Analysis and Applications (2019)
© World Scientific Publishing Company
DOI: 10.1142/S0219530519410082

A randomized incremental primal-dual method for decentralized consensus optimization

Chenxi Chen* and Yunmei Chen†
Department of Mathematics, University of Florida
Gainesville, Florida 32611, USA
**chenc@ufl.edu*
†yun@ufl.edu

Xiaojing Ye
Department of Mathematics and Statistics
Georgia State University, Atlanta
Georgia 30303, USA
xye@gsu.edu

Received 30 July 2019
Accepted 29 October 2019
Published 27 November 2019

We consider a class of convex decentralized consensus optimization problems over connected multi-agent networks. Each agent in the network holds its local objective function privately, and can only communicate with its directly connected agents during the computation to find the minimizer of the sum of all objective functions. We propose a randomized incremental primal-dual method to solve this problem, where the dual variable over the network in each iteration is only updated at a randomly selected node, whereas the dual variables elsewhere remain the same as in the previous iteration. Thus, the communication only occurs in the neighborhood of the selected node in each iteration and hence can greatly reduce the chance of communication delay and failure in the standard fully synchronized consensus algorithms. We provide comprehensive convergence analysis including convergence rates of the primal residual and consensus error of the proposed algorithm, and conduct numerical experiments to show its performance using both uniform sampling and important sampling as node selection strategy.

Keywords: Decentralized consensus optimization; primal-dual method; incremental dual; importance sampling.

Mathematics Subject Classification 2010: 90C06, 90C25, 90C30

[†] Corresponding author.

1. Introduction

In this paper, we are interested in solving the following class of convex decentralized consensus optimization (DCO) problems:

$$\min_{\hat{x}\in\mathbb{R}^n} \hat{f}(\hat{x}) \quad \text{where } \hat{f}(\hat{x}) := \sum_{i=1}^{m} f_i(\hat{x}). \tag{1.1}$$

The problem (1.1) is defined on a simple, connected, undirected graph (network) $G = (V, E)$, where G consists of a node set $V = \{1, 2, \ldots, m\}$ and an edge set $E \subseteq V \times V$ such that $(i, j) \in E$ if and only if i and j are connected by an edge. Each node i can only communicate with its neighbors $j \in N_i := \{j \,|\, (i, j) \in E\}$ (we also denote $G_i := \{i\} \cup N_i$ for later use) during the computation. To establish convergence and the rates of our algorithm in this paper, we assume that each $f_i : \mathbb{R}^n \to \mathbb{R}$, which is held privately at node i, is convex and has Lipschitz continuous gradient.

To present our algorithmic development and convergence analysis in a more general and concise way, we adopt the notation $M = L \otimes I_n \in \mathbb{R}^{mn\times mn}$ which is a square matrix consisting of $m \times m$ blocks with the (i, j)th block as $l_{ij} I_n \in \mathbb{R}^{n\times n}$. Here, I_n is the $n \times n$ identity matrix, $L = [l_{ij}] := I - D^{-1/2} A D^{-1/2} \in \mathbb{R}^{m\times m}$ is the (normalized) graph Laplacian matrix, where A is the adjacency matrix of G, $D = \operatorname{diag}(d_1, \ldots, d_m)$ is the diagonal matrix with the degree $d_i = |N_i|$ of node i as the ith diagonal entry, and $\otimes$ is the Kronecker product. During the computation, each node i may obtain an estimate $x_i \in \mathbb{R}^n$ of the underlying solution $x^* \in \mathbb{R}^n$ of (1.1). We hence define $x := [x_1; \ldots; x_n] \in \mathbb{R}^{mn}$ by stacking the column vectors x_i's vertically. Then the problem (1.1) can be rewritten as an equivalent, constrained optimization problem[a]:

$$\min_{x\in\mathbb{R}^{mn}} f(x) := \sum_{i=1}^{m} f_i(x_i) \quad \text{subject to } Mx = 0, \tag{1.2}$$

due to the definition of L and the Perron–Frobenius theorem. More precisely, L is a symmetric matrix with zero row sums, and the eigenvalues of L are in $[0, 2)$ with 0 being the eigenvalue of multiplicity 1. Moreover, $Lu = 0$ if and only if $u = (c, \ldots, c)^\top \in \mathbb{R}^n$ for some constant $c \in \mathbb{R}$. Therefore, $Mx = 0$ if and only if $x_1 = x_2 = \cdots = x_m \in \mathbb{R}^n$, and hence the equivalency of (1.1) and (1.2). We further partition the rows of M into m sub-matrices such that $M = [M_1; \ldots; M_m]$ where the sub-matrix $M_i \in \mathbb{R}^{n\times mn}$ consists of the rows $(i-1)n+1$ to in of M. Note that L assumes equal weights for all edges and use the binary adjacency matrix A in its

[a] It is convenient to use $\hat{f} : \mathbb{R}^n \to \mathbb{R}$ in (1.1) and $f : \mathbb{R}^{mn} \to \mathbb{R}$ in (1.2) interchangeably as the meaning will be clear in the context. In particular, for consensual $x = [\hat{x}; \ldots; \hat{x}] \in \mathbb{R}^{mn}$ that consists of m identical copies of $\hat{x} \in \mathbb{R}^n$ (by stacking these column vectors $\hat{x}$ vertically following the standard Matlab syntax $[\cdot; \cdot]$), there is $f(x) = \hat{f}(\hat{x})$. Therefore, we only use f in the remainder of the paper.

definition. If the graph is weighted, we can use weighted Laplacian matrix L and all results presented in this paper follow similarly.

Following the theory of Lagrangian multipliers, we introduce a dual variable $z = [z_1; \dots; z_n] \in \mathbb{R}^{mn}$ for the constraint $Mx = 0$, and further rewrite the problem (1.2) as a saddle point problem

$$\min_{x \in \mathbb{R}^{mn}} \max_{z \in \mathbb{R}^{mn}} \left\{ \sum_{i=1}^{m} f_i(x_i) + \langle Mx, z \rangle \right\}. \tag{1.3}$$

Therefore, the problems (1.1)–(1.3) are all equivalent and hence can be referred interchangeably. Note that $\langle Mx, z \rangle = \sum_{i=1}^{m} \langle M_i x, z_i \rangle = \sum_{i=1}^{m} \langle x_i, M_i z \rangle$.

The DCO problem (1.1) has a wide range of applications such as distributed machine learning [6, 12, 20, 24], sensor networks [17, 35, 41], smart grids [13, 27], multiple-agent control and coordination [32, 33, 50]. In these applications, it is often uneconomical, difficult, and sometimes impossible to have a central (master) node that communicates with every individual node and processes all the data, due to various reasons including the extremely large sizes of local datasets and privacy issues. These limitations prohibit the transmission of $f_i(x)$ between nodes. Hence, a more promising approach of sensor network applications is to let the nodes share their own estimates of x^*, the optimal solution of the *global* optimization problem (1.1), only with their neighbor nodes during the computation. In addition, the estimate x_i obtained by node i should be consensual, namely $x_1 = \cdots = x_n$ or $Mx = 0$ in (1.2), upon convergence. This ensures that one can retrieve an accurate approximation to x^* from any node on the network, as required in real-world decentralized consensus network applications.

However, the applications of modern large-scale networks are severely hindered by the increased chance of communication delays or failures in the standard synchronous DCO setting, where all nodes are required to complete exchanging local estimates (or equivalent) with neighbors in each iteration. To overcome this issues, there have been a number of asynchronous algorithms, which are considered in a variety of application settings to allow computation and/or communication delays to some extent [4, 10, 31, 40, 43, 46, 47]. However, these methods are often based on very specific assumptions of the delays, which may not be practical in real-world applications.

Instead of building an asynchronous algorithm that allows delays, in this work we focus on how to greatly reduce the chance of delays by only requiring communications within local neighborhood during iterations. The main idea of our proposed method, called Randomized Incremental Primal-Dual (RIPD) method, is that at each iteration the dual variable over the network G is only updated at a randomly selected node (say i), then the partially updated dual variable is exerted to establish an estimator for updating of the primal variable over the entire network. Therefore, in this iteration, communications only occur in the local neighborhood G_i of i, which greatly reduces delays compared to global synchronizations in the standard DCO setting. Although such local communication strategy increases the total number of iterations, we will show that the total node-to-node communication number remains low, with the additional benefit of reducing delays per iteration for significantly improved efficiency overall.

The contributions of this paper mainly lie in two aspects as follows. First, we develop a RIPD method for convex DCO problem (1.1). Our proposed algorithm incorporates the idea of randomized incremental gradients (RIGs) into the primal-dual formulation, which only requires randomized incremental dual variable to update primal variables. More precisely, in each iteration, the algorithm only updates the dual variable at a randomly selected node, other nodes on the network can keep using outdated dual variables without communicating with their neighbors. Then the partially updated dual variable is exerted to establish an estimator for updating of the primal variable over the entire network. This significantly lowers the per-iteration communication and synchronization requirement, which can greatly reduce the chance of delays in the standard DCO setting. Second, we provide a comprehensive convergence analysis for the proposed RIPD method which fully characterizes the primal residual and consensus error. The convergence analysis mainly relies on the estimate of duality gap function. We are able to show that the RIPD can achieve the iteration complexity of $O(\bar{L}/\epsilon + m\bar{l}/\epsilon)$ with the total number of communication in the order of $O(d\bar{L}/\epsilon + md\bar{l}/\epsilon)$, where $d := \sum_{i=1}^{m} p_i d_i$, p_i is the probability that node i is selected at each iteration, $\bar{L} := \max(L_f, \sqrt{mL_f})$, $\bar{l} := \max_{1\leq i\leq m} l_i$ and $l_i := (\sum_{i=1}^{m} l_{ij}^2)^{1/2}$, and L_f is the Lipschitz constant of the gradient of the objective function f.

The remainder of this paper is organized as follows. Section 2 reviews the related work in the literature of DCO. In Sec. 3, we propose our RIPD algorithm, and present its main convergence properties under the uniform sampling and important sampling for the random incremental dual update setting. A comprehensive convergence analysis RIPD is carried out in Sec. 4. Section 5 presents the numerical results of the proposed algorithm. Finally, Sec. 6 concludes this paper.

