11], [12], distributed ADMM (D-ADMM) [13]–[15], EXTRA [16], [17], DIGing [18], exact diffusion [19], and beyond. While these algorithms have good convergence rates in the number of iterations (D-ADMM, EXTRA, DIGing, and exact diffusion all converge linearly to the exact solution assuming strong convexity and using constant step-sizes), the amount of communication per iteration is relatively high. Depending on the density of the network, this approach costs $O(n)$ computation and $O(n)$–$O(n^2)$ communications per iteration.

Another approach is based on the (random) walk (sub)gradient method [20]–[22], where a variable x will move through a (random) succession of agents in the network. At each iteration, the agent that receives x updates it using a subgradient of f_i at x, followed by sending x to one (random) neighbor. The right plot in Fig. 1 illustrates the updates of x^k along a walk $(1, i, 8, 1, 2, i, 6, \cdots)$. Since only one node and one link are used at each iteration. this approach only costs $O(1)$ computation and $O(1)$ communication per iteration. Another line of work [23], [24] also takes random walks but deals with stochastic gradients and adaptive networks in order to track objects and adapt to drifts. With constant step-sizes, these algorithms improve their iterates (linearly or sublinearly under respective assumptions) to a *solution neighborhood* and do not converge to an exact solution. If the step-size is small, the neighborhood will be proportionally small but convergence becomes more slowly. For applications where convergence to the exact solution is required, decaying step-sizes must be used, which lead to slow convergence.

B. Proposed work

In this paper, we propose a new random walk algorithm for decentralized consensus optimization that uses a *fixed step-size* and converges to the *exact solution*. It is significantly faster than the existing random-walk subgradient incremental methods, though in this conference paper, we limit our effort in this paper to just convex functions and a static network. For decentralized least squares specifically, we establish a linear convergence rate and total communication complexity. The latter is roughly at

$$O\big(\ln(1/\epsilon)\cdot(\ln(n)/(1-\lambda))^2\big), \qquad (2)$$

where ϵ is a target accuracy and λ is a certain connectivity measure. Let m be the number of edges in the network. We show that if

$$\lambda \le 1-\frac{\ln^{4/3}(n)}{m^{2/3}} \approx 1-\frac{1}{m^{2/3}}, \qquad (3)$$

which indicates at least a moderate connectivity, then our algorithm is more communication efficient than all the methods reviewed above. Our numerical simulations also support this claim.

C. Derivation and discussions

Our algorithm can be derived by modifying existing algorithms, for example, ADMM [25], [26] and PPG [27]. Since the reader is familiar with ADMM, we choose it as our start point. We call our new algorithm Walk ADMM (W-ADMM).

W-ADMM and the aforementioned D-ADMM introduce the same set of local variables for the agents: one y_i for each agent i. However, W-ADMM uses only n constraints, specifically $x = y_i$ for every node i, fewer than the m constraints in D-ADMM: $y_i = y_j$ for every edge (i,j). When information exchanges between the agents are subject to constraint topologies, W-ADMM has an advantage in that information of one agent (one y_i) is quickly passed through x immediately to all the other agents. D-ADMM has a disadvantage as its information must prorogate through (possibly long) chains of agents. Without resorting to graph spectral analysis, it is not difficult to see W-ADMM would run faster than D-ADMM if both formulations were solved by a standard ADMM algorithm (D-ADMM is a standard ADMM algorithm applied to a decentralized formulation).

Due to the communication restriction in decentralization, x and $y_1,\dots,y_n$ in W-ADMM cannot be updated together, so W-ADMM cannot be solved by a standard ADMM algorithm. We modify ADMM so that it updates (x, y_i) for only one i each time and the i-sequence follows a walk. The efficiency of W-ADMM, therefore, depends on how long it takes the walk to visit all the agents. This is known as the *cover time*. If W-ADMM only needs visit every agent once (which is the case when we apply W-ADMM to compute the mean of a set of numbers, one at each agent), the cover time is the total complexity of W-ADMM. For random walks and cover time, we refer the reader to Chapter 11 in [28].

For general problems, W-ADMM must visit each agent many times to converge. Its efficiency depends on how frequently all of the agents are visited. For a random walk, this can be described by the *mixing time* of the underlying Markov chain. We first present an assumption.

Assumption 1. *The random walk* $(i_k)_{k\ge 0}$, $i_k \in V$, *forms an irreducible and aperiodic Markov chain with transition probability matrix* $\mathbf{P}\in\mathbb{R}^{n\times n}$ *where* $[\mathbf{P}]_{ij} = p(i_{k+1} = j | i_k = i) \in [0,1]$ *and stationary distribution* π *satisfying* $\pi^T\mathbf{P} = \pi^T$.

If the underlying network is a complete graph, we can choose P so that $P_{ij} = p(i_{k+1} = j|i_k = i) = \frac{1}{n}$ for all $i,j \in V$, a case analyzed in [29, §2.6.1] (barring asynchronicity). For a more general network that is connected, we need the *mixing time* (for given $\delta > 0$), which is defined as the smallest integer $\mathcal{J}(\delta)$ such that, for all $i \in V$,

$$\big\|[\mathbf{P}^{\mathcal{J}(\delta)}]_{i,:} - \pi^\mathsf{T}\big\| \le \delta\pi_*, \qquad (4)$$

where $\pi_* := \min_{i\in V}\pi_i$, and $[\mathbf{P}^{\mathcal{J}(\delta)}]_{i,:}$ denotes the ith row of $\mathbf{P}^{\mathcal{J}(\delta)}$. This inequality states: regardless of current state i and time k, the probability of visiting each state j after $\mathcal{J}(\delta)$ more steps is $(\delta\pi_*)$-close to π_j, that is, for all $i,j\in V$,

$$\big|[\mathbf{P}^{\mathcal{J}(\delta)}]_{ij} - \pi_j\big| \le \delta\pi_*. \qquad (5)$$

A good reference for mixing time is [28]. The mixing time requirement, inequality (4), is guaranteed to hold for[1]

$$\mathcal{J}(\delta) := \left\lceil \frac{1}{1-\sigma(\mathbf{P})}\ln\frac{\sqrt{2}}{\delta\pi_*}\right\rceil \qquad (6)$$

for $\sigma(\mathbf{P}) := \sup\big\{\|f^\mathsf{T}\mathbf{P}\|/\|f\| : f^\mathsf{T}\mathbf{1} = 0, f\in\mathbb{R}^n\big\}$.

We will use inequality (5) to show the sufficient descent of a Lyapunov function L^k, which was introduced first in [30] and extended in [31] and we adopt. With existing techniques, we would only show $L^k \ge L^{k+1}$ and the sequence having a lower bound. But, for a random walk $(i_k)_{k\ge 0}$, which is neither essential cyclic nor i.i.d. random (except with complete graphs), we must use a new analytic technique, motivated by the recent paper [32], to apply (5) and establish convergence in terms of asymptotic stationarity.

The communication complexity bound of W-ADMM for decentralized least squares is given in term of $\sigma(\mathbf{P})$. This quantity also determines those of D-ADMM, EXTRA, and exact diffusion. Therefore, we can compare their communication complexities. For moderately well connected networks, we show in §V that the bound of W-ADMM is the lowest.

Even though D-ADMM, EXTRA, and exact diffusion use more total communications, the communications over different edges in each iteration are concurrent, so they *may* take less *total communication time*. However, this time will increase if different edges have different communication

[1]Here is a trivial proof. For any $k \ge 1$, by definition, it holds $[\mathbf{P}^k]_{i,:} - \pi^\mathsf{T} = \big([\mathbf{P}^{k-1}]_{i,:} - \pi^\mathsf{T}\big)\mathbf{P}$, and $\big([\mathbf{P}^k]_{i,:} - \pi^\mathsf{T}\big)\mathbf{1} = 0$. Hence, $\|[\mathbf{P}^k]_{i,:} - \pi^\mathsf{T}\| \le \|[\mathbf{P}^{k-1}]_{i,:} - \pi^\mathsf{T}\|\sigma(\mathbf{P}) \le \cdots \le \|\mathbf{I}_{i,:} - \pi^\mathsf{T}\|\sigma^k(\mathbf{P})$. We can bound $\|\mathbf{I}_{i,:} - \pi^\mathsf{T}\|^2 \le (1-\pi_i)^2 + \sum_{j\ne i}\pi_j^2 \le (1-\pi_*)^2 + (1-\pi_*)^2 = 2(1-\pi_*)^2$. Therefore, by ensuring $\sqrt{2}(\sigma(\mathbf{P}))^{\mathcal{J}(\delta)}(1-\pi_*) \le \delta\pi_*$, which simplifies to condition (6) by Taylor series, we guarantee (4) to hold.

latencies and bandwidths, and if synchronization overhead is included. In an ideal situation where every communication takes the same amount of time and synchronization has no overhead at all, W-ADMM is found to be slower in time, unsurprisingly.

Although this paper does not discuss data privacy, W-ADMM protects privacy better than diffusion, consensus, D-ADMM, etc., since the communication path is random and only the current approximate solution x^k is sent out by the active agent. It is difficult for an agent to track its neighbors.

