Weighted ADMM for Fast Decentralized Network Optimization

Qing Ling, Yaohua Liu, Wei Shi, and Zhi Tian

***Abstract*—In this paper, we propose a weighted alternating direction method of multipliers (ADMM) to solve the consensus optimization problem over a decentralized network. In the proposed algorithm, every node holds its local objective function, exchanges its current iterate with a subset of neighbors, carries on local computation, and eventually reaches an optimal and consensual solution that minimizes the summation of the local objective functions. Compared with the conventional ADMM that is popular in decentralized network optimization, the weighted ADMM is able to reduce the communication cost spent in the optimization process through tuning the weight matrices, which assign beliefs on the neighboring iterates. We first prove convergence and establish linear convergence rate of the weighted ADMM. Second, we maximize the derived convergence speed and obtain the best weight matrices on a given topology. Third, observing that exchanging information with all the neighbors is expensive, we maximize the convergence speed while limit the number of communication arcs. This strategy finds a subset of arcs within the underlying topology to fulfill the optimization task while leads to a favorable tradeoff between the number of iterations and the communication cost per iteration. Numerical experiments demonstrate advantages of the weighted ADMM over its conventional counterpart in expediting the convergence speed and reducing the communication cost.**

***Index Terms*—Alternating direction method of multipliers (ADMM), Decentralized network, communication cost, consensus optimization.**

I. Introduction

PROPELLED by the rapid progress of data acquisition, communication and networking technologies, information processing and decision making over decentralized networks have attracted noticeable research interest in these years. A group of geographically distributed nodes, which are equipped with sensing, communicating and computing abilities, collaboratively accomplish an information processing or decision making task over an underlying network topology. A typical task is decentralized consensus optimization, in which n nodes solve

$$\min_{x} \quad \sum_{i=1}^{n} f_i(x). \tag{1}$$

Here $x \in \mathcal{R}^p$ is the common optimization variable and $f_i : \mathcal{R}^p \to \mathcal{R}$ is the local objective function of node i. Such a problem formulation appears in various applications, for example, wireless communications and networking [2], [3], spectrum sensing of cognitive radios [4], [5], monitoring and optimization of smart grids [6], [7], distributed control of networked robots [8]–[10], to name a few.

In a decentralized algorithm that solves (1), every node holds its local objective function, exchanges its current iterate with a subset of neighbors, carries on local computation, and eventually reaches an optimal solution that is consensual to all the nodes. In this optimization process, communication cost is one of the key considerations of implementation. Reducing the amount of information exchange among the nodes alleviates burden on bandwidth, improves system robustness, and enables fast information processing and decision making. In this paper, we propose a weighted alternating direction method of multipliers (ADMM) to solve (1), aiming at reducing the communication cost via a principled design.

A. Related Works

Decentralized optimization algorithms that solve (1) include gradient/subgradient methods [11], [12] and their accelerated versions [13], diffusion methods [14], [15], dual averaging methods [16], [17], Newton methods [18], [19], and ADMM [2], [3], [20]–[22]. Among these algorithms, the decentralized ADMM has shown fast convergence in both practice and theory. When (1) is a convex program, ADMM converges to the optimal solution at a sublinear rate of $O(1/k)$ with k being the number of iterations [3]. Its linear rate of $O(\tau^k)$, where $\tau \in (0, 1)$ is a topology-dependent constant, is established in [23] given that the local objective functions are strongly convex. ADMM is also able to utilize special composite structures or introduce surrogates of the local objective functions so as to significantly simplify the computation, while still keep its favorable convergence properties [24]–[27].