2. Related Work

Early attempts to the DCO problem (1.1) focus on the globally synchronized setting. Under such setting, the decentralized gradient descent method [33] can solve the DCO problem (1.1) with diminishing step sizes. However, with constant step size policy, the iterates only converge to a point in the neighborhood of a solution. This issue is fixed by the decentralized exact first-order algorithm (EXTRA) developed in [38]. By introducing an error-correction term into the scheme of the decentralized gradient descent algorithm, it can converge consensually to the exact solution of (1.1) with a fixed large step size independent of the network size. EXTRA has a convergent rate $O(1/N)$ for general convex smooth f_i and a linear rate for restricted strongly convex f, where N is the number of iterations. Another approach to solve the DCO problem (1.1) is to use the alternating direction method of multipliers

(ADMM) to solve problem (1.2), where the equality constraints ensure the consensus property. ADMM is first applied to distributed optimization in [2] and further popularized in [1, 7, 11, 26, 28, 30, 36, 49]. ADMM (or linearized ADMM) exhibits an $O(1/N)$ convergence rate for a convex smooth objective function [1, 42], and a linear convergence rate for strongly convex smooth problems [39]. Furthermore, in [16] an ADMM based method achieves a $O(1/\sqrt{N})$ rate for solving non-convex global consensus problem.

Primal-Dual method has also been studied for DCO problems (e.g., [3, 8, 15, 18, 29, 34]). The works in [3, 8, 34] developed random coordinate decent primal-dual algorithms for distributed and asynchronous optimization. The numerical results showed high performance of these methods, but no convergence rates are given. The work in [29] presented a primal-dual based algorithm that uses local stochastic averaging gradients to achieve a linear rate for smooth and strongly convex problems. The work [15] develops a stochastic proximal gradient method for solving problem (1.1). It randomly activates edges with given probability in each iteration. The primal and dual variables which are related to activated edges will be updated. The consensus in [15] is achieved by communication in a global scale, due to the fact that all the edges in network are probably activated. A rate of convergence $O(1/N)$ is achieved for (1.1) with convex objective function.

During the past few years, RIG methods have emerged as an important class of first-order methods for finite-sum optimization problems [5, 9, 14, 19, 25, 37, 44, 48]. RIG methods are designed to reduce the number of full gradient evaluations but improve the convergence rate of stochastic gradient descent algorithm. For instance, the stochastic variance reduced gradient (SVRG) method presented in [19] iteratively updates the gradient of one randomly selected function in the summation and re-evaluating the exact gradient from time to time, to reduce the number of full gradient evaluation. SVRG is extended to solve proximal finite-sum problems in [44]. Later, the work in [48] shows that SVRG can be accelerated to achieve an optimal rate for minimizing the sum of strongly convex smooth functions. The randomized primal-dual gradient (RPDG) developed in [22] also achieves a similar rate for minimizing the sum of strongly convex smooth functions. In [45], a synchronous and an asynchronous primal-dual methods of multipliers (PDMM) for distributed optimization over a graph is proposed. In the asynchronous setting of PDMM, a node becomes active, updates its primal and dual variables, and sends them to its neighbors in each iteration. Convergence is established for a predefined order of node activations.

Recently, there have been several works on stochastic distributed primal-dual type methods for solving distributed optimization problems directly to improve communication efficiency. In [21], a decentralized communication sliding method is proposed for solving nonsmooth decentralized convex problems, in where the subproblem of the primal variable is approximately solved by an iterative subgradient descent procedure, and the inter-node communications only occur during the updates of the dual variables. The work in [23] proposes a centralized primal-dual gradient method. The structure of this network consists of slave agents and a master server. The master server dedicates to update the primal information using incremental gradients, and slave agents update dual variables. The idea of incremental gradients is similar to RPDG in [22], but the initialization in [22] requires an evaluation of full gradients from all the locally stored functions, while [23] does not require evaluation of full gradient at all. Inspired by the advancements in distributed algorithms, especially the work in [22, 23], the main focus of this work is to propose a RIPD algorithm that requires less communications at each iteration, but maintains the same convergence rate as its deterministic counterpart.

3. Proposed Randomized Incremental Primal-Dual Method

In this section, we present the details of our proposed RIPD. In RIPD, each node i maintains x_i and $\bar{x}_i$ for its own primal variable, and z_j and $\tilde{z}_j$ for the dual variable of each of its neighbor $j \in N_i$, which will be updated according the rules of RIPD during iterations. These variables with superscript t indicate their current values at iteration t. In each iteration t, RIPD selects one node $i_t = i \in V$ randomly according to probability $(p_1, \ldots, p_m)$. Then all nodes in the neighborhood G_i of this node i perform a weighted sum of its local primal variable obtained at the previous two iterations (see (3.1)) to obtain $\bar{x}_j^t$, and the neighbors in N_i send their $\bar{x}_j^t$ to i, which updates its own dual variable z_i to z_i^{t+1}, and then use it to establish an estimator $\tilde{z}_i{}^{t+1}$ as shown in (3.2) and (3.3). Then the node i sends $\tilde{z}_i^{t+1}$ back to its neighbors. Each node $j \neq i$ (i.e. the neighbors of i and those outside of G_i) simply sets $z_j^{t+1}, \tilde{z}_j^{t+1}$ to their old values z_j^t as in (3.3). Therefore $\tilde{z}^{t+1}$, with only $\tilde{z}_i^{t+1}$ actually updated, serves as the estimator for the dual variable of the entire network. To compensate the bias of this partially updated dual variable, the node i performs an extrapolation in (3.3) to obtain $\tilde{z}_i^{t+1}$ according to the probability p_i. All nodes on the network then perform the gradient descent (3.4) using the weighted sum of dual variables $M_j\tilde{z}^{t+1}$ to obtain the new x_j^{t+1}. Here $M_j z = \sum_{l \in G_j} l_{jl} z_l = \sum_{l=1}^m l_{jl} z_l$ is the weighted sum (using weights l_{jl}) of z_l in the neighbor $l \in G_j$. Since only the neighbors of i receive an updated $\tilde{z}_i^{t+1}$, each node j outside of G_i performs the update of x_j^{t+1} effectively based on their old copy of z_l^t for $l \in N_j$ without any communications. Therefore, the communication cost is only $2d_i$ in this iteration[b] (or $\sum_{i=1}^m 2p_i d_i$ expectedly in any iteration), in contrast to $2|E|$ per iteration in the standard DCO. The proposed RIPD algorithm is summarized in Algorithm 1..

We present the main convergence results for RIPD in the following theorem.

The RIPD algorithm can achieve a rate of convergence of $O(1/N)$ in terms of both primal residual $f(x) - f(x^*)$ and consensus error $\|Mx\|$, where N is the iteration number. The proofs involve several lemmas and are postponed to the next section.

[b] We count the communication number by one for every estimate sent by a node i and received by another node j.

Algorithm 1. Randomized incremental primal-dual (RIPD) method

```
For node $i$, initialize $x_i^1 \in X_i$, $z_i^1 \in Z_i$, $x_i^0 = x_i^1$, $i = 1, \ldots, m$.
for $t = 1, \ldots, N$ do
    Randomly choose $i_t$ according to $P(i_t = i) = p_i$, $1 \le i \le m$.
    Update $x^t$, $z^t$ as follows:
        $\bar{x}_j^t = \theta_t(x_j^t - x_j^{t-1}) + x_j^t, \quad \text{if } j \in G_{i_t},$    (3.1)
        $z_i^{t+1} = \begin{cases} \arg\min_{z_i \in Z_i} \langle -M_i \bar{x}^t, z_i \rangle + \frac{\tau_t}{2}\|z_i - z_i^t\|^2 & \text{if } i = i_t, \\ z_i^t & \text{if } i \neq i_t, \end{cases}$    (3.2)
        $\tilde{z}_i^{t+1} = \begin{cases} p_i^{-1}(z_i^{t+1} - z_i^t) + z_i^t & \text{if } i = i_t \\ z_i^t & \text{if } i \neq i_t, \end{cases}$    (3.3)
        $x_j^{t+1} = \arg\min_{x_j \in X_j} \langle \nabla f_j(x_j^t) + M_j \tilde{z}^{t+1}, x_j \rangle + \frac{\eta_t}{2}\|x_j - x_j^t\|^2$    (3.4)
end for
Return $(\underline{x}^N, \underline{z}^N) = \frac{1}{N}\sum_{t=1}^N (x^{t+1}, z^{t+1})$.
```

Theorem 3.1. *Let* $(\underline{x}^N, \underline{z}^N)$ *be generated by Algorithm* 1., *and* (x^*, z^*) *be a saddle point of problem* (1.3). *Suppose that the parameters in Algorithm* 1. *satisfy the following conditions for all* $t \ge 1$:

$$\eta_t \ge \eta_{t-1} \ge L_f + \max_{1 \le i \le m} \left\{ \frac{4 l_i^2}{\tau p_i} \right\}, \quad \tau_t = \tau, \quad \theta_t = 1. \tag{3.5}$$

Then the primal residual is bounded by

$$\mathbb{E}[f(\underline{x}^N) - f(x^*)] \le \frac{\eta_1 \|x^* - x^1\|^2}{2N} + \frac{1}{N}\sum_{i=1}^m \frac{\tau \|z_i^1\|^2}{2p_i}, \tag{3.6}$$

and the consensus error is bounded by

$$\mathbb{E}[\|M\underline{x}^N\|] \le \frac{U\|x^* - x^1\|}{N} + \sum_{i=1}^m \frac{V_i \|z_i^* - z_i^1\|}{N}, \tag{3.7}$$

where the constants are

$$U = \frac{\tau}{\underline{p}}\sqrt{\frac{\eta_1}{2\underline{C}}} + \bar{l}\sqrt{\frac{\eta_1}{2C}}, \quad V_i = \frac{\tau}{\underline{p}}\sqrt{\frac{\tau}{2p_i\underline{C}}} + \bar{l}\sqrt{\frac{\tau}{2p_i C}}, \tag{3.8}$$

$$C = \frac{\eta_N - L_f}{4} - \sum_{i=1}^m \frac{p_i l_i^2}{2\tau_N}, \quad \underline{C} = \min_{1 \le i \le m} \left\{ \frac{\tau}{2p_i} - \frac{\|M_i\|^2}{\eta_N - L_f} \right\}, \tag{3.9}$$

and $\underline{p} = \min_{1 \le i \le m} p_i$ *and* $\bar{l} = \max_{1 \le i \le m} l_i$.