II. Derivation of Walk ADMM

By defining

$$\mathcal{Y} := \mathrm{col}\{y_1, y_2, \cdots, y_n\} \in \mathbb{R}^{np},\ F(\mathcal{Y}) := \sum_{i=1}^{n} f_i(y_i), \quad (7)$$

we can compactly rewrite problem (1) as

$$\begin{aligned} \underset{x,\ \mathcal{Y}}{\text{minimize}} \quad & \frac{1}{n} F(\mathcal{Y}), \\ \text{subject to} \quad & \mathbb{1} \otimes x - \mathcal{Y} = 0, \end{aligned} \quad (8)$$

where $\mathbb{1} = [1\ 1 \ldots 1]^{\mathsf{T}} \in \mathbb{R}^n$ and $\otimes$ is the Kronecker product. The augmented Lagrangian for problem (8) is

$$L_\beta(x, \mathcal{Y}; \mathcal{Z}) := \frac{1}{n}\Big(F(\mathcal{Y}) + \langle \mathcal{Z}, \mathbb{1} \otimes x - \mathcal{Y}\rangle + \frac{\beta}{2}\|\mathbb{1} \otimes x - \mathcal{Y}\|^2\Big), \quad (9)$$

where $\mathcal{Z} := \mathrm{col}\{z_1, \cdots, z_n\} \in \mathbb{R}^{np}$ is the dual variable (Lagrange multipliers) and $\beta > 0$ is a constant parameter. The standard ADAM algorithm is an iteration that minimize $L_\beta(x, \mathcal{Y}; \mathcal{Z})$ in x and then $\mathcal{Y}$, followed by updating $\mathcal{Z}$. Applying it to problem (8) yields

$$x^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\Big(y_i^k - \frac{z_i^k}{\beta}\Big), \quad (10a)$$

for all $i \in V$ do in parallel:

$$y_i^{k+1} = \arg\min_{y_i}\Big\{f_i(y_i) + \frac{\beta}{2}\Big\|x^{k+1} + \frac{z_i^k}{\beta} - y_i\Big\|^2\Big\}, \quad (10b)$$

$$z_i^{k+1} = z_i^k + \beta(x^{k+1} - y_i^{k+1}). \quad (10c)$$

Note that step (10a) needs to collect information from all agents. To decentralize (10a), we limit steps (10b) and (10c) to update only y_{i_k} and z_{i_k}, keeping the remaining $\{y_i\}_{i \neq i_k}$, $\{z_i\}_{i \neq i_k}$ unchanged, arriving at W-ADMM:

$$x^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\Big(y_i^k - \frac{z_i^k}{\beta}\Big), \quad (11a)$$

$$y_i^{k+1} = \begin{cases} \arg\min_{y_i}\{f_i(y_i) + \frac{\beta}{2}\Big\|x^{k+1} + \frac{z_i^k}{\beta} - y_i\Big\|^2\}, & i = i_k, \\ y_i^k, & \text{o.w.}, \end{cases} \quad (11b)$$

$$z_i^{k+1} = \begin{cases} z_i^k + \beta(x^{k+1} - y_i^{k+1}), & i = i_k \\ z_i^k, & \text{o.w.}. \end{cases} \quad (11c)$$

W-ADMM initializes $\{y_i^0\}_{i=1}^n$ and $\{z_i^0\}_{i=1}^n$ so that

$$x^1 = \sum_{i=1}^{n}(y_i^0 - \frac{z_i^0}{\beta}) = 0. \quad (12)$$

We can satisfy (12) by setting $y_i^0 = 0$ and $z_i^0 = 0$ for any $i \in \{1, \cdots, n\}$. Since only variables y_{i_k} and z_{i_k} are updated at iteration k, step (10a) or (11a) for every next iteration now uses just local variables and is thus decentralized:

$$x^{k+2} = x^{k+1} + \big(y_{i_k}^{k+1} - \frac{z_{i_k}^{k+1}}{\beta}\big) - \big(y_{i_k}^k - \frac{z_{i_k}^k}{\beta}\big). \quad (13)$$

The decentralized implementation of W-ADMM is presented in Algorithm 1. The variable x^k is updated by agent i_k and passed as a token to its neighbor i_{k+1}.

Algorithm 1: Walk ADMM (W-ADMM)

Initialization: initialize y_i^0 and z_i^0 so that (12) holds;
Repeat for $k = 0, 1, 2, 3 \cdots$ until convergence
  agent i_k do:
    update y^{k+1} according to (11b);
    update z^{k+1} according to (11c);
    update x^{k+2} according to (13);
    send x^{k+2} via edge (i_k, i_{k+1}) to agent i_{k+1};
End

Remark. We can avoid solving a minimization problem in step (11b) by using the cheaper gradient descent, like in diffusion, consensus, EXTRA, DIGing, and exact diffusion. If f_i is differentiable, we replace (11b) with the update:

$$y_i^{k+1} = \begin{cases} x_i^{k+1} + z_i^k - \frac{1}{\beta}\nabla f_i(y_i^k), & i = i_k \\ y_i^k, & \text{o.w.} \end{cases} \quad (11b')$$

Compare to (11b), update (11b') saves computations but can cause more iterations and thus more total communications. One can choose between (11b) and (11b') based on computation and communication tradeoffs in applications. Since this paper focuses on communication-efficient methods, the analyses in later sections apply only to (11b).

III. Convergence

In this section we present convergence of W-ADMM. Below, we let $\mathcal{X} := \{(x, \mathcal{Y}) \in \mathbb{R}^p \times \mathbb{R}^{np} : x = y_i \text{ for } i \in V\}$ denote the set of feasible solutions to Problem (8), and $\mathcal{Y} := \{\mathcal{Y} \in \mathbb{R}^{np} : y_i = y_j \text{ for } i, j \in V\}$ be the projection of $\mathcal{X}$ to $\mathcal{Y}$. We make the following assumptions.

Assumption 2. *The objective function $F(\mathcal{Y})$ is bounded from below over $\mathcal{Y}$, and $F(\mathcal{Y})$ is coercive over $\mathcal{Y}$, that is, $F(\mathcal{Y}^k) \to \infty$ for any sequence $\mathcal{Y}^k \in \mathcal{Y}$ and $\|\mathcal{Y}^k\| \to \infty$.*

Assumption 2 is *not* over $\mathbb{R}^{np}$, and it is easy to satisfy.

Assumption 3. *Each $f_i(x)$ is L-Lipschitz differentiable, that is, for any $u, v \in \mathbb{R}^p$,*

$$\|\nabla f_i(u) - \nabla f_i(v)\| \le L\|u - v\|, \quad i = 1, \ldots, n. \quad (14)$$

We introduce the Lyapunov function based on (9):

$$L_\beta^k := L_\beta(x^k, \mathcal{Y}^k; \mathcal{Z}^k). \quad (15)$$

By Assumptions 1–3, we establish the descent of L_β^k.

Lemma 1. *Under Assumptions 2 and 3, for $\beta \ge 2L+2$, the iterates $(x^k, \mathcal{y}^k, \mathcal{z}^k)$ generated by W-ADMM* (11)*, or Algorithm 1, satisfy the following properties:*

1) $L_\beta^k - L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k) = \frac{\beta}{2}\|x^{k+1} - x^k\|^2$;
2) $L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k) - L_\beta^{k+1} \ge \frac{1}{n}\|\mathcal{y}^{k+1} - \mathcal{y}^k\|^2$;
3) $L_\beta^k - L_\beta^{k+1} \ge \frac{\beta}{2}\|x^{k+1} - x^k\|^2 + \frac{1}{n}\|\mathcal{y}^k - \mathcal{y}^{k+1}\|_2^2$;
4) $(L_\beta^k)_{k\ge 0}$ *is lower bounded and convergent.*

Proof. See the Appendix. ☐

Theorem 1. *Under Assumptions 1–3, for $\beta \ge 2L+2$, the gradient of L_β^k with respect to $(x^k, \mathcal{y}^k, \mathcal{z}^k)$, ∇L_β^k, satisfies*

$$\lim_{k\to\infty} \mathbb{E}\|\nabla L_\beta^k\| = 0. \tag{16}$$

Proof. The main proof idea is summarized as follows. We first define

$$g^k := \begin{bmatrix} \nabla_x L_\beta^{k+1} \\ \nabla_\mathcal{y} L_\beta^{k+1} \\ \nabla_\mathcal{z} L_\beta^{k+1} \end{bmatrix},\ q_v^k := \begin{bmatrix} \nabla_x L_\beta^{k+1} \\ \nabla_{y_v} L_\beta^{k+1} \\ \nabla_{z_v} L_\beta^{k+1} \end{bmatrix}, \tag{17}$$

where $v \in V$ is an agent index, and g^k is the gradient of L_β^{k+1}. Our next step establishes the following inequality

$$\mathbb{E}\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 \le C \sum_{t=k-\mathcal{J}(\delta)-1}^{k} \mathbb{E}\|x^t - x^{t+1}\|^2 + \mathbb{E}\|\mathcal{y}^t - \mathcal{y}^{t+1}\|^2, \tag{18}$$

where C is a positive constant. Lemma 1 implies the right-hand side of (18) converges to 0 and thus $\lim_{k\to\infty}\mathbb{E}\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 = 0$. Next, we prove the bound

$$\mathbb{E}\|g^{k-\mathcal{J}(\delta)-1}\|^2 \le C'\,\mathbb{E}\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 \tag{19}$$

where C' is a positive constant and reach the conclusion

$$\lim_{k\to\infty} \mathbb{E}\|g^k\|^2 = \lim_{k\to\infty} \mathbb{E}\|g^{k-\mathcal{J}(\delta)-1}\|^2 = 0. \tag{20}$$