ADMM is originally developed to solve centralized optimization problems with linear constraints [28], [29]. It splits primal optimization variables into two sets, minimizes them in an alternating direction manner, and follows with a dual gradient ascent step. Its sublinear convergence rate is established in [30] and [31] and linear convergence rate is established in [32] and [33]. ADMM became popular because of its simplicity

Manuscript received October 27, 2015; revised April 6, 2016 and July 8, 2016; accepted August 10, 2016. Date of publication August 25, 2016; date of current version September 21, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsung-Hui Chang. The work of Q. Ling was supported in part by the China NSF under Grant 61573331, in part by the Anhui NSF under Grant 1608085QF130, and in part by the CAS under Grant XDA06011203. The work of Z. Tian was supported in part by the US NSF under Grant AST-1547329. This paper was presented in part at the 41th International Conference on Acoustics, Speech, and Signal Processing, Shanghai, China, March 20–25, 2016 [1].

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

W. Shi is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: wilburs@illinois.edu).

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2602803

in implementation and stability of computation. Readers are referred to the survey paper [34] for some of its applications.

To apply ADMM in decentralized optimization, we need to first introduce a set of consensus constraints to the unconstrained problem (1), and then use the technique of variable splitting. To be specific, we introduce at every node and every arc a local copy of x, and for every arc, force the local copies of the arc and the two attached nodes to be equal. This way, ADMM has two sets of local copies, those of the arcs and those of the nodes, which are alternatingly optimized. Of particular note, the local copies of the arcs are eventually eliminated in the resultant algorithm. Not surprisingly, convergence speed of the conventional decentralized ADMM is determined by condition numbers of the local objective functions, spectral properties of the underlying topology and the stepsize of dual gradient ascent (namely, the ADMM penalty factor) [23]. However, the conventional ADMM is unable to achieve the best communication efficiency due to two reasons. First, there is only one parameter, the ADMM penalty factor, which can be tuned to maximize the convergence speed and consequently minimize the required number of iterations. Second, at every iteration, every node has to exchange its current iterate with all of its neighbors, which leads to a large amount of information exchange per iteration.

B. Our Contributions and Paper Organization

This paper proposes a weighted ADMM to solve the decentralized optimization problem (1) and address the two aforementioned disadvantages of the conventional ADMM. Intuitively, one can assign different weights to the consensus constraints on different arcs. Through tuning the weights, we have more flexibility to maximize the convergence speed than in the conventional ADMM. Furthermore, by setting some weights as zeros, we are able to avoid information exchange over a subset of arcs and hence reduce the communication cost per iteration. The intuitive idea of weight tuning is made rigorous by our analytical delineation of the convergence speed as a function of the weights, which is one of the main contributions of this paper as well.

Section II develops the weighted ADMM following this intuitive idea and discusses its connection with the conventional ADMM. Section III proves convergence and establishes linear rate of convergence for the weighted ADMM. We provide an explicit expression that shows how the convergence speed is determined by the weight matrices. Such an expression enables optimal design of the weights, which leads to two design strategies we develop in Section IV. The first one gives two optimal design strategies. The first one simply maximizes the convergence speed, while the second one further confines the number of communication arcs for the sake of reducing the amount of information exchange at every iteration. Numerical experiments in Section V demonstrate advantages of the weighted ADMM over its conventional counterpart in expediting the convergence speed and reducing the communication cost. Section VI concludes the paper.

C. Notations

Throughout the paper, define $e = [1; 1; \cdots; 1] \in \mathcal{R}^n$ as an all-one vector. For a matrix M, define $\|M\|_{\rm F}$ as its Frobenius norm and $\|M\|_0$ as its number of nonzero elements (or its pseudo ℓ_0 norm by convention). Given a positive semidefinite matrix G, the G-norm of M is defined as $\|M\|_G \triangleq \sqrt{\langle M, GM\rangle}$ where $\langle\cdot,\cdot\rangle$ denotes the inner product operator. The null space of M is denoted by $\text{Null}(M)$ and the span of M by $\text{Span}(M)$. The largest and the smallest nonzero eigenvalues of G are denoted by $\sigma_{\max}(G)$ and $\tilde{\sigma}_{\min}(G)$, respectively. For a square matrix A, denote $\text{OffDiag}(A)$ as a matrix whose off-diagonal elements are identical to those of A and diagonal elements are zeros.