Next, we provide two possible parameter settings with uniform sampling or importance sampling for Algorithm 1.. Uniform sampling is a standard setting where all nodes are selected with equal probability $p_i = 1/m$. In contrast, importance sampling improves convergence by selecting nodes of higher degree with greater probability. Corollary 3.2 provides the details of such importance sampling strategy.

We first give a parameter setting with uniform sampling for Algorithm 1..

Corollary 3.1. *Suppose that the parameters are given as*

$$p_i = \frac{1}{m}, \quad \tau_t = \tau, \quad \eta_t = L_f + \frac{4m\bar{l}^2}{\tau}, \tag{3.10}$$

where $\tau > 0$ is a constant. Then, $\{\eta_t\}_{t=1}^N$, $\{\tau_t\}_{t=1}^N$ satisfy (3.5).

Now, we provide an example of parameter setting with importance sampling in Corollary 3.2, in which each node is picked with certain node-related importance.

Corollary 3.2. *Suppose that the parameters are given as*

$$p_i = \frac{l_i^\alpha}{\|l\|_\alpha^\alpha}, \quad \tau_t = \tau, \quad \eta_t = L_f + \frac{4\|l\|_\alpha^\alpha}{\tau}\bar{l}^{2-\alpha}, \tag{3.11}$$

where $\alpha \in [0,2], \tau > 0$ are constants, $\|l\|_\alpha^\alpha = \sum_{i=1}^m l_i^\alpha$. Then, $\{\eta_t\}_{t=1}^N, \{\tau_t\}_{t=1}^N$ satisfy (3.5).

Comparing Corollary 3.1 with 3.2, we can see that taking importance sampling can improve the performance of the RIPD algorithm. For example, if we set $\alpha = 1$ in (3.10), then the step size η_t^{-1} in RIPD is $1/(L_f + 4\tau^{-1}\|l\|_1\bar{l})$. While by taking uniform sampling as in Corollary 3.1, the step size is $1/(L_f + 4\tau^{-1}m\bar{l}^2)$. By the definition of $\bar{l}$, we have that $\|l\|_1 \leq m\bar{l}$, which implies that importance sampling may allow a greater step size η_t^{-1} than that in uniform sampling.

From Theorem 3.1 and Corollary 3.1, we can have the estimate for the iteration complexity of $O(\bar{L}/\epsilon + m\bar{l}/\epsilon)$ for RIPD to obtain an ϵ-solution x^ϵ to problem (1.2), i.e. $\mathbb{E}[f(x^\epsilon) - f(x^*)] < \epsilon$, and $\mathbb{E}[\|Mx^\epsilon\|] < \epsilon$. The total communication complexity of RIPD to obtain an ϵ-solution of problem (1.2) is $O(d\bar{L}/\epsilon + md\bar{l}/\epsilon)$.

4. Convergence Analysis

In this section, we conduct comprehensive convergence analysis of Algorithm 1.. First of all, we introduce the duality gap function and provided some of its important properties. Denote $W = X \times Z$. For any $w = (x, z), w' = (x', z') \in W$, we define

$$Q(w, w') = f(x) - f(x') + \langle Mx, z' \rangle - \langle Mx', z \rangle. \tag{4.1}$$

It can be easily seen that $w = (x, z)$ is a solution of problem (1.3) if and only if $Q(w, w') \leq 0$ for all $w' = (x', z') \in W$ due to the convex–concave structure of (1.3). This suggests us to define the duality gap function as follows.

Definition 4.1. For problems (1.3) with compact feasible set $W = X \times Z$, the duality gap function is defined as

$$d(w) = \sup_{w' \in W} Q(w, w'). \tag{4.2}$$

For problems (1.3) with closed but unbounded feasible set W, the duality gap function is defined as

$$d_Z(v, w) = \sup_{z' \in Z} \{Q((x, z), (x^*, z')) - \langle v, z' \rangle\}, \tag{4.3}$$

where $v \in X$ and x^* is optimal for problem (1.1).

In this paper, we consider the case with $Z = Z_1 \times \cdots \times Z_m = \mathbb{R}^{mn}$ where $Z_i \in \mathbb{R}^n$, which is necessary for the decentralized consensus problem (1.3) but more challenging than the compact W case. We denote $d(v, w) := d_Z(v, w)$ for notation simplicity.

Proposition 4.1 states the relation between the duality gap function, the primal residue and the consensus error for problem (1.3).

Proposition 4.1. *Let $Z = \mathbb{R}^{mn}$. Suppose the random variables $w = (x, z)$ and v satisfy $\mathbb{E}[d(v, w)] < \infty$, where $d(v, w)$ is defined in* (4.3), *then the following identities hold almost surely* (*a.s.*):

$$f(x) - f(x^*) = d(v, w), \tag{4.4}$$

$$Mx = v. \tag{4.5}$$

In addition, if $\mathbb{E}[d(v, w)] \le \epsilon$, and $\mathbb{E}[\|v\|] \le \delta$, we have

$$\mathbb{E}[f(x) - f(x^*)] \le \epsilon \quad \textit{and} \quad \mathbb{E}[\|Mx\|] \le \delta. \tag{4.6}$$

Proof. By the definition of the duality gap function for $Z = \mathbb{R}^{mn}$, there is

$$d(v, w) = f(x) - f(x^*) + \sup_{z' \in Z} \langle Mx - v, z' \rangle, \tag{4.7}$$

where we used the fact that $Mx^* = 0$ since x^* is optimal for (1.3). Thus, $d(v, w) < \infty$ if and only if $v = Mx$. Furthermore, $\mathbb{E}[d(v, w)] < \infty$ implies $\text{Prob}[d(v, w) < \infty] = 1$, then $\text{Prob}[v = Mx] = 1$, which implies $Mx = v$ a.s. as in (4.5). Hence, $d(v, w) = f(x) - f(x^*)$ a.s. as in (4.4). Then (4.6) follows immediately. □

We now present several lemmas that will be critically useful for the convergence analysis later.

Lemma 4.1. *Suppose f has L_f-Lipschitz continuous gradient ∇f, then the following estimate holds for any $x \in X$:*

$$f(x^{t+1}) - f(x) \le \langle \nabla f(x^t), x^{t+1} - x \rangle + \frac{L_f}{2} \|x^{t+1} - x^t\|^2. \tag{4.8}$$

Proof. By the Lipschitz continuity of ∇f, we have

$$f(x^{t+1}) \le f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t \rangle + \frac{L_f}{2}\|x^{t+1} - x^t\|^2. \tag{4.9}$$

Moreover, by the convexity of f, there is $f(x) \ge f(x^t) + \langle \nabla f(x^t), x - x^t \rangle$ for all $x \in X$. Combining these two inequalities yields (4.8). □

Lemma 4.2. *Suppose the nonnegative real sequence* $\{a_t\}$ *is non-increasing, then*

$$\sum_{t=1}^{N} a_t(\|x - x^t\|^2 - \|x - x^{t+1}\|^2) \le a_1\|x - x^1\|^2 - a_N\|x - x^{N+1}\|^2. \tag{4.10}$$

Proof. The result follows immediately by rearranging the sum in the left-side into

$$a_1\|x - x^1\|^2 - a_N\|x - x^{N+1}\|^2 + \sum_{t=2}^{N}(a_t - a_{t-1})\|x - x^t\|^2, \tag{4.11}$$

and using the fact that $a_t \le a_{t-1}$ for all t. □

The following lemma provides three identities to characterize the conditional first and second moments of $\tilde{z}^{t+1}$ and z^{t+1} given $\mathcal{F}_t$, the filtration of the randomized consensus process in Algorithm 1. up to the tth iteration. For notation simplicity, we denote $\mathbb{E}_{|t}[X] = \mathbb{E}[X|\mathcal{F}_t]$ for any random variable (vector) X.

Lemma 4.3. *Let* $Z_i = \mathbb{R}^n$ *and* $\hat{z}_i^{t+1} := \arg\min_{z_i \in Z_i} \langle -M_i \bar{x}^t, z_i \rangle + \frac{\tau_t}{2}\|z_i - z_i^t\|^2$ *for* $i = 1, \ldots, m$. *Then the following identities hold*:

$$\mathbb{E}_{|t}[\tilde{z}_i^{t+1}] = \hat{z}_i^{t+1}, \tag{4.12}$$

$$\mathbb{E}_{|t}[\|z_i^t - z_i^{t+1}\|^2] = p_i\|z_i^t - \hat{z}_i^{t+1}\|^2, \tag{4.13}$$

$$\mathbb{E}_{|t}[\|z_i - z_i^{t+1}\|^2] = p_i\|z_i - \hat{z}_i^{t+1}\|^2 + (1 - p_i)\|z_i - z_i^t\|^2, \quad \forall\, z_i \in Z_i. \tag{4.14}$$

Proof. Recall the definition of $\tilde{z}_i^{t+1}$ in Algorithm 1., we have

$$\tilde{z}_i^{t+1} = \begin{cases} p_i^{-1}(z_i^{t+1} - z_i^t) + z_i^t, & \text{if } i = i_t, \\ z_i^t, & \text{if } i \ne i_t. \end{cases} \tag{4.15}$$

Note that, at the tth iteration, $\text{Prob}[\tilde{z}_i^{t+1} = p_i^{-1}(z_i^{t+1} - z_i^t) + z_i^t = p_i^{-1}(\hat{z}_i^{t+1} - z_i^t) + z_i^t] = \text{Prob}[i_t = i] = p_i$, and $\text{Prob}[\tilde{z}_i^{t+1} = z_i^t] = \text{Prob}[i_t \ne i] = 1 - p_i$. Then it follows immediately that

$$\mathbb{E}[\tilde{z}_i^{t+1}] = p_i \cdot (p_i^{-1}(\hat{z}_i^{t+1} - z_i^t) + z_i^t) + (1 - p_i) \cdot z_i^t = \hat{z}_i^{t+1}.$$

The identities (4.13) and (4.14) follow similarly. □

Now, we provide an important estimate to bound the duality gap function (4.3) at (x^t, z^t) generated by Algorithm 1..