The complete proof is given in the Appendix. ☐

IV. Linear convergence for Least Squares

In this section, we focus on the decentralize least-squares problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{n}\sum_{i=1}^n \frac{1}{2}\|\mathbf{A}_i y_i - b_i\|^2, \\ &\text{subject to} && y_1 = y_2 = \cdots = y_n = x, \end{aligned} \tag{21}$$

which is a special case (8) with local objective $f_i(y_i) := \frac{1}{2}\|\mathbf{A}_i y_i - b_i\|^2$ and gradient $\nabla f_i(y_i) = \mathbf{A}_i^\top(\mathbf{A}_i y_i - b_i)$. The Lipschitz constant L in Assumption 3 equals $\sigma_{\max}^* := \max_i \sigma_{\max}(\mathbf{A}_i^\top \mathbf{A}_i)$, where $\sigma_{\max}(\cdot)$ takes largest eigenvalue.

We apply W-ADMM (or Algorithm 1) starting from

$$y_i^0 = (\beta\mathbf{I} - \mathbf{A}_i^\top\mathbf{A}_i)^{-1}(\mathbf{A}_i^\top b_i),\ \forall i \in V, \tag{22}$$

$$z_i^0 = \nabla f_i(y_i^0) = \mathbf{A}_i^\top(\mathbf{A}_i y_i^0 - b_i),\ \forall i \in V, \tag{23}$$

where (22) is well defined for $\beta > \max_i \sigma_{\max}(\mathbf{A}_i^\top\mathbf{A}_i)$. This is to ensure $y_i^0 - z_i^0/\beta = 0$ and thus (12) for all $k \ge 0$.

We analyze the complexities of W-ADMM for problem (21) based on the Lyapunov function $h_\beta(\mathcal{y}) : \mathbb{R}^{np} \to \mathbb{R}$,

$$\begin{aligned} h_\beta(\mathcal{y}) := \frac{1}{n}\sum_{i=1}^n \Big(\frac{\beta}{2}\|y_i^k\|^2 - \frac{1}{2}\|\mathbf{A}_i y_i^k\|^2 + \frac{1}{2}\|b_i\|^2\Big) \\ - \frac{\beta}{2}\|\mathbf{T}\mathcal{y} + c\|^2, \end{aligned} \tag{24}$$

where $\mathbf{T} := \frac{1}{n}\big[(\mathbf{I} - \frac{1}{\beta}\mathbf{A}_1^\top\mathbf{A}_1), \ldots, (\mathbf{I} - \frac{1}{\beta}\mathbf{A}_n^\top\mathbf{A}_n)\big] \in \mathbb{R}^{p\times np}$ and $c := \frac{1}{n\beta}\sum_{i=1}^n \mathbf{A}_i^\top b_i \in \mathbb{R}^p$. The following lemma relates $h_\beta(\mathcal{y})$ and the augmented Lagrangian sequence.

Lemma 2. *With initialization* (22) *and* (23)*, it holds that*

$$h_\beta(\mathcal{y}^k) = L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k). \tag{25}$$

Proof. From the optimality condition of (11b), we can verify

$$\mathbf{A}_{i_k}^\top(\mathbf{A}_{i_k} y_{i_k}^{k+1} - b_{i_k}) = \beta x^{k+1} + z_{i_k} - \beta y_{i_k}^{k+1} \overset{\text{(a)}}{=} z_{i_k}^{k+1}, \tag{26}$$

for $k \ge 1$, where (a) follows from (11c). In W-ADMM, each pair of y_i and z_i is either updated together, or both not updated. Then by applying (23) and (26), we get

$$z_i^k = \mathbf{A}_i^\top(\mathbf{A}_i y_i^k - b_i),\ \forall i \in V,\ k \ge 0. \tag{27}$$

Substituting (27) into (11a) yields $x^{k+1} = \mathbf{T}\mathcal{y}^k + c, \quad \forall k \ge 0$. Eliminating z_i^k and x^{k+1} in $L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k)$ using the above formulas produces (25). ☐

The following lemma establishes that $h_\beta(\mathcal{y})$ is strongly convex and Lipschitz differentiable.

Lemma 3. *For a network with $n \ge 2$ agents, and the parameter $\beta > \sigma_{\max}^*$, where $\sigma_{\max}^* := \max_i \sigma_{\max}(\mathbf{A}_i^\top\mathbf{A}_i)$, the function $h_\beta(\cdot)$ is*

1) *strongly convex with modulus $\nu = \frac{(n-1)(\beta - \sigma_{\max}^*)}{n^2}$, and*
2) *Lipschitz differentiable with Lipschitz constant $\bar{L} = \frac{\beta}{n}\big(1 - \frac{1}{n}\big(1 - \frac{\sigma_{\max}^*}{\beta}\big)^2\big)$.*

Proof. As a quadratic function, $h_\beta(\cdot)$ is ν-strongly convex with $\bar{L}$-Lipschitz gradients if, and only if, its Hessian (by (24)) $\mathbf{H}$ satisfies

$$\nu\mathbf{I} \preceq \mathbf{H} := \frac{\beta}{n}\mathbf{I}_{np} - \frac{1}{n}\mathbf{A} - \beta\mathbf{T}^\top\mathbf{T} \preceq \bar{L}\mathbf{I}, \tag{28}$$

where $\mathbf{A} := \operatorname{diag}\left(\mathbf{A}_1^\top\mathbf{A}_1, \mathbf{A}_2^\top\mathbf{A}_2, \cdots, \mathbf{A}_n^\top\mathbf{A}_n\right)$. With $\beta > \max_i \sigma_{\max}(\mathbf{A}_i^\top\mathbf{A}_i)$, we define the symmetric positive definite matrices $\mathbf{D}_i := \left(\mathbf{I} - \frac{1}{\beta}\mathbf{A}_i^\top\mathbf{A}_i\right)^{1/2}$ for $i \in V$. The spectral norm of $\mathbf{D}_i$ satisfies

$$\left(1 - \frac{\sigma_{\max}^*}{\beta}\right)^{\frac{1}{2}} \le \left(1 - \frac{\sigma_{\max}(\mathbf{A}_i^\top\mathbf{A}_i)}{\beta}\right)^{\frac{1}{2}} \le \|\mathbf{D}_i\| \le 1. \tag{29}$$

Stacking $\mathbf{D}_i$'s into

$$\mathbf{D} := \begin{bmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_n \end{bmatrix}. \tag{30}$$

Then, for any vector $\mathcal{w} := \mathrm{col}\{w_1, \cdots, w_n\} \in \mathbb{R}^{np}$ where $w_i \in \mathbb{R}^p$, we have the interval bounds for $\|\mathrm{diag}(\mathbf{D})\mathcal{w}\|$:

$$\|\mathrm{diag}(\mathbf{D})\mathcal{w}\| = \left\| \begin{bmatrix} \mathbf{D}_1 w_1 \\ \vdots \\ \mathbf{D}_n w_n \end{bmatrix} \right\| \tag{31}$$

$$\in \Big[\Big(1 - \frac{\sigma^*_{\max}}{\beta}\Big)^{\frac{1}{2}} \|\mathcal{w}\|, \|\mathcal{w}\| \Big]. \tag{32}$$

It is easy to check

$$\mathcal{w}^\mathsf{T}\mathbf{H}\mathcal{w} = \frac{\beta}{n}(\mathrm{diag}(\mathbf{D})\mathcal{w})^\mathsf{T}\Big(\mathbf{I} - \frac{1}{n}\mathbf{D}^\mathsf{T}\mathbf{D}\Big)(\mathrm{diag}(\mathbf{D})\mathcal{w}). \tag{33}$$

Therefore, we get (28) from

$$\mathcal{w}^\mathsf{T}\mathbf{H}\mathcal{w} \geq \frac{\beta}{n}\big(1 - \frac{1}{n}\big)\|\mathrm{diag}(\mathbf{D})\mathcal{w}\|^2 \tag{34}$$

$$\geq \underbrace{\frac{\beta}{n}\big(1 - \frac{1}{n}\big)\big(1 - \frac{\sigma^*_{\max}}{\beta}\big)}_{\nu} \|\mathcal{w}\|^2 \tag{35}$$

and

$$\mathcal{w}^\mathsf{T}\mathbf{H}\mathcal{w} \leq \frac{\beta}{n}\Big(\|\mathrm{diag}(\mathbf{D})\mathcal{w}\|^2 - \frac{1}{n}\Big) \tag{36}$$

$$\leq \frac{\beta}{n}\Big(\|\mathcal{w}\|^2 - \frac{1}{n}\big(1 - \frac{\sigma^*_{\max}}{\beta}\big)^2\|\mathcal{w}\|^2\Big) \tag{37}$$

$$= \underbrace{\frac{\beta}{n}\big(1 - \frac{1}{n}\big(1 - \frac{\sigma^*_{\max}}{\beta}\big)^2\big)}_{\bar{L}} \|\mathcal{w}\|^2. \tag{38}$$