Lemma 4.4. *Let* (x^{t+1}, z^{t+1}) *be generated by Algorithm* 1., *then the following estimate holds for any* $(x, z) \in W$:

$$\begin{aligned}\mathbb{E}_{|t}[Q((x^{t+1}, z^{t+1}), (x, z))] \le \mathbb{E}_{|t}\Bigg[&\langle M(x^{t+1} - x^t), z - \tilde{z}^{t+1}\rangle \\ &- \theta_t\langle M(x^t - x^{t-1}), z - \tilde{z}^t\rangle + \langle Mx, \tilde{z}^{t+1} - z^{t+1}\rangle \\ &+ \frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2 - \frac{\eta_t - L_f}{2}\|x^{t+1} - x^t\|^2 \\ &+ \sum_{i=1}^m \frac{\tau_t p_i^{-1}}{2}\left(\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2 - \|z_i^t - z_i^{t+1}\|^2\right) \\ &+ \theta_t\langle M_{i_t}(x^t - x^{t-1}), p_{i_t}^{-1}(z_{i_t}^{t+1} - z_{i_t}^t)\rangle \\ &- \theta_t\langle M_{i_{t-1}}(x^t - x^{t-1}), (p_{i_{t-1}}^{-1} - 1)\cdot(z_{i_{t-1}}^t - z_{i_{t-1}}^{t-1})\rangle\Bigg]. \qquad (4.16)\end{aligned}$$

Proof. By the definition of Q and Lemma 4.1, we have

$$\begin{aligned}Q((x^{t+1}, z^{t+1}), (x, z)) &= f(x^{t+1}) - f(x) + \langle Mx^{t+1}, z\rangle - \langle Mx, z^{t+1}\rangle \\ &\le \langle \nabla f(x^t), x^{t+1} - x\rangle + \frac{L_f}{2}\|x^{t+1} - x^t\|^2 \\ &\quad + \langle Mx^{t+1}, z\rangle - \langle Mx, z^{t+1}\rangle. \qquad (4.17)\end{aligned}$$

On the other hand, the optimality condition of x_j^{t+1} in (3.4) Algorithm 1. implies

$$\begin{aligned}&\langle \nabla f_j(x_j^t), x_j^{t+1}\rangle + \langle M_j\tilde{z}^{t+1}, x_j^{t+1}\rangle + \frac{\eta_t}{2}\|x_j^{t+1} - x_j^t\|^2 \\ &\quad \le \langle \nabla f_j(x_j^t), x_j\rangle + \langle M_j\tilde{z}^{t+1}, x_j\rangle + \frac{\eta_t}{2}\|x_j - x_j^t\|^2 - \frac{\eta_t}{2}\|x_j - x_j^{t+1}\|^2, \qquad (4.18)\end{aligned}$$

for all $x_j \in X_j$. Therefore, summing (4.18) over $i = 1, \ldots, m$ yields

$$\begin{aligned}\langle \nabla f(x^t), x^{t+1} - x\rangle \le \frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2 - \frac{\eta_t}{2}\|x^{t+1} - x^t\|^2 \\ + \langle M\tilde{z}^{t+1}, x - x^{t+1}\rangle, \qquad (4.19)\end{aligned}$$

where $x = [x_1; \ldots; x_m] \in \mathbb{R}^{mn}$. Combining (4.17) and (4.19) yields

$$\begin{aligned}Q((x^{t+1}, z^{t+1}), (x, z)) &\le \langle M\tilde{z}^{t+1}, x - x^{t+1}\rangle + \langle Mx^{t+1}, z\rangle - \langle Mx, z^{t+1}\rangle \\ &\quad + \frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2 - \frac{\eta_t - L_f}{2}\|x^{t+1} - x^t\|^2. \qquad (4.20)\end{aligned}$$

Due to the optimality condition of $\hat{z}_i^{t+1}$ in (3.2), we have

$$\langle -M_i\bar{x}^t, \hat{z}_i^{t+1}\rangle + \frac{\tau_t}{2}\|\hat{z}_i^{t+1} - z_i^t\|^2 \le \langle -M_i\bar{x}^t, z_i\rangle + \frac{\tau_t}{2}\|z_i - z_i^t\|^2 - \frac{\tau_t}{2}\|z_i - \hat{z}_i^{t+1}\|^2,$$

for all $z_i \in Z_i$, which can be rearranged into

$$\langle M_i\bar{x}^t, z_i - \hat{z}_i^{t+1}\rangle \le \frac{\tau_t}{2}\|z_i - z_i^t\|^2 - \frac{\tau_t}{2}\|z_i - \hat{z}_i^{t+1}\|^2 - \frac{\tau_t}{2}\|\hat{z}_i^{t+1} - z_i^t\|^2.$$

Then, by (4.12), we have $\mathbb{E}_{|t}[\langle M_i\bar{x}^t, z_i - \hat{z}_i^{t+1}\rangle] = \mathbb{E}_{|t}[\langle M_i\bar{x}^t, z_i - \tilde{z}_i^{t+1}\rangle]$. Furthermore, by (4.13) and (4.14) in Lemma 4.3, we have

$$\begin{aligned}
&\mathbb{E}_{|t}[\langle M_i\bar{x}^t, z_i - \tilde{z}_i^{t+1}\rangle] \\
&\quad\le \mathbb{E}_{|t}\left[\frac{\tau_t}{2}\|z_i - z_i^t\|^2 - \frac{\tau_t}{2}\|z_i - \hat{z}_i^{t+1}\|^2 - \frac{\tau_t}{2}\|\hat{z}_i^{t+1} - z_i^t\|^2\right] \\
&\quad\le \mathbb{E}_{|t}\left[\frac{\tau_t p_i^{-1}}{2}\left(\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2 - \|z_i^t - z_i^{t+1}\|^2\right)\right],
\end{aligned} \tag{4.21}$$

summing over $i = 1, \ldots, m$ of which implies that

$$\mathbb{E}_{|t}\left[\langle M\bar{x}^t, \tilde{z}^{t+1} - z\rangle + \sum_{i=1}^{m}\frac{\tau_t}{2p_i}(\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2 - \|z_i^t - z_i^{t+1}\|^2)\right] \ge 0. \tag{4.22}$$

Combining this inequality and (4.20), we can get

$$\begin{aligned}
&\mathbb{E}_{|t}[Q((x^{t+1}, z^{t+1}), (x, z))] \\
&\quad\le \mathbb{E}_{|t}\Bigg[\langle M\tilde{z}^{t+1}, x - x^{t+1}\rangle + \langle Mx^{t+1}, z\rangle - \langle Mx, z^{t+1}\rangle + \langle M\bar{x}^t, \hat{z}^{t+1} - z\rangle \\
&\qquad + \frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2 - \frac{\eta_t - L_f}{2}\|x^{t+1} - x^t\|^2 \\
&\qquad + \sum_{i=1}^{m}\frac{\tau_t p_i^{-1}}{2}(\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2 - \|z_i^t - z_i^{t+1}\|^2)\Bigg] \\
&\quad= \mathbb{E}_{|t}\Bigg[\langle M(x^{t+1} - x^t), z - \tilde{z}^{t+1}\rangle - \theta_t\langle M(x^t - x^{t-1}), z - \tilde{z}^t\rangle \\
&\qquad + \theta_t\langle M(x^t - x^{t-1}), \tilde{z}^{t+1} - \tilde{z}^t\rangle + \langle Mx, \tilde{z}^{t+1} - z^{t+1}\rangle \\
&\qquad + \frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2 - \frac{\eta_t - L_f}{2}\|x^{t+1} - x^t\|^2 \\
&\qquad + \sum_{i=1}^{m}\frac{\tau_t p_i^{-1}}{2}(\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2 - \|z_i^t - z_i^{t+1}\|^2)\Bigg].
\end{aligned} \tag{4.23}$$

Now, we focus on the term $\langle M(x^t - x^{t-1}), \tilde{z}^{t+1} - \tilde{z}^t\rangle$ above. From the updating rule of z^{t+1} in (3.2) and $\tilde{z}^{t+1}$ in (3.3) of Algorithm 1., we know that

$$\tilde{z}_i^t - z_i^t = \begin{cases} (p_i^{-1} - 1)\cdot(z_i^t - z_i^{t-1}), & \text{if } i = i_{t-1}, \\ 0, & \text{if } i \neq i_{t-1}, \end{cases} \tag{4.24}$$

and that

$$\tilde{z}_i^{t+1} - z_i^t = \begin{cases} p_i^{-1}(z_i^{t+1} - z_i^t), & \text{if } i = i_t, \\ 0, & \text{if } i \neq i_t. \end{cases} \tag{4.25}$$

Combining (4.24) and (4.25) yields

$$\begin{aligned} \langle M(x^t - x^{t-1}), \tilde{z}^{t+1} - \tilde{z}^t\rangle &= \sum_{i=1}^m \langle M_i(x^t - x^{t-1}), \tilde{z}_i^{t+1} - z_i^t\rangle \\ &\quad + \sum_{i=1}^m \langle M_i(x^t - x^{t-1}), z_i^t - \tilde{z}_i^t\rangle \\ &= \langle M_{i_t}(x^t - x^{t-1}), p_{i_t}^{-1}(z_{i_t}^{t+1} - z_{i_t}^t)\rangle \\ &\quad - \langle M_{i_{t-1}}(x^t - x^{t-1}), (p_{i_{t-1}}^{-1} - 1)(z_{i_{t-1}}^t - z_{i_{t-1}}^{t-1})\rangle. \end{aligned} \tag{4.26}$$

Plugging this into (4.23) implies (4.16), which completes the proof. □

Now, we are ready to prove the main convergence result, i.e. Theorem 3.1.