□

Lemma 4. *With $\beta > \sigma^*_{\max}$, the unique minimizer of $h_\beta(\cdot)$ is $\mathcal{y}^\star := \mathrm{col}\{y^\star_1, \cdots, y^\star_n\}$ with $y^\star_i \equiv x^\star = (\sum_{i=1}^n \mathbf{A}_i^\mathsf{T}\mathbf{A}_i)^{-1}(\sum_{i=1}^n \mathbf{A}_i^\mathsf{T} b_i)$. These components are also the unique solution to* (21)*, as well as the unique minimizer of $\sum_{i=1}^n \frac{1}{2}\|\mathbf{A}_i x - b_i\|^2$.*

Proof. Since $\mathcal{y}^*$ must satisfy $\nabla h_\beta(\mathcal{y}^*) = 0$, we have

$$\nabla_i h_\beta(\mathcal{y}^*) = \frac{\beta}{n}(y^\star_i - \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i y^\star_i) - \frac{\beta}{n}(\mathbf{I} - \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i)(\mathbf{T}\mathcal{y}^\star + c)$$

$$= \frac{\beta}{n}(\mathbf{I} - \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i)(y^\star_i - \mathbf{T}\mathcal{y}^\star - c) = 0. \tag{39}$$

Since $\mathbf{I} - \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i \succ 0$ with $\beta > \sigma^*_{\max}$, we conclude

$$y^\star_i - \mathbf{T}\mathcal{y}^\star - c = 0, \ \forall i = 1, \ldots, n, \tag{40}$$

which implies $y^\star$ given in the Lemma. It is easy to verify the rest of the Lemma using optimality conditions. □

Define **one epoch** as $\mathcal{J}(\delta)$ iterations, and let

$$h^\star_\beta := \min_{\mathcal{y}}\{h_\beta(\mathcal{y})\}, \quad F_t := \mathbb{E} h_\beta(\mathcal{y}^{t\mathcal{J}(\delta)}) - h^\star_\beta. \tag{41}$$

The next lemma is fundamental to the remaining analysis.

Lemma 5. *Under Assumption 1 and $\beta > \sigma^*_{\max}$, for any $\delta > 0$, we have*

$$F_t^2 \leq \frac{2\beta^2\mathcal{J}(\delta)}{n(1-\delta)\pi_*}\left(F_t - F_{t+1}\right) \cdot \mathbb{E}\|\mathcal{y}^{t\mathcal{J}(\delta)} - \mathcal{y}^\star\|^2, \tag{42}$$

where $\mathcal{J}(\delta)$ is defined in (6).

Proof. We first upper bound $\|\nabla h_\beta(\mathcal{y}^k)\|^2$. Verify

$$\nabla_i h_\beta(\mathcal{y}^k) = \frac{\beta}{n}\mathbf{D}_i^2\left(y^k_i - \mathbf{T}\mathcal{y}^k - c\right). \tag{43}$$

Investigate step (11b) for $i = i_k$ as

$$\begin{aligned} y^{k+1}_i &= \arg\min_y \frac{1}{2}\|\mathbf{A}_i y - b_i\|^2 + \frac{\beta}{2}\|y - x^{k+1} - \frac{1}{\beta}z^k_i\|^2 \\ &= (\mathbf{A}_i^\mathsf{T}\mathbf{A}_i + \beta\mathbf{I})^{-1}\left(\mathbf{A}_i^\mathsf{T} b_i + \beta x^{k+1} + z^k_i\right) \\ &\overset{(a)}{=} (\mathbf{A}_i^\mathsf{T}\mathbf{A}_i + \beta\mathbf{I})^{-1}\left(\beta\mathbf{T}\mathcal{y}^k + \beta c + \mathbf{A}_i^\mathsf{T}\mathbf{A}_i y^k_i\right) \\ &= y^k_i + (\mathbf{I} + \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i)^{-1}\left(\mathbf{T}\mathcal{y}^k + c - y^k_i\right) \\ &\overset{(43)}{=} y^k_i - \frac{n}{\beta}(\mathbf{I} + \frac{1}{\beta}\mathbf{A}_i^\mathsf{T}\mathbf{A}_i)^{-1}\mathbf{D}_i^{-2}\left(\nabla_i h_\beta(\mathcal{y}^k)\right), \end{aligned} \tag{44}$$

where (a) follows from (27) and $\mathbf{T}$'s definition. Thence,

$$\|\nabla_{i_k} h_\beta(\mathcal{y}^k)\| = \frac{\beta}{n}\Big\|(\mathbf{I} - \frac{1}{\beta^2}(\mathbf{A}_{i_k}^\mathsf{T}\mathbf{A}_{i_k})^2)(y^{k+1}_{i_k} - y^k_{i_k})\Big\|$$

$$\leq \frac{1}{n}(\beta - \frac{{\sigma^*_{\max}}^2}{\beta})\|\mathcal{y}^{k+1} - \mathcal{y}^k\|, \tag{45}$$

For any $k \geq \mathcal{J}(\delta) - 1$, we further have

$$\begin{aligned} &\|\nabla_{i_k} h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2 \\ =&\|\nabla_{i_k} h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1}) - \nabla_{i_k} h_\beta(\mathcal{y}^k) + \nabla_{i_k} h_\beta(\mathcal{y}^k)\|^2 \\ \leq& 2\|\nabla_{i_k} h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1}) - \nabla_{i_k} h_\beta(\mathcal{y}^k)\|^2 + 2\|\nabla_{i_k} h_\beta(\mathcal{y}^k)\|^2 \\ \overset{(45)}{\leq}& \frac{2\beta^2\mathcal{J}'}{n^2}\sum_{d=k-\mathcal{J}(\delta)+1}^{k-1}\|\mathcal{y}^{d+1} - \mathcal{y}^d\|^2 + \frac{2}{n^2}(\beta - \frac{{\sigma^*_{\max}}^2}{\beta})^2\|\mathcal{y}^{k+1} - \mathcal{y}^k\|^2 \\ \leq& \max\Big\{\frac{2\beta^2\mathcal{J}'}{n^2}, \frac{2}{n^2}(\beta - \frac{{\sigma^*_{\max}}^2}{\beta})^2\Big\}\sum_{d=k-\mathcal{J}(\delta)+1}^{k}\|\mathcal{y}^{d+1} - \mathcal{y}^d\|^2 \\ \leq& \frac{2\beta^2\mathcal{J}(\delta)}{n^2}\sum_{d=k-\mathcal{J}(\delta)+1}^{k}\|\mathcal{y}^{d+1} - \mathcal{y}^d\|^2, \end{aligned} \tag{46}$$

where $\mathcal{J}' = \mathcal{J}(\delta) - 1$ and the last inequality holds because $\beta > \sigma^*_{\max}$. With the filtration $\mathcal{X}^k = \sigma\{\mathcal{y}^0, \cdots, \mathcal{y}^k, i_0, \cdots, i_{k-1}\}$,

$$\begin{aligned} &\mathbb{E}\left(\|\nabla_{i_k} h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2 | \mathcal{X}^{k-\mathcal{J}(\delta)+1}\right) \\ &= \mathbb{E}\left(\|\nabla_{i_k} h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2 | \mathcal{y}^{k-\mathcal{J}(\delta)+1}, i_{k-\mathcal{J}(\delta)}\right) \\ &= \sum_{j=1}^N [\mathbf{P}^{\mathcal{J}(\delta)}]_{i_{k-\mathcal{J}(\delta)}, j}\|\nabla_j h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2 \\ &\overset{(5)}{\geq} (1-\delta)\pi_*\|\nabla h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2. \end{aligned} \tag{47}$$

Reverting the sides of (47) and taking expectation over $\mathcal{X}^{k-\mathcal{J}(\delta)+1}$, followed by applying (46), we have for $k \geq \mathcal{J}(\delta) - 1$

$$\begin{aligned} &\mathbb{E}\|\nabla h_\beta(\mathcal{y}^{k-\mathcal{J}(\delta)+1})\|^2 \\ &\leq \frac{2\beta^2\mathcal{J}(\delta)}{n^2(1-\delta)\pi_*}\sum_{d=k-\mathcal{J}(\delta)+1}^{k}\mathbb{E}\left(\|\mathcal{y}^{d+1} - \mathcal{y}^d\|^2\right). \end{aligned} \tag{48}$$

Notice that

$$h_\beta(\mathcal{y}^k) - h_\beta(\mathcal{y}^{k+1})$$

$$\overset{(25)}{=} L_\beta(x^{k+1}, y^k; z^k) - L_\beta(x^{k+2}, y^{k+1}; z^{k+1})$$
$$= L_\beta(x^{k+1}, y^k; z^k) - L_\beta^{k+1} + L_\beta^{k+1} - L_\beta(x^{k+2}, y^{k+1}; z^{k+1})$$
$$\geq \frac{1}{n}\|y^k - y^{k+1}\|^2, \tag{49}$$

where the last line follows from parts 1 and 2 of Lemma 1. Combining (49) and (48), we get

$$\mathbb{E}\|\nabla h_\beta(y^{k-\mathcal{J}(\delta)+1})\|^2 \leq \frac{2\beta^2 \mathcal{J}(\delta)}{n(1-\delta)\pi_*} \mathbb{E}\left(h_\beta(y^{k-J+1}) - h_\beta(y^{k+1})\right). \tag{50}$$

Now with $k = (t+1)\mathcal{J}(\delta) - 1$, (50) reduces to

$$\mathbb{E}\|\nabla h_\beta(y^{t\mathcal{J}(\delta)})\|^2 \leq \frac{2\beta^2 \mathcal{J}(\delta)}{n(1-\delta)\pi_*} \mathbb{E}\left(h_\beta(y^{t\mathcal{J}(\delta)}) - h_\beta(y^{(t+1)\mathcal{J}(\delta)})\right) \overset{(41)}{=} \frac{2\beta^2 \mathcal{J}(\delta)}{n(1-\delta)\pi_*} (F_t - F_{t+1}) \tag{51}$$

By the convexity of $h_\beta(\cdot)$,

$$\mathbb{E}h_\beta(y^{t\mathcal{J}(\delta)}) - h_\beta^\star \leq \mathbb{E}\langle \nabla h_\beta(y^{t\mathcal{J}(\delta)}), y^{t\mathcal{J}(\delta)} - y^\star\rangle. \tag{52}$$

Since both sides of (52) are nonnegative, we square them and use the Cauchy-Schwarz inequality to get

$$F_t^2 \leq \mathbb{E}\|\nabla h_\beta(y^{t\mathcal{J}(\delta)})\|^2 \cdot \mathbb{E}\|y^{t\mathcal{J}(\delta)} - y^\star\|^2. \tag{53}$$

Substituting (50) into (53) completes the proof. □

Now we are ready to establish the linear convergence rate of the sequence $(F_t)_{t\geq 0}$.

Theorem 2. *Under Assumption 1, for $\beta > 2\sigma^*_{\max} + 2$, we have linear convergence (with ν given in Lemma 3):*

$$F_{t+1} \leq \left(1 + \frac{n(1-\delta)\pi_*\nu}{4\beta^2\mathcal{J}(\delta)}\right)^{-1} F_t, \quad \forall\, t \geq 0. \tag{54}$$

Proof. By the strong convexity of $h_\beta(\cdot)$ and $y^\star = \arg\min h_\beta(y)$, it holds for any $y \in \mathbb{R}^{np}$ that,

$$\frac{\nu}{2}\|y - y^\star\|^2 \leq h_\beta(y) - h_\beta(y^\star). \tag{55}$$

Hence,

$$\mathbb{E}\|y^{t\mathcal{J}(\delta)} - y^\star\|^2 \leq \frac{2(\mathbb{E}h_\beta(y^{t\mathcal{J}(\delta)}) - h_\beta^\star)}{\nu} = \frac{2F_t}{\nu}. \tag{56}$$

Substituting (56) into (42), we have

$$F_t^2 \leq \frac{C}{\nu}(F_t - F_{t+1})F_t, \quad \text{where} \quad C = \frac{4\beta^2\mathcal{J}(\delta)}{n(1-\delta)\pi_*}. \tag{57}$$

By (49), the sequence $\{h_\beta(y^k)\}$ is non-increasing, implying $0 \leq F_{t+1} \leq F_t$. This together with (57) yields

$$F_t F_{t+1} \leq \frac{C}{\nu}(F_t - F_{t+1})F_t, \tag{58}$$

which is equivalent to (54). □

Theorem 2 states that W-ADMM for decentralized least squares converges linearly by epoch (every $\mathcal{J}(\delta)$ iterations).

V. Communication Analysis

This section derives and compares communication complexities. First, we establish the communication complexity of W-ADMM. From (55) and (41), we have

$$\mathbb{E}\|y^{t\mathcal{J}(\delta)} - y^\star\|^2 \leq \frac{2}{\nu}F_t \overset{(54)}{\leq} \left(\frac{2}{\nu}\right)\left(1 + \frac{n(1-\delta)\pi_*\nu}{4\beta^2\mathcal{J}(\delta)}\right)^{-t} F_0. \tag{59}$$

To achieve mean-square deviation $G_t := \mathbb{E}\|y^{t\mathcal{J}(\delta)} - y^\star\|^2 \leq \epsilon$, it is enough to have

$$\left(\frac{2F_0}{\nu}\right)\left(1 + \frac{n(1-\delta)\pi_*\nu}{4\beta^2\mathcal{J}(\delta)}\right)^{-t} \leq \epsilon, \tag{60}$$

which is implied by

$$t = \ln\left(\frac{2F_0}{\nu\epsilon}\right) \Big/ \ln\left(1 + \frac{n(1-\delta)\pi_*\nu}{4\beta^2\mathcal{J}(\delta)}\right). \tag{61}$$

Since β can be regarded as constants that are independent of network size n, and ν is $O(\frac{1}{n})$, we can write:

$$t = O\Big(\ln\big(\frac{n}{\epsilon}\big) / \ln\big(1 + \frac{(1-\delta)\pi_*}{\mathcal{J}(\delta)}\big)\Big) \tag{62}$$

For each epoch t, there are $\mathcal{J}(\delta)$ iterations, which use $O(\mathcal{J}(\delta))$ communication. Hence, to guarantee $G_t \leq \epsilon$, the total communication complexity is

$$O\Big(\Big(\ln\big(\frac{n}{\epsilon}\big) / \ln\big(1 + \frac{(1-\delta)\pi_*}{\mathcal{J}(\delta)}\big)\Big) \cdot \mathcal{J}(\delta)\Big) \tag{63}$$

Recall the definition of $\mathcal{J}(\delta)$ in (6), by setting δ as $1/2$, the communication complexity is

$$O\Big(\underbrace{\Big(\ln\big(\frac{n}{\epsilon}\big) / \ln\big(1 + \frac{(1-\sigma(\mathbf{P}))\pi_*}{2\ln\frac{2}{\pi_*}}\big)\Big)}_{\text{epoch number}} \cdot \underbrace{\frac{1}{1-\sigma(\mathbf{P})}\ln\frac{2}{\pi_*}}_{\text{comm. per epoch}}\Big), \tag{64}$$

where we remember $\sigma(\mathbf{P}) := \sup_{f\in\mathbb{R}^n: f^\mathsf{T}\mathbf{1}=0} \|f^\mathsf{T}\mathbf{P}\|/\|f\|$.

For simplicity of expression and comparison, in the succeeding parts we assume that the Markov chain is reversible with $\mathbf{P} = \mathbf{P}^\mathsf{T}$ and it admits a uniform stationary distribution $\pi^T = \pi^T\mathbf{P}$ with:

$$\pi = [1/n, \cdots, 1/n]^\mathsf{T} \in \mathbb{R}^n,$$

which implies $\pi_* = \min_i \pi_i = 1/n$. With $\mathbf{P}$ being a symmetric real matrix, we also have $\sigma(\mathbf{P}) = \lambda_2(\mathbf{P}) = \max\{|\lambda_i(\mathbf{P})| : \lambda_i(\mathbf{P}) \neq 1\}$. We get the total communication complexity of W-ADMM as

$$O\Big(\underbrace{\Big(\ln\big(\frac{n}{\epsilon}\big) \Big/ \ln\big(1 + \frac{1-\lambda_2(\mathbf{P})}{2n\ln(2n)}\big)\Big)}_{\text{epoch numbers}} \cdot \underbrace{\Big(\frac{\ln(n)}{1-\lambda_2(\mathbf{P})}\Big)}_{\text{comm. per epoch}}\Big) \tag{65}$$

A. Communication comparisons

For comparison, we list the communication complexities of some existing algorithms. D-ADMM [15] has:

$$O\Big(\underbrace{\Big(\ln\big(\frac{1}{\epsilon}\big) \Big/ \ln\big(1 + \sqrt{1-\lambda_2(\mathbf{P})}\big)\Big)}_{\text{iteration numbers}} \cdot \underbrace{m}_{\text{comm. per iter.}}\Big) \tag{66}$$

where m is the number of edges. The communication complexity of EXTRA [16] is

$$O\bigg(\Big(\ln\big(\frac{1}{\epsilon}\big)\Big/\ln\big(2-\lambda_2(\mathbf{P})\big)\Big)\cdot m\bigg) \tag{67}$$

As to exact diffusion [19], the communication complexity is

$$O\bigg(\Big(\ln\big(\frac{1}{\epsilon}\big)\Big/\ln\big(1+\frac{1-\lambda_2(\mathbf{P})}{\lambda_2(\mathbf{P})+C}\big)\Big)\cdot m\bigg) \tag{68}$$

where C only depends on L/ν, independent of $\lambda_2(\mathbf{P})$ and n.