Proof of Theorem 3.1. By the convexity of $Q(w, w')$ in w and the definition of $(\underline{x}^N, \underline{z}^N)$, we know

$$Q((\underline{x}^N, \underline{z}^N), (x, z)) \leq \frac{1}{N}\sum_{t=1}^N Q((x^{t+1}, z^{t+1}), (x, z))$$

for any $(x, z) \in W$. By Lemma 4.4, we have

$$\begin{aligned} \mathbb{E}[Q((\underline{x}^N, \underline{z}^N), (x, z))] \leq \frac{1}{N}\cdot\mathbb{E}\Bigg\{ &\sum_{t=1}^N [\langle M(x^{t+1} - x^t), z - \tilde{z}^{t+1}\rangle \\ &- \theta_t\langle M(x^t - x^{t-1}), z - \tilde{z}^t\rangle] \\ &+ \sum_{t=1}^N \left[\frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2\right] + \langle Mx, s^N\rangle \\ &+ \sum_{t=1}^N \sum_{i=1}^m \frac{\tau_t p_i^{-1}}{2}\left[\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2\right] - \Lambda_0 \Bigg\}, \end{aligned} \tag{4.27}$$

where $s^N := \frac{1}{N}\sum_{t=1}^N(\tilde{z}^{t+1} - z^{t+1})$, and Λ_0 above denotes

$$\begin{aligned}\Lambda_0 := &\sum_{t=1}^N \frac{\eta_t - L_f}{2}\|x^{t+1} - x^t\|^2 + \sum_{t=1}^N \frac{\tau_t p_{i_t}^{-1}}{2}\|z_{i_t}^t - z_{i_t}^{t+1}\|^2 \\ &- \sum_{t=1}^N \theta_t \langle M_{i_t}(x^t - x^{t-1}), p_{i_t}^{-1}(z_{i_t}^{t+1} - z_{i_t}^t)\rangle \\ &+ \sum_{t=1}^N \theta_t \langle M_{i_{t-1}}(x^t - x^{t-1}), (p_{i_{t-1}}^{-1} - 1)\cdot(z_{i_{t-1}}^t - z_{i_{t-1}}^{t-1})\rangle, \end{aligned} \tag{4.28}$$

and we used the fact that $\sum_{t=1}^N\sum_{i=1}^m \frac{\tau_t p_i^{-1}}{2}\|z_i^t - z_i^{t+1}\|^2 = \sum_{t=1}^N \frac{\tau_t p_{i_t}^{-1}}{2}\|z_{i_t}^t - z_{i_t}^{t+1}\|^2$. By setting $\theta_t = 1$ for all t as in (3.5), the first two terms on the right side of (4.27) becomes

$$\begin{aligned}&\sum_{t=1}^N[\langle M(x^{t+1} - x^t), z - \tilde{z}^{t+1}\rangle - \theta_t\langle M(x^t - x^{t-1}), z - \tilde{z}^t\rangle] \\ &\quad = \langle M(x^{N+1} - x^N), z - \tilde{z}^{N+1}\rangle - \theta_1\langle M(x^1 - x^0), z - \tilde{z}^1\rangle \\ &\quad = \langle M(x^{N+1} - x^N), z - \tilde{z}^{N+1}\rangle, \end{aligned} \tag{4.29}$$

where the last equality is due to the initialization $x_1 = x_0$. For the second summation term in (4.27), we know that the sequence $\{\eta_t\}_{t=1}^N$ is non-increasing, and hence by Lemma 4.2 we have

$$\sum_{t=1}^N\left[\frac{\eta_t}{2}\|x - x^t\|^2 - \frac{\eta_t}{2}\|x - x^{t+1}\|^2\right] \le \frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2. \tag{4.30}$$

Moreover, by the setting of τ_t in (3.5), we have

$$\begin{aligned}&\sum_{t=1}^N\sum_{i=1}^m \frac{\tau_t p_i^{-1}}{2}[\|z_i - z_i^t\|^2 - \|z_i - z_i^{t+1}\|^2] \\ &\qquad \le \sum_{i=1}^m p_i^{-1}\left[\frac{\tau_1}{2}\|z_i - z_i^1\|^2 - \frac{\tau_N}{2}\|z_i - z_i^{N+1}\|^2\right]. \end{aligned} \tag{4.31}$$

Plugging (4.29)–(4.31) into (4.27), we obtain

$$\begin{aligned}\mathbb{E}[Q((\underline{x}^N, \underline{z}^N), (x, z))] \le \frac{1}{N}\cdot\mathbb{E}\Bigg\{&\frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2 \,, \\ &+ \sum_{i=1}^m p_i^{-1}\left[\frac{\tau_1}{2}\|z_i - z_i^1\|^2 - \frac{\tau_N}{2}\|z_i - z_i^{N+1}\|^2\right] - \Lambda\Bigg\}, \end{aligned}$$

where we used the updating rule of $\tilde{z}^{t+1}$ in Algorithm 1., and

$$\Lambda := \Lambda_0 + -\langle M(x^{N+1} - x^N),\ z - z^{N+1}\rangle.$$

Using the definition of Λ_0 in (4.28) and reorganizing Λ yield

$$\Lambda \geq \left[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 - \langle M(x^{N+1} - x^N), z - z^{N+1}\rangle\right] + \Lambda_1 + \Lambda_2 + \Lambda_3 + \sum_{t=2}^{N} \frac{\eta_{t-1} - L_f}{2}\|x^t - x^{t-1}\|^2, \tag{4.32}$$

where

$$\Lambda_1 = \frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 + \frac{\tau_N p_{i_N}^{-1}}{4}\|z_{i_N}^N - z_{i_N}^{N+1}\|^2 \geq 0 \tag{4.33}$$

$$\begin{aligned}\Lambda_2 &= \sum_{t=2}^{N}\left[\frac{\tau_t p_{i_t}^{-1}}{4}\|z_{i_t}^t - z_{i_t}^{t+1}\|^2 - \langle M_{i_t}(x^t - x^{t-1}), p_{i_t}^{-1}(z_{i_t}^{t+1} - z_{i_t}^t)\rangle\right]\\ &\geq -\sum_{t=2}^{N}\frac{\|M_{i_t}(x^t - x^{t-1})\|^2}{\tau_t p_{i_t}} \geq -\sum_{t=2}^{N}\frac{\|M_{i_t}\|^2}{\tau_t p_{i_t}}\|x^t - x^{t-1}\|^2,\end{aligned} \tag{4.34}$$

$$\begin{aligned}\Lambda_3 &= \sum_{t=2}^{N}\left[\frac{\tau_{t-1}}{4p_{i_{t-1}}}\|z_{i_{t-1}}^{t-1} - z_{i_{t-1}}^t\|^2 + \langle M_{i_{t-1}}(x^t - x^{t-1}),\right.\\ &\qquad \left.(p_{i_{t-1}}^{-1} - 1)\cdot(z_{i_{t-1}}^t - z_{i_{t-1}}^{t-1})\rangle\right]\\ &\geq -\sum_{t=2}^{N}\frac{(1 - p_{i_{t-1}})^2\|M_{i_{t-1}}(x^t - x^{t-1})\|^2}{\tau_{t-1}p_{i_{t-1}}}\\ &\geq -\sum_{t=2}^{N}\frac{(1 - p_{i_{t-1}})^2\|M_{i_{t-1}}\|^2}{\tau_{t-1}p_{i_{t-1}}}\|x^t - x^{t-1}\|^2.\end{aligned} \tag{4.35}$$

Substituting (4.33)–(4.35) in (4.32), we obtain

$$\begin{aligned}\Lambda &\geq \frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 - \langle M(x^{N+1} - x^N), z - z^{N+1}\rangle\\ &\quad + \sum_{t=2}^{N}\left(\frac{\eta_{t-1} - L_f}{2} - \frac{\|M_{i_t}\|^2}{\tau_t p_{i_t}} - \frac{(1 - p_{i_{t-1}})^2\|M_{i_{t-1}}\|^2}{\tau_{t-1}p_{i_{t-1}}}\right)\|x^t - x^{t-1}\|^2\\ &\geq \frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 - \langle M(x^{N+1} - x^N), z - z^{N+1}\rangle,\end{aligned}$$

where we obtained the second inequality by using (3.5), the definition $l_i = \|M_i\|$, and observing that

$$\frac{\eta_t - L_f}{4} \geq \max_{1\leq i\leq m}\left\{\frac{l_i^2}{\tau p_i}\right\} \geq \frac{l_i^2}{\tau p_i} > \max\left\{\frac{(1-p_i)^2 l_i^2}{\tau p_i}, \frac{l_i^2}{2\tau p_i}\right\}, \quad \forall\, t, i. \tag{4.36}$$

To sum up, we have an estimate of the duality gap function $Q((\underline{x}^N, \underline{z}^N), (x, z))$ as follows,

$$\begin{aligned}
\mathbb{E}[Q((\underline{x}^N, \underline{z}^N), (x, z))] &\leq \frac{1}{N}\cdot\mathbb{E}\Bigg\{\frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2 + \langle Mx, s^N\rangle \\
&\quad + \sum_{i=1}^m p_i^{-1}\cdot\left[\frac{\tau_1}{2}\|z_i - z_i^1\|^2 - \frac{\tau_N}{2}\|z_i - z_i^{N+1}\|^2\right] \\
&\quad - \left[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 - \langle M(x^{N+1} - x^N), z - z^{N+1}\rangle\right]\Bigg\} \\
&= \frac{1}{N}\cdot\mathbb{E}\Bigg\{\frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2 + \sum_{i=1}^m \frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2 + \langle Mx, s^N\rangle \\
&\quad - \Bigg[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 + \langle M(x^{N+1} - x^N), z^{N+1}\rangle \\
&\quad + \sum_{i=1}^m \frac{p_i^{-1}\tau_N}{2}\|z_i^{N+1}\|^2\Bigg] \sum_{i=1}^m \langle p_i^{-1}(\tau_1 z_i^1 - \tau_N z_i^{N+1}) + M_i(x^N - x^{N+1}), z_i\rangle\Bigg\}.
\end{aligned} \tag{4.37}$$