Considering the case $\epsilon \le 1/\mathrm{e}$, it holds $\ln(n/\epsilon) \le \ln n \cdot \ln(1/\epsilon)$. Since $\ln(1+x) \approx x$ for x close to 0, W-ADMM in (65) can be simplified to:

$$O\bigg(\ln\big(\frac{1}{\epsilon}\big)\cdot\frac{n\ln^3(n)}{(1-\lambda_2(\mathbf{P}))^2}\bigg). \tag{69}$$

We similarly simplify the communication complexities in (66), (67), and (68). They are listed in Table I. Clearly, D-ADMM has a better communication complexity than EXTRA and exact diffusion.

Algorithm	comm. complexity
W-ADMM	$O\bigg(\ln\big(\frac{1}{\epsilon}\big)\cdot\frac{n\ln^3(n)}{(1-\lambda_2(\mathbf{P}))^2}\bigg)$
D-ADMM [15]	$O\Big(\ln\big(\frac{1}{\epsilon}\big)\cdot\big(\frac{m}{(1-\lambda_2(\mathbf{P}))^{1/2}}\big)\Big)$
EXTRA [16]	$O\Big(\ln\big(\frac{1}{\epsilon}\big)\cdot\big(\frac{m}{1-\lambda_2(\mathbf{P})}\big)\Big)$
Exact diffusion [19]	$O\Big(\ln\big(\frac{1}{\epsilon}\big)\cdot\big(\frac{m}{1-\lambda_2(\mathbf{P})}\big)\Big)$

TABLE I: Communication complexities of various algorithm when $\lambda_2(\mathbf{P})$ is close to 1

By comparison, W-ADMM is more communication efficient than D-ADMM when

$$\frac{n\ln^3(n)}{(1-\lambda_2(\mathbf{P}))^2} \le \frac{m}{(1-\lambda_2(\mathbf{P}))^{1/2}}, \tag{70}$$

equivalent to

$$\lambda_2(\mathbf{P}) \le 1 - \frac{n^{2/3}[\ln(n)]^2}{m^{2/3}} \approx 1 - \big(\frac{n}{m}\big)^{2/3}, \tag{71}$$

where the approximation holds for $\ln(n) \ll n$ and with $\ln(n)$ ignored. Condition (71) means the network has moderately good connectivity. When this holds, W-ADMM exhibits superior communication efficiency than the algorithms in comparison.

Let us consider three classes of graphs for concrete communication complexities.

Example 1 (Complete graph) In a complete graph, every agent connects with all the other nodes. The number of edges $m = O(n^2)$ and $\lambda_2(\mathbf{P}) = 0$. Consequently, the communication complexity of W-ADMM is $O(\ln(1/\epsilon)n\ln^3(n))$ while those of the other algorithms are $O(\ln(1/\epsilon)n^2)$. Noticing $\ln^3(n) \ll n$,W-ADMM is more communication efficient.

Example 2 (Random graph) Consider the random graphs by Edgar Gilbert [33], $G(n,p)$, in which an n-node graph is generated with each edge populating independently with probability $p \in (0,1)$. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ denote the adjacency matrix of the generated graph, with $A_{i,j} = 1$ if nodes i and j are connected, and 0 otherwise. The transition probability matrix $\mathbf{P}$ has $P_{i,j} = \frac{A_{i,j}}{\sqrt{d_i d_j}}$, where $d_i = \sum_{j=1}^n A_{i,j}$ is the degree of node i. It can be shown $\mathbb{E}[m] = \frac{p(n^2-n)}{2} = O(n^2)$. By Theorem 2 of [34], $1-\lambda_2(\mathbf{P})$ concentrates around $\lambda_{n-1}(\mathbf{I} - \bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-\frac{1}{2}})$, where $\bar{\mathbf{D}} = (n-1)p\mathbf{I}$, and

$$\bar{A}_{i,j} = \begin{cases} p & i \ne j \\ 0 & i = j. \end{cases} \tag{72}$$

Since $\bar{\mathbf{A}}$ is a Toeplitz matrix, one can verify that

$$\lambda_{n-1}(\mathbf{I} - \bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-\frac{1}{2}}) = \frac{n}{n-1}, \tag{73}$$

that is, $1-\lambda_2(\mathbf{P}) = O(1)$. With such setting, W-ADMM a communication complexity of roughly $O(\ln(1/\epsilon)n\ln^3 n)$ while the other algorithms have $O(\ln(1/\epsilon)n^2)$. Hence, W-ADMM is more communication-efficient.

Example 3 (Cycle graph) Consider a cycle, where each agent connects with its previous and next neighbors. One can verify that

$$1-\lambda_2(\mathbf{P}) = O(1-\cos(2\pi/n)) = O\left(1/n^2\right), \tag{74}$$

and $m = O(n)$. Hence, W-ADMM has a communication complexity of roughly $O(\ln(1/\epsilon)n^5\ln^3 n)$ while, in (66), D-ADMM has $O(n^2\ln(1/\epsilon))$, and in (68) –(68), EXTRA and exact diffusion have $O(n^3\ln(1/\epsilon))$, so W-ADMM is less communication-efficient.

VI. Numerical Experiments

In this section, we compare W-ADMM with existing state-of-the-art decentralized methods through numerical experiments. Consider a network of 50 nodes that are randomly placed in a 30×30 square. Any two nodes within a distance of 15 are connected; others are not. We set the probability transition matrix $\mathbf{P}$ as $[\mathbf{P}]_{ij} = 1/\max\{|\mathcal{N}_i|,|\mathcal{N}_j|\}$ and $[\mathbf{P}]_{ii} = 1-\sum_{j\ne i}[\mathbf{P}]_{ij}$.

The first experiment uses the least-squares problem (21) with $\mathbf{A}_i \in \mathbb{R}^{5\times10}$, $x \in \mathbb{R}^{10}$ and $b_i \in \mathbb{R}^5$. Each entry in $\mathbf{A}_i$ is generated from the standard Gaussian distribution, and $b_i := \mathbf{A}_i x_0 + v_i$, where $x_0 \sim \mathcal{N}(0, I_{10})$ and $v_i \sim \mathcal{N}(0, 0.1\times I_5)$. Fig. 2 compares different algorithms. For the random-walk (RW) incremental algorithm, we have used both a fixed step-size of 0.001 and a sequence of decaying step-sizes $\min\{0.01, 80/k\}$. For other algorithms, we have hand-optimized their parameters.

In the left plot of Fig. 2, we count one communication for each transmission of a p-length vector ($p = 10$ is the dimension of x). It is observed that W-ADMM is much more communication efficient than the other algorithms. In the right plot of Fig. 2, we illustrate the running times of these methods.

While a running time should in general include the times of computing, communication, and other overheads, we only include communication time and allows simultaneous communication over multiple edges for non-incremental algorithms. However, we assume each communication follows an i.i.d. exponential distribution with parameter 1. Each iteration of D-ADMM, EXTRA, and exact diffusion waits for the completion of the slowest communication (out of $2m$ communications), which determines the communication time of that iteration. In contrast, random-walk incremental algorithms and W-ADMM only use one communication per iteration. Under our setting, W-ADMM takes longer to converge than D-ADMM, EXTRA, and exact diffusion.

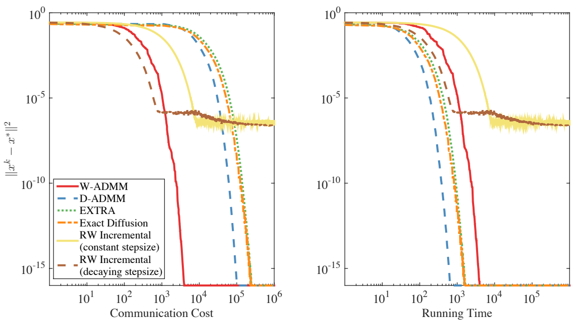

Fig. 2: Performance of decentralized algorithms on least squares.

The second experiment solves the logistic regression problem

$$\min_{x\in\mathbb{R}^p} \ \frac{1}{n}\sum_{i=1}^{n}\frac{1}{b}\Big[\sum_{j=1}^{b}\log\left(1+\exp(-y_{ij}v_{ij}^{\mathsf{T}}x)\right)\Big], \tag{75}$$

where $y_{ij} \in \{-1, 1\}$ denotes the label of the jth sample kept by the ith agent, and $v_{ij} \in \mathbb{R}^p$ represents its feature vector, and there are b samples kept by each agent. In this experiment, we set $b = 10, p = 5$. Each sample feature $v_{ij} \sim \mathcal{N}(0, 1)$. To generate y_{ij}, we first generate a random vector $x^0 \in \mathbb{R}^5 \sim \mathcal{N}(0, I)$. Then we generate a uniformly distributed variable $z_{ij} \sim \mathcal{U}(0, 1)$, and if $z_{ij} \le 1/[1 + \exp(-v_{ij}^{\mathsf{T}}x^0)]$, y_{ij} is taken as 1; otherwise y_{ij} is set as -1. We run the simulation over the same network as the above least-square problem. The communication efficiency of W-ADMM is also observed in Fig. 3.

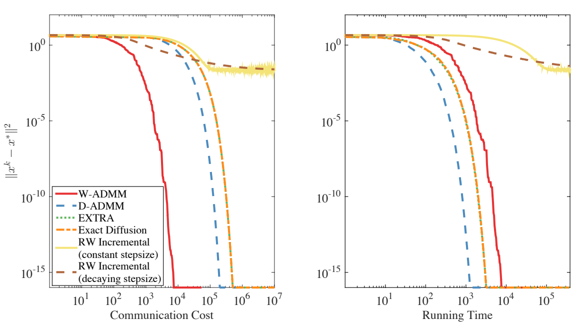

Fig. 3: Performance of decentralized algorithms on logistic regression.

VII. Conclusion

We have proposed a (random) walk algorithm, called W-ADMM, for convex decentralized optimization. The (random) walk carries the current solution x and lets it updated by every visited agent. The sequence of x converges to the optimal solution with a fixed step-size, which makes W-ADMM more efficient than the existing algorithms based on random walks. We have found W-ADMM requires less total communication than popular algorithms such as D-ADMM, EXTRA, and exact diffusion though longer wall-clock time to converge. Random walks also add another layer of privacy protection. Possible future work includes accelerating this algorithm and extending it to nonconvex optimization and varying networks.

VIII. Appendix

Proof of Lemma 1. We prove statement 1). Remember that agent i_k is activated at iteration k. From the optimality condition of (11b) for $i = i_k$, we get

$$\nabla f_i(y_{i_k}^{k+1}) - (z_{i_k}^k + \beta(x^{k+1} - y_{i_k}^{k+1})) = 0. \tag{76}$$

Substituting the above into (11c) yields

$$\nabla f_i(y_i^{k+1}) = z_i^{k+1}, \quad \text{for} \ \ i = i_k. \tag{77}$$

Hence, for $i = i_k$ it holds that

$$\begin{aligned}\|z_i^{k+1}-z_i^k\| &\overset{(a)}{=} \|z_i^{k+1}-z_i^{k(i)+1}\| \overset{(77)}{=} \|\nabla f_i(y_i^{k+1})-\nabla f_i(y_i^{k(i)+1})\| \\ &\overset{(14)}{\le} L\|y_i^{k+1} - y_i^{k(i)+1}\| \overset{(b)}{=} L\|y_i^{k+1} - y_i^k\|.\end{aligned} \tag{78}$$

where $k(i)$ denotes the last iteration at or before k when agent i is activated, that is, $k(i) := \max\{\ell : \ i_\ell = i, \ \ell < k\}$ (if agent i has not been visited before iteration k, $k(i)$ is set as 0). Equality (a) holds because $z_{i_k}^k = z_{i_k}^{k(i)+1}$ and equality (b) holds because $y_{i_k}^k = y_{i_k}^{k(i)+1}$. On the other hand, when $i \neq i_k$, i.e., agent i is not activated at k, it is obvious that $\|z_i^{k+1} - z_i^k\| = L\|y_i^{k+1} - y_i^k\| = 0$. As a result, we have

$$\|\mathcal{z}^{k+1} - \mathcal{z}^k\| \le L\|\mathcal{y}^{k+1} - \mathcal{y}^k\|. \tag{79}$$

Next, we rewrite the augmented Lagrangian in (9) as

$$L_\beta\left(x, \mathcal{y}; \mathcal{z}\right) = \frac{1}{n}\left(F(\mathcal{y}) + \frac{\beta}{2}\|\mathbb{1}\otimes x - \mathcal{y} + \frac{\mathcal{z}}{\beta}\|^2 - \frac{\|\mathcal{z}\|^2}{2\beta}\right). \tag{80}$$

Applying the cosine identity $\|b + c\|^2 - \|a + c\|^2 = \|b - a\|^2 + 2\langle a + c, b - a\rangle$, we have

$$\begin{aligned}&L_\beta^k - L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k) \\ &= \frac{\beta}{2n}\left\|\mathbb{1}\otimes x^k - \mathcal{y}^k + \frac{\mathcal{z}^k}{\beta}\right\|^2 - \frac{\beta}{2n}\left\|\mathbb{1}(x^{k+1})^{\mathsf{T}} - \mathcal{y}^k + \frac{\mathcal{z}^k}{\beta}\right\|^2 \\ &= \frac{\beta}{2n}\sum_{i=1}^{n}\left(\|x^k - x^{k+1}\|^2 + 2\big\langle x^{k+1} - y_i^k + \frac{z_i^k}{\beta}, x^k - x^{k+1}\big\rangle\right) \\ &\overset{(10a)}{=} \frac{\beta}{2}\|x^k - x^{k+1}\|^2,\end{aligned} \tag{81}$$

which completes the proof for statement 1). Next, we prove statement 2) of the Lemma. From the Lagrangian (9),

$$\begin{aligned}&L_\beta(x^{k+1}, \mathcal{y}^k; \mathcal{z}^k) - L_\beta^{k+1} \\ &= \frac{1}{n}\Big(f_{i_k}(y_{i_k}^k) + \langle z_{i_k}^k, x^{k+1} - y_{i_k}^k\rangle + \frac{\beta}{2}\|x^{k+1} - y_{i_k}^k\|^2 \\ &\quad - f_{i_k}(y_{i_k}^{k+1}) - \langle z_{i_k}^{k+1}, x^{k+1} - y_{i_k}^{k+1}\rangle - \frac{\beta}{2}\|x^{k+1} - y_{i_k}^{k+1}\|^2\Big) \\ &\overset{(a)}{=} \frac{1}{n}\Big(f_{i_k}(y_{i_k}^k) - f_{i_k}(y_{i_k}^{k+1}) + \frac{\beta}{2}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2 \\ &\qquad - \langle y_{i_k}^k - y_{i_k}^{k+1}, z_{i_k}^{k+1}\rangle - \frac{1}{\beta}\|z_{i_k}^{k+1} - z_{i_k}^k\|^2\Big)\end{aligned}$$

$$\stackrel{(b)}{=} \frac{1}{n}\Big(f_{i_k}(y_{i_k}^k) - f_{i_k}(y_{i_k}^{k+1}) + \frac{\beta}{2}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2 - \langle y_{i_k}^k - y_{i_k}^{k+1}, \nabla f_{i_k}(y_{i_k}^{k+1})\rangle - \frac{1}{\beta}\|z_{i_k}^{k+1} - z_{i_k}^k\|^2\Big)$$

$$\stackrel{(c)}{\geq} \frac{1}{n}\Big(-\frac{L}{2}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2 + \frac{\beta}{2}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2 - \frac{L^2}{\beta}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2\Big)$$

$$\stackrel{(d)}{\geq} \frac{1}{n}\|y_{i_k}^k - y_{i_k}^{k+1}\|^2 = \frac{1}{n}\|\mathcal{y}^k - \mathcal{y}^{k+1}\|^2. \quad (82)$$

where equality (a) holds due to $\|b+c\|^2 - \|a+c\|^2 = \|b-a\|^2 + 2\langle a+c, b-a\rangle$ and recursion (11c), equality (b) holds because of (77), inequality (c) holds because of (78) and as $f_i(\cdot)$ is L-Lipschitz differentiable, and inequality (d) follows from the assumption $\beta \geq 2L+2$. Statement 3) holds by combining statements 1) and 2). To prove statement 4), verify

$$L_\beta\left(x^k, \mathcal{y}^k; \mathcal{z}^k\right) = \frac{1}{n}\sum_{j=1}^n \left(f_j(y_j^{k(j)+1}) + \langle z_j^{k(j)+1}, x^k - y_j^{k(j)+1}\rangle\right) + \frac{\beta}{2n}\|\mathbf{1}\otimes x^k - \mathcal{y}^k\|^2$$

$$\stackrel{(77)}{=} \frac{1}{n}\sum_{j=1}^n \left(f_j(y_j^{k(j)+1}) + \langle \nabla f_j(y_j^{k(j)+1}), x^k - y_j^{k(j)+1}\rangle\right) + \frac{\beta}{2n}\|\mathbf{1}\otimes x^k - \mathcal{y}^k\|^2 \quad (83)$$

$$\stackrel{(a)}{\geq} \frac{1}{n}\sum_{j=1}^n f_j(x^k) + \frac{\beta - L}{2n}\|\mathbf{1}\otimes x^k - \mathcal{y}^k\|^2$$

$$\geq \min_x\left\{\frac{1}{n}\sum_{j=1}^n f_j(x^k)\right\} + \frac{\beta - L}{2n}\|\mathbf{1}\otimes x^k - \mathcal{y}^k\|^2 > -\infty,$$

where (a) holds as each f_j is Lipschitz differentiable. Therefore, $L_\beta\left(x^k, \mathcal{y}^k; \mathcal{z}^k\right)$ is lower bounded. This, with the monotonicity statement 3), concludes that L_β^k is convergent and completes the proof for statement 4). □

Proof of Theorem 1. Compute the subdifferentials of the augmented Lagrangian (80) with the updates in (11):

$$\nabla_x L_\beta^{k+1} = -\frac{\beta}{n}(y_{i_k}^{k+1} - y_{i_k}^k) + \frac{1}{n}(z_{i_k}^{k+1} - z_{i_k}^k), \quad (84)$$

$$\nabla_{y_j} L_\beta^{k+1} = \frac{1}{n}\left(\nabla f_j(y_j^{k+1}) - z_j^{k+1} + \beta(y_j^{k+1} - x^{k+1})\right), \quad (85)$$

$$\nabla_{z_j} L_\beta^{k+1} = \frac{1}{n}\left(x^{k+1} - y_j^{k+1}\right). \quad (86)$$