By reorganizing above inequality, we obtain

$$\begin{aligned}
&\mathbb{E}\left[Q((\underline{x}^N, \underline{z}^N), (x, z)) + \frac{1}{N}\sum_{i=1}^m \langle p_i^{-1}(\tau_1 z_i^1 - \tau_N z_i^{N+1}) + M_i(x^N - x^{N+1}), z_i\rangle\right] \\
&\quad \leq \frac{1}{N}\cdot\mathbb{E}\Bigg\{\frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2 + \sum_{i=1}^m \frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2 + \langle Mx, s^N\rangle \\
&\qquad - \Bigg[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 + \langle M(x^{N+1} - x^N), z^{N+1}\rangle \\
&\qquad + \sum_{i=1}^m \frac{p_i^{-1}\tau_N}{2}\|z_i^{N+1}\|^2\Bigg]\Bigg\} \\
&\quad = \frac{1}{N}\cdot\mathbb{E}\Bigg\{\frac{\eta_1}{2}\|x - x^1\|^2 - \frac{\eta_N}{2}\|x - x^{N+1}\|^2 + \sum_{i=1}^m \frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2 + \langle Mx, s^N\rangle \\
&\qquad - \left[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 - \sum_{i=1}^m \frac{p_i}{2\tau_N}\|M_i(x^N - x^{N+1})\|^2\right]\Bigg\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N}\cdot \mathbb{E}\left\{\frac{\eta_1}{2}\|x-x^1\|^2 - \frac{\eta_N}{2}\|x-x^{N+1}\|^2 + \sum_{i=1}^{m}\frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2 + \langle Mx, s^N\rangle\right.\\
&\quad \left. - \left(\frac{\eta_N - L_f}{4} - \sum_{i=1}^{m}\frac{p_i l_i^2}{2\tau_N}\right)\|x^{N+1}-x^N\|^2\right\}\\
&\le \frac{1}{N}\cdot \mathbb{E}\left[\frac{\eta_1}{2}\|x-x^1\|^2 + \sum_{i=1}^{m}\frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2\right],
\end{aligned}
$$

where we obtained the last inequality by using (3.5) and observing that

$$
\sum_{i=1}^{m} p_i l_i^2 = \sum_{i=1}^{m} p_i^2 \frac{l_i^2}{p_i} \le \max_{1\le i\le m}\left\{\frac{l_i^2}{p_i}\right\}\sum_{i=1}^{m} p_i^2 < \max_{1\le i\le m}\left\{\frac{l_i^2}{p_i}\right\} < \frac{\eta_t - L_f}{2\tau^{-1}}. \tag{4.38}
$$

Now, for $i = 1, \dots, m$, we set

$$
v_i := -\frac{1}{N}[p_i^{-1}(\tau_1 z_i^1 - \tau_N z_i^{N+1}) + M_i(x^N - x^{N+1})]. \tag{4.39}
$$

By setting $x = x^*$ in (4.38), we have $Mx^* = 0$ due to the optimality of x^* and thus obtain $\mathbb{E}[Q((x^N, z^N), (x^*, z)) - \langle v, z\rangle] \le \frac{1}{N}\cdot\mathbb{E}[\frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{p_i^{-1}\tau_1}{2}\|z_i^1\|^2]$ for any $z \in Z$, where the right-hand side is bounded and independent of z. Therefore, $\mathbb{E}[d(v, (x^N, z^N))] < \infty$, and hence we obtain (3.6) and $M\underline{x}^N = v$ a.s. due to Proposition 4.1.

Next, to estimate the consensus error $v = M\underline{x}^N$ defined in (4.39), we only need to bound $\|\tau_1 z_i^1 - \tau_N z_i^{N+1}\|$ and $\|x^N - x^{N+1}\|$. For a saddle point (x^*, z^*) of problem (1.3), we know that $Q((\underline{x}^N, \underline{z}^N), (x^*, z^*)) \ge 0$ due to the optimality of (x^*, z^*). Thus, by (4.37), we have

$$
\begin{aligned}
&\mathbb{E}\left[\frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2\right]\\
&\quad \le \mathbb{E}\left[\langle M(x^{N+1} - x^N),\ z^* - z^{N+1}\rangle - \sum_{i=1}^{m}\frac{\tau_N}{2p_i}\|z_i^* - z_i^{N+1}\|^2\right.\\
&\qquad \left. + \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2\right],\\
&\quad \le \mathbb{E}\left[\sum_{i=1}^{m}\frac{p_i}{2\tau_N}\|M_i(x^{N+1} - x^N)\|^2 + \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2\right],
\end{aligned}
$$

which implies that

$$
\mathbb{E}\left[\left(\frac{\eta_N - L_f}{4} - \sum_{i=1}^{m}\frac{p_i l_i^2}{2\tau_N}\right)\|x^{N+1} - x^N\|^2\right] \le \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2.
$$

Hence, we have

$$\mathbb{E}\left[\|x^{N+1} - x^N\|\right] \leq \sqrt{\frac{\eta_1}{2C}}\|x^* - x^1\| + \sum_{i=1}^{m}\sqrt{\frac{\tau_1}{2p_iC}}\|z_i^* - z_i^1\|, \tag{4.40}$$

where C is defined in (3.9) and hence is positive by (4.38). Similarly, (4.37) above implies that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{m}\frac{\tau_N}{2p_i}\|z_i^* - z_i^{N+1}\|^2\right] \\
&\quad\leq \mathbb{E}\Bigg[\langle M(x^{N+1} - x^N), z^* - z^{N+1}\rangle - \frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 \\
&\qquad + \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2\Bigg], \\
&\quad\leq \mathbb{E}\Bigg[\sum_{i=1}^{m} l_i^2\|x^{N+1} - x^N\|\|z_i^* - z_i^{N+1}\| - \frac{\eta_N - L_f}{4}\|x^{N+1} - x^N\|^2 \\
&\qquad + \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2\Bigg], \\
&\quad\leq \mathbb{E}\left[\sum_{i=1}^{m}\frac{l_i^2\|z_i^* - z_i^{N+1}\|^2}{\eta_N - L_f} + \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2\right],
\end{aligned} \tag{4.41}$$

which implies that

$$\mathbb{E}\left[\sum_{i=1}^{m}\left(\frac{\tau_N}{2p_i} - \frac{l_i^2}{\eta_N - L_f}\right)\|z_i^* - z_i^{N+1}\|^2\right] \leq \frac{\eta_1}{2}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i}\|z_i^* - z_i^1\|^2. \tag{4.42}$$

Therefore, we have

$$\mathbb{E}[\|z^* - z^{N+1}\|^2] \leq \frac{\eta_1}{2\underline{C}}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\frac{\tau_1}{2p_i\underline{C}}\|z_i^* - z_i^1\|^2,$$

where $\underline{C}$ is defined in (3.9) and hence must be positive due to (4.36). Notice that $\|z^1 - z^{N+1}\|^2 \leq 2\|z^* - z^1\|^2 + 2\|z^* - z^{N+1}\|^2$, we can have an estimate for $\|z^* - z^{N+1}\|^2$ as following:

$$\mathbb{E}[\|z^1 - z^{N+1}\|^2] \leq \frac{\eta_1}{\underline{C}}\|x^* - x^1\|^2 + \sum_{i=1}^{m}\left(2 + \frac{\tau_1}{p_i\underline{C}}\right)\|z_i^* - z_i^1\|^2,$$

from which we have

$$\mathbb{E}[\|z^1 - z^{N+1}\|] \le U\|x^* - x^1\| + \sum_{i=1}^{m} V_i\|z_i^* - z_i^1\|, \tag{4.43}$$

where U and V_i are defined in (3.8). Therefore, combining (4.39), (4.40) and (4.43) yields (3.7). □

5. Numerical Experiments

In this section, we present several numerical results of Algorithm 1. on synthetic DCO problem on networks. We simulate five two-dimensional (2D) lattice networks with different sizes: 2 × 5, 3 × 6, 3 × 7, 3 × 8 and 5 × 8. The size m of the networks are hence 10, 18, 21, 24 and 40, respectively. We also simulated an Erdős–Rényi random network of size $m = 50$ and average degree 4. The dimension of unknown x is set to $n = 5$. We randomly generate the ground truth $x_{\text{true}} \in \mathbb{R}^n$. For each node i in the simulated network, we randomly generate matrices $A_i \in \mathbb{R}^{q\times n}$ with $q = 5$, and normalize each column of A_i. Then b_i is obtained by $b_i = A_i x_{\text{true}} + \epsilon_i$, where noise ϵ_i is generated from normal distribution $\mathcal{N}(0, 0.0001)$. The objective function $f_i(x)$ of node i is defined as $\frac{1}{2}\|A_i x - b_i\|^2$, and $f(x) = \sum_{i=1}^{m} f_i(x)$. Then

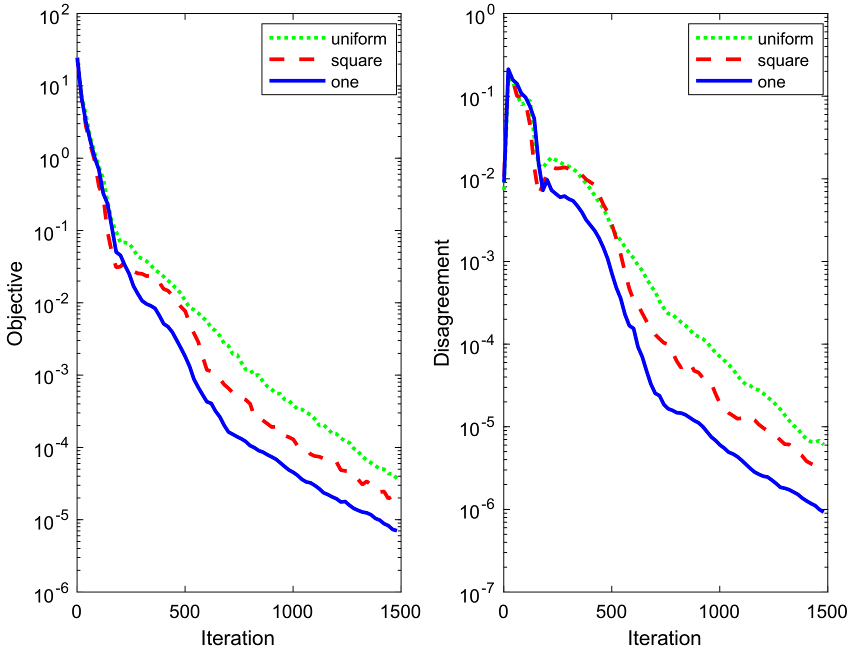

Fig. 1. Comparisons on decentralized least squares by local communication on 2×5 network. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

we know that the Lipschitz constant L_f of ∇f is given by $\max_{1\le i\le m} L_i$, where L_i is the Lipschitz constant of ∇f_i, $L_i = \|A_i^\top A_i\|_2$. The initialization of x_i is set to be 0 for all nodes.