For notational brevity, we define g^k and q_v^k as in (17). For any $\delta \in (0,1)$ and $k \geq \mathcal{J}(\delta)+1$, we get via the triangle inequality:

$$\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 = \|q_{i_k}^{k-\mathcal{J}(\delta)-1} - q_{i_k}^k + q_{i_k}^k\|^2 \quad (87)$$

$$\leq 2\|q_{i_k}^{k-\mathcal{J}(\delta)-1} - q_{i_k}^k\|^2 + 2\|q_{i_k}^k\|^2 \quad (88)$$

$$\leq 2\underbrace{\|g^{k-\mathcal{J}(\delta)-1} - g^k\|^2}_{A} + 2\underbrace{\|q_{i_k}^k\|^2}_{B} \quad (89)$$

Below, we upper bound the two terms, A and B, separately. To bound term A as (93) below, we bound each of the three parts of g^k. The 1st part is

$$\|\nabla_x L_\beta^{k-\mathcal{J}(\delta)} - \nabla_x L_\beta^{k+1}\|^2 \leq 2\|\nabla_x L_\beta^{k-\mathcal{J}(\delta)}\|^2 + 2\|\nabla_x L_\beta^{k+1}\|^2$$

$$\leq \frac{2(\beta+L)^2}{n^2}\Big(\|\mathcal{y}^{k+1} - \mathcal{y}^k\|^2 + \|\mathcal{y}^{k-\mathcal{J}(\delta)} - \mathcal{y}^{k-\mathcal{J}(\delta)-1}\|^2\Big), \quad (90)$$

where the 2nd inequality follows from (84) and (78). Then by (85), we bound the 2nd part of g^k

$$\|\nabla_{y_j} L_\beta^{k-\mathcal{J}(\delta)-1} - \nabla_{y_j} L_\beta^{k+1}\|^2$$

$$\stackrel{(a)}{\leq} \frac{4L^2 + 4\beta^2}{n^2}\|y_j^{k-\mathcal{J}(\delta)-1} - y_j^{k+1}\|^2 + \frac{4}{n^2}\|z_j^{k-\mathcal{J}(\delta)-1} - z_j^{k+1}\|^2 + \frac{4\beta^2}{n^2}\|x^{k-\mathcal{J}(\delta)-1} - x^{k+1}\|^2$$

$$\leq \frac{(\mathcal{J}(\delta)+2)(4+4\beta^2+4L^2)}{n^2}\sum_{t=k-\mathcal{J}(\delta)-1}^{k}\Big(\|x^t - x^{t+1}\|^2 + \|y_j^t - y_j^{t+1}\|^2 + \|z_j^t - z_j^{t+1}\|^2\Big), \quad (91)$$

where (a) uses the inequality of arithmetic and geometric means and the Lipschitz differentiability of f_j in Assumption 3. From (86), the 3rd part of g^k can be bounded as

$$\|\nabla_{z_j} L_\beta^{k-\mathcal{J}(\delta)-1} - \nabla_{z_j} L_\beta^{k+1}\|^2$$

$$\leq \frac{2}{n^2}(\|x^{k-\mathcal{J}(\delta)-1} - x^{k+1}\|^2 + \|y_j^{k-\mathcal{J}(\delta)-1} - y_j^{k+1}\|^2)$$

$$\leq \frac{2(\mathcal{J}(\delta)+2)}{n^2}\sum_{t=k-\mathcal{J}(\delta)-1}^{k}\Big(\|x^t - x^{t+1}\|^2 + \|y_j^t - y_j^{t+1}\|^2\Big). \quad (92)$$

Substituting (90), (91) and (92) into term A and applying (79), we get a constant C_1, depending on J, β, L and n, such that

$$A \leq C_1\sum_{t=k-\mathcal{J}(\delta)-1}^{k}\left(\|x^t - x^{t+1}\|^2 + \|\mathcal{y}^t - \mathcal{y}^{t+1}\|^2\right). \quad (93)$$

As for the term B, using (85) and (86) and then (11c) and (77), we have

$$\nabla_{y_{i_k}} L_\beta^{k+1} = \frac{1}{n}\left(z_{i_k}^k - z_{i_k}^{k+1}\right), \quad (94)$$

$$\nabla_{z_{i_k}} L_\beta^{k+1} = \frac{1}{n\beta}(z_{i_k}^{k+1} - z_{i_k}^k), \quad (95)$$

Then we get the bound by applying (77):

$$B \leq C_2\|\mathcal{y}^{k+1} - \mathcal{y}^k\|^2, \quad (96)$$

where $C_2 := \frac{L^2+(\beta+L)^2}{n^2} + \frac{L^2}{n^2\beta^2}$. Then substituting (93) and (96) into (89) and taking expectations on both sides, we have

$$\mathbb{E}\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 \leq (C_1+C_2)\sum_{t=k-\mathcal{J}(\delta)-1}^{k}\mathbb{E}\|x^t - x^{t+1}\|^2 + \mathbb{E}\|\mathcal{y}^t - \mathcal{y}^{t+1}\|^2 \quad (97)$$

With statements 3) and 4) of Lemma 1, we get

$$\sum_{k=0}^{\infty}\left(\mathbb{E}\|x^k - x^{k+1}\|^2 + \mathbb{E}\|\mathcal{y}^k - \mathcal{y}^{k+1}\|_2^2\right) < +\infty. \quad (98)$$

Hence, we get the convergence

$$\lim_{k\to\infty}\mathbb{E}\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 = 0. \quad (99)$$

To this end, create the filtration of sigma-algebra

$$\chi^k := \sigma\left(x^0, \cdots, x^k, y^0, \cdots, y^k, z^0, \cdots, z^k, i_0, \cdots, i_{k-1}\right). \tag{100}$$

Compute the conditional expectation:

$$\begin{aligned}
&\mathbb{E}\left(\|q_{i_k}^{k-\mathcal{J}(\delta)-1}\|^2 \mid \chi^{k-\mathcal{J}(\delta)}\right) \\
&= \sum_{j=1}^{n} [\mathbf{P}^{\mathcal{J}(\delta)}]_{i_{k-\mathcal{J}(\delta)},j}\left(\|\nabla_x L_\beta^{k-\mathcal{J}(\delta)}\|^2 + \|\nabla_{y_j} L_\beta^{k-\mathcal{J}(\delta)}\|^2 \right. \\
&\quad \left. + \|\nabla_{z_j} L_\beta^{k-\mathcal{J}(\delta)}\|^2\right) \\
&\overset{(a)}{\geq} (1-\delta)\pi_* \|g^{k-\mathcal{J}(\delta)-1}\|^2,
\end{aligned} \tag{101}$$

where (a) follows from (5) and the definition of g^k in (17). Then, with (99), it holds

$$\lim_{k\to\infty} \mathbb{E}\|g^k\|^2 = \lim_{k\to\infty} \mathbb{E}\|g^{k-\mathcal{J}(\delta)-1}\|^2 = 0. \tag{102}$$

By the Schwarz inequality $\left(\mathbb{E}\|g^k\|\right)^2 \leq \mathbb{E}\|g^k\|^2$, we have

$$\lim_{k\to\infty} \mathbb{E}\|g^k\| = 0. \tag{103}$$

□

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, 2001, vol. 4, pp. 2033–2036.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[4] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[5] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[6] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[7] J. C. Duchi, A. 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, 2012.

[8] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.

[9] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[10] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, 2015.

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

[12] 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.

[13] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[14] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

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

[16] 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.

[17] 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.

[18] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *arXiv:1607.03218*, Jul. 2016.

[19] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact dffusion for distributed optimization and learning – Part I: Algorithm development," *Submitted for publication*, Also available on arXiv, Feb. 2017.

[20] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, 1997.

[21] S. S. Ram, A Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 691–717, 2009.

[22] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[23] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[24] C. G. Lopes and A. H. Sayed, "Randomized incremental protocols over adaptive networks," in Proc. *ICASSP*, Dallas, TX, 2010, pp. 3514–3517.

[25] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 9, no. R2, pp. 41–76, 1975.

[26] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[27] E. K. Ryu and W. Yin, "Proximal-proximal-gradient method," *arXiv:1708.06908*, 2017.

[28] D. A. Levin and Y. Peres, *Markov chains and mixing times*, vol. 107, Second Edition, American Mathematical Soc., 2017.

[29] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: an algorithmic framework for asynchronous parallel coordinate updates," *ArXiv e-prints arXiv:1506.02396*, June 2015.

[30] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[31] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *arXiv preprint arXiv:1511.06324*, 2015.

[32] T. Sun and W. Yin, "Markov block coordinate descent," *UCLA CAM Report 18-20*, 2018.

[33] E. N. Gilbert, "Random graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[34] F. Chung and M. Radcliffe, "On the spectra of general random graphs," *the electronic journal of combinatorics*, vol. 18, no. 1, pp. 215, 2011.