The performance is evaluated by the primal residual $f(x^t) = \frac{1}{2}\ \sum_{i=1}^m \|A_i x_i^t - b_i\|^2$ and disagreement $\sum_{i=1}^m \|x_i^t - x_{\text{mean}}^t\|^2$, where x_{mean}^t is the average value of x_i^t at tth iteration, $x_{\text{mean}}^t = \frac{1}{m}\sum_{i=1}^m x_i^t$. We test three sampling strategies: uniform sampling (curves labeled by "uniform" in figures), importance sampling of $p_i \propto l_i$ (curves labeled by "one" in figures), and importance sampling of $p_i \propto l_i^2$ (curves labeled by "square" in figures). In all tests, we set $\tau_t = \tau = 2$, and η_t as in (3.10) for the "uniform" case and (3.2) for the "one" and "square" cases (with $\alpha = 1$ and 2, respectively). The primal residual and disagreement versus iteration t for the six networks are plotted in Figs. 1–6, respectively. In Fig. 2, we can observe that the primal residual $f(x^t)$ and disagreement both converges to 0 as stated in theoretical analysis. Besides, we see that importance sampling show some advantages over uniform sampling, both in the primal residual and disagreement. The step size η in RIPD with importance sampling is greater than step size in uniform sampling. Larger step size may accelerate the convergence of proposed method and result better practical performance.

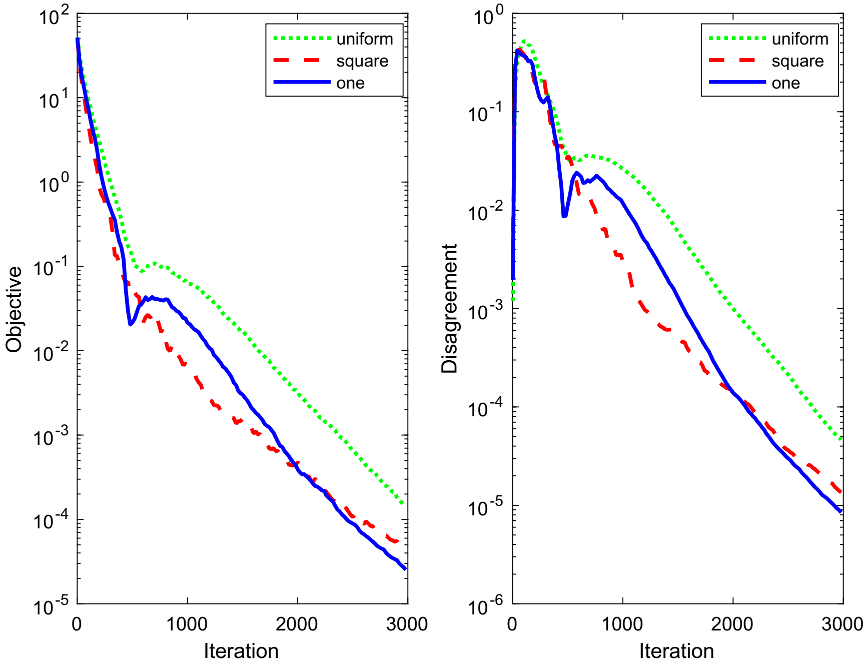

Fig. 2. Comparisons on decentralized least squares by local communication on 3×6 network. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

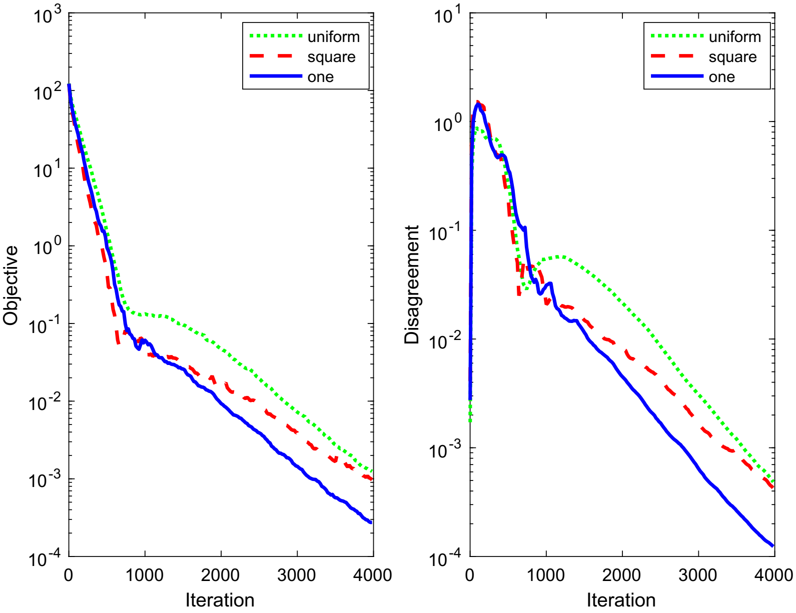

Fig. 3. Comparisons on decentralized least squares by local communication on 3×7 network. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

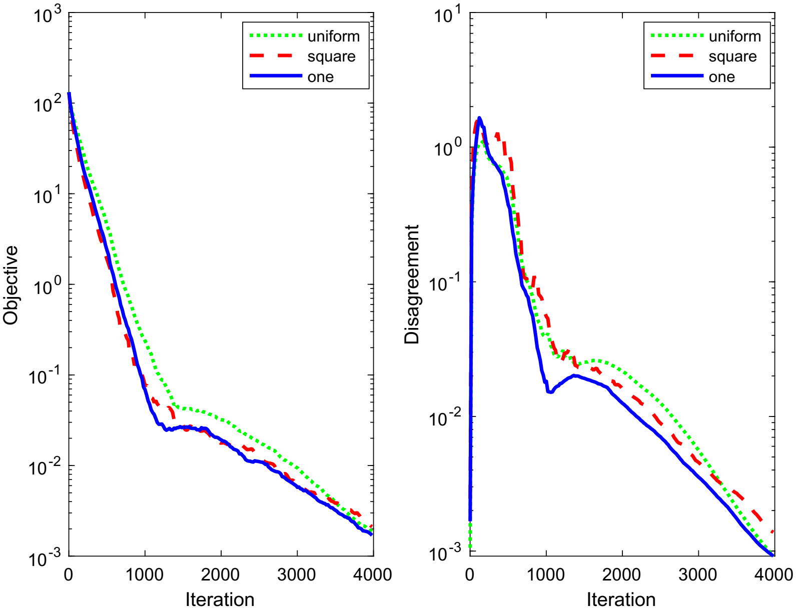

Fig. 4. Comparisons on decentralized least squares by local communication on 3×8 network. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

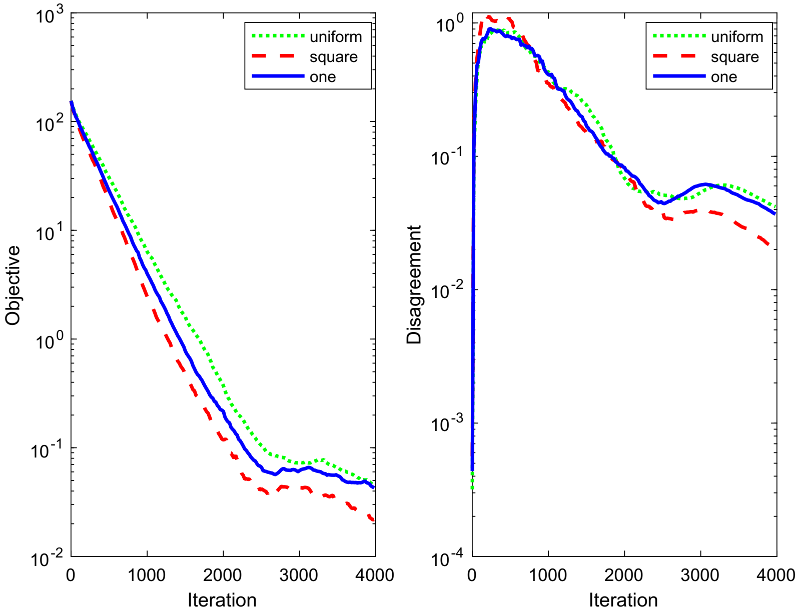

Fig. 5. Comparisons on decentralized least squares by local communication on 5 × 8 network. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

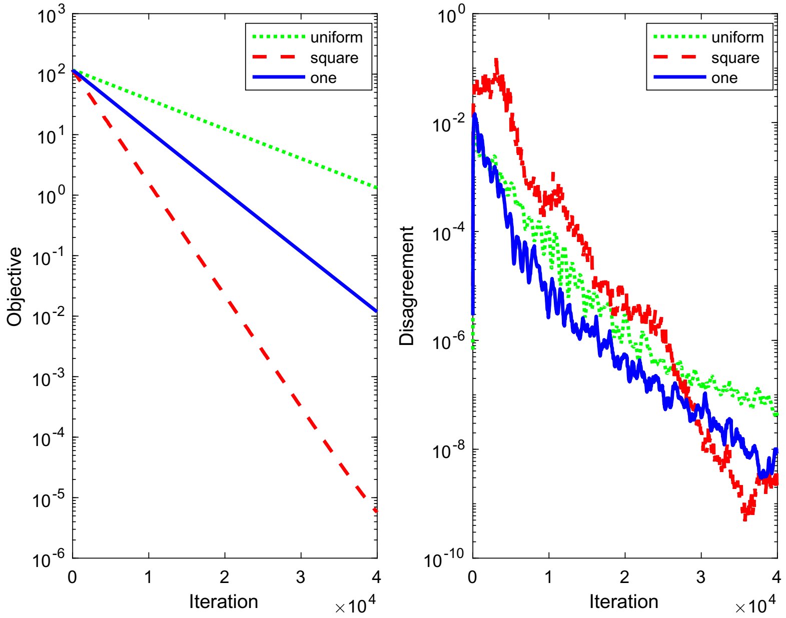

Fig. 6. Comparisons on decentralized least squares by local communication on an Erdos random network of size 50 and average degree 4. The agents in each iteration are activated with different probability distributions. Left: the objective function values. Right: the disagreements.

6. Conclusion

We present a RIPD method for solving a class of smooth convex optimization problems. The dual variable over the network in each iteration is only updated at a randomly selected node, whereas the dual variables elsewhere remain the same as in the previous iteration. Thus, the communication only occurs in the neighborhood of the selected node in each iteration and hence can greatly reduce the chance of communication delay and failure in the standard fully synchronized consensus algorithms. The proposed method converges to optimal solution with a provable rate of $O(1/t)$, where t is the iteration number.

Acknowledgement

This research was supported in part by National Science Foundation (NSF) under grants DMS-1620342, DMS-1719932, CMMI-1745382, DMS-1818886 and DMS1925263.

References

[1] N. S. Aybat, Z. Wang, T. Lin and S. Ma, Distributed linearized alternating direction method of multipliers for composite convex consensus optimization, *IEEE Trans. Automat. Control* **63**(1) (2018) 5–20.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*: *Numerical Methods*, Vol. 23 (Prentice Hall, Englewood Cliffs, NJ, 1989).

[3] P. Bianchi, W. Hachem and F. Iutzeler, A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization, preprint (2014), arXiv 1407.0898.

[4] P. Bianchi, W. Hachem and F. Iutzeler, A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization, *IEEE Trans. Automat. Control* **61**(10) (2016) 2947–2957.

[5] D. Blatt, A. O. Hero and H. Gauchman, A convergent incremental gradient method with a constant step size, *SIAM J. Optim.* **18**(1) (2007) 29–51.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* **3**(1) (2011) 1–122.

[7] T.-H. Chang, M. Hong and X. Wang, Multi-agent distributed optimization via inexact consensus admm, *IEEE Trans. Signal Process.* **63**(2) (2015) 482–497.

[8] P. L. Combettes and J.-C. Pesquet, Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping, *SIAM J. Optim.* **25**(2) (2015) 1221–1248.

[9] A. Defazio, F. Bach and S. Lacoste-Julien, SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives, in *NIPS Proc. Conf. Advances in Neural Information Processing Systems* (Montréal Canada, 2014), pp. 1646–1654.

[10] J. C. Duchi, S. Chaturapruek and C. Ré, Asynchronous stochastic convex optimization, preprint (2015), arXiv:1508.00882.

[11] T. Erseghe, D. Zennaro, E. Dall'Anese and L. Vangelista, Fast consensus by the alternating direction multipliers method, *IEEE Trans. Signal Process.* **59**(11) (2011) 5523–5537.

[12] P. A. Forero, A. Cano and G. B. Giannakis, Consensus-based distributed support vector machines, *J. Mach. Learn. Res.* **11** (2010) 1663–1707.

[13] L. Gan, U. Topcu and S. H. Low, Optimal decentralized protocol for electric vehicle charging, *IEEE Trans. Power Syst.* **28**(2) (2013) 940–951.

[14] E. Hazan and H. Luo, Variance-reduced and projection-free stochastic optimization, in *Int. Conf. Machine Learning* (New York, USA, 2016), pp. 1263–1271.

[15] M. Hong and T.-H. Chang, Stochastic proximal gradient consensus over random networks, *IEEE Trans. Signal Process.* **65**(11) (2017) 2933–2948.

[16] M. Hong, Z.-Q. Luo and M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM J. Optim.* **26**(1) (2016) 337–364.

[17] F. Iutzeler, P. Ciblat, W. Hachem and J. Jakubowicz, New broadcast based distributed averaging algorithm over wireless sensor networks, in *2012 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Kyoto, Japan, 2012), pp. 3117–3120.

[18] D. Jakovetić, J. M. Moura and J. Xavier, Linear convergence rate of a class of distributed augmented lagrangian algorithms, *IEEE Trans. Automat. Control* **60**(4) (2015) 922–936.

[19] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in *NIPS Proc. Conf. Advances in Neural Information Processing Systems* (Lake Tahoe, USA, 2013), pp. 315–323.

[20] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin and M. I. Jordan, Mlbase: A distributed machine-learning system, in *CIDR*, Vol. 1 (2013), pp. 2–1.

[21] G. Lan, S. Lee and Y. Zhou, Communication-efficient algorithms for decentralized and stochastic optimization, preprint (2017), arXiv:1701.03961.

[22] G. Lan and Y. Zhou, An optimal randomized incremental gradient method, *Math. Program.* **171**(1–2) (2017) 1–49.

[23] G. Lan and Y. Zhou, Random gradient extrapolation for distributed and stochastic optimization, *SIAM J. Optim.* **28**(4) (2018) 2753–2782.

[24] M. Li, D. G. Andersen, A. J. Smola and K. Yu, Communication efficient distributed machine learning with the parameter server, in *NIPS Proc. Conf. Advances in Neural Information Processing Systems* (Montreal, Canada, 2014), pp. 19–27.

[25] H. Lin, J. Mairal and Z. Harchaoui, A universal catalyst for first-order optimization, in *NIPS Proc. Conf. Advances in Neural Information Processing Systems* (Montreal, Canada, 2015), pp. 3384–3392.

[26] Q. Ling, W. Shi, G. Wu and A. Ribeiro, Dlm: Decentralized linearized alternating direction method of multipliers, *IEEE Trans. Signal Process.* **63**(15) (2015) 4051–4064.

[27] C.-H. Lo and N. Ansari, Decentralized controls and communications for autonomous distribution networks in smart grid, *IEEE Trans. Smart Grid* **4**(1) (2013) 66–77.

[28] A. Makhdoumi and A. Ozdaglar, Broadcast-based distributed alternating direction method of multipliers, in *2014 52nd Annual Allerton Conf. Communication*, *Control*, *and Computing* (*Allerton*) (IEEE, Monticello, Illinois, USA, 2014), pp. 270–277.

[29] A. Mokhtari and A. Ribeiro, Dsa: Decentralized double stochastic averaging gradient algorithm, *J. Mach. Learn. Res.* **17**(1) (2016) 2165–2199.

[30] J. F. Mota, J. M. Xavier, P. M. Aguiar and M. Puschel, D-admm: A communication-efficient distributed algorithm for separable optimization, *IEEE Trans. Signal Process.* **61**(10) (2013) 2718–2723.

[31] A. Nedić, Asynchronous broadcast-based convex optimization over a network, *IEEE Trans. Automat. Control* **56**(6) (2011) 1337–1351.

[32] A. Nedić and A. Olshevsky, Distributed optimization over time-varying directed graphs, *IEEE Trans. Automat. Control* **60**(3) (2015) 601–615.

[33] A. Nedic and A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, *IEEE Trans. Automat. Control* **54**(1) (2009) 48–61.

[34] J.-C. Pesquet and A. Repetti, A class of randomized primal-dual algorithms for distributed optimization, preprint (2014), arXiv:1406.6404.

[35] M. Rabbat and R. Nowak, Distributed optimization in sensor networks, in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks* (ACM, 2004), pp. 20–27.

[36] I. D. Schizas, A. Ribeiro and G. B. Giannakis, Consensus in ad hoc WSNS with noisy links—part I: Distributed estimation of deterministic signals, *IEEE Trans. Signal Process.* **56**(1) (2008) 350–364.

[37] M. Schmidt, N. Le Roux and F. Bach, Minimizing finite sums with the stochastic average gradient, *Math. Program.* **162**(1–2) (2017) 83–112.

[38] W. Shi, Q. Ling, G. Wu and W. Yin, Extra: An exact first-order algorithm for decentralized consensus optimization, *SIAM J. Optim.* **25**(2) (2015) 944–966.

[39] W. Shi, Q. Ling, K. Yuan, G. Wu and W. Yin, On the linear convergence of the admm in decentralized consensus optimization, *IEEE Trans. Signal Process.* **62**(7) (2014) 1750–1761.

[40] B. Sirb and X. Ye, Decentralized consensus algorithm with delayed and stochastic gradients, *SIAM J. Optim.* **28**(2) (2018) 1232–1254.

[41] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi and R. LaHusen, Air-dropped sensor network for real-time high-fidelity volcano monitoring, in *Proc. 7th Int. Conf. Mobile Systems, Applications, and Services* (ACM, 2009), pp. 305–318.

[42] E. Wei and A. Ozdaglar, On the $o\,(1 = k)$ convergence of asynchronous distributed alternating direction method of multipliers, in *Global Conf. Signal and Information Processing (GlobalSIP), 2013 IEEE* (IEEE, 2013), pp. 551–554.

[43] T. Wu, K. Yuan, Q. Ling, W. Yin and A. H. Sayed, Decentralized consensus optimization with asynchrony and delays, in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers* (Pacific Grove, CA, USA, 2016).

[44] L. Xiao and T. Zhang, A proximal stochastic gradient method with progressive variance reduction, *SIAM J. Optim.* **24**(4) (2014) 2057–2075.

[45] G. Zhang and R. Heusdens, Distributed optimization using the primal-dual method of multipliers, *IEEE Trans. Signal Inform. Process. Networks* **4**(1) (2017) 173–187.

[46] R. Zhang and J. Kwok, Asynchronous distributed admm for consensus optimization, in *Proc. 31st Int. Conf. Machine Learning (ICML-14)* (2014), pp. 1701–1709.

[47] L. Zhao, W.-Z. Song, X. Ye and Y. Gu, Asynchronous broadcast-based decentralized learning in sensor networks, *Int. J. Parallel, Emergent Distrib. Syst.* **33**(6) (2017), pp. 1–19.

[48] S. Zheng and J. T. Kwok, Stochastic variance-reduced admm, preprint (2016), arXiv:1604.07070.

[49] H. Zhu, A. Cano and G. B. Giannakis, Distributed consensus-based demodulation: Algorithms and error analysis, *IEEE Trans. Wireless Commun.* **9**(6) (2010), 2044–2054.

[50] M. Zhu and S. Martínez, *Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments* (Springer, 2015).