II. Algorithm Development

In this section, we propose a weighted ADMM to solve the decentralized network optimization problem (1), aiming at reducing the communication cost of the conventional ADMM. Connection between the two algorithms is also explained.

A. Problem Statement

Network model. Throughout this paper, we consider a bidirectionally connected network consisting of n nodes and r edges. We describe the network as an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes with cardinality $|\mathcal{V}| = n$ and $\mathcal{E}$ is the set of arcs with cardinality $|\mathcal{E}| = 2r$. Nodes i and j are neighbors of each other if $(i, j) \in \mathcal{E}$ and, by the symmetry of the network, $(j, i) \in \mathcal{E}$. The set of node i's neighbors is denoted as $\mathcal{N}_i$, whose cardinality $|\mathcal{N}_i|$ is the degree of node i.

Communication model. This paper designs an iterative decentralized algorithm to solve the network optimization problem (1). At every iteration of the algorithm, every node i communicates with a set of other nodes $\mathcal{C}_i$, sending its current local estimate and receiving the others'. It is assumed that the communication and iterative updating steps are synchronous among all the nodes. Furthermore, in order to guarantee that the algorithm is decentralized, every node is only allowed to communicate with those nodes in its neighbor set; that is to say, for every node i we must have $\mathcal{C}_i \subseteq \mathcal{N}_i$. Notice that $\mathcal{N}_i$ comes from physical limits of the network while $\mathcal{C}_i$ is user-designed. Intuitively, taking $\mathcal{C}_i \subset \mathcal{N}_i$ will decrease the convergence speed but save the communication cost per iteration, as we will discuss in the rest of this paper.

We consider the costs of two communication schemes, broadcast and unicast. In the broadcast scheme, upon sending a local estimate, node i broadcasts once to all the nodes in $\mathcal{C}_i$. Suppose that the dimension of each local estimate is p. The communication cost of the whole network at every iteration is pn. In the unicast scheme, upon sending a local estimate, node i contacts every individual node in $\mathcal{C}_i$ separately. The communication cost is hence $p\sum_{i=1}^n |\mathcal{C}_i|$ per iteration. Here the communication costs are measured in terms of the energy and time spent in sending messages. Note that for both the broadcast and unicast schemes, the network receives $\sum_{i=1}^n |\mathcal{C}_i|$ local estimates at every iteration. The receiving costs of management and post-processing are hence proportional to $p\sum_{i=1}^n |\mathcal{C}_i|$. To simplify the discussion, this paper mainly focuses on the sending costs, because the receiving costs are proportional to the sending cost of the unicast scheme.

B. Weighted ADMM

In the weighted ADMM, every node i maintains a local variable $x_i \in \mathcal{R}^p$, which is a copy of the optimization variable x in (1). Node i also keeps a local variable $\lambda_i \in \mathcal{R}^p$, which plays the role of Lagrange multiplier as we will explain in Section II-C. Both x_i and λ_i are updated using information collected from the nodes in $\mathcal{C}_i$. However, only x_i is transmitted to the nodes in $\mathcal{C}_i$; λ_i is kept private.

Collect all local variables x_i and λ_i in two matrices

$$X \triangleq \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \in \mathcal{R}^{n\times p} \quad \text{and} \quad \Lambda \triangleq \begin{pmatrix} \lambda_1^T \\ \lambda_2^T \\ \vdots \\ \lambda_n^T \end{pmatrix} \in \mathcal{R}^{n\times p}.$$

Define an aggregate objective function $f(X) = \sum_{i=1}^n f_i(x_i)$. Denote $\mathcal{D} \subset \mathcal{R}^{n\times n}$ as the set of n-by-n diagonal matrices whose diagonal elements are positive, and $\mathcal{A} \subset \mathcal{R}^{n\times n}$ as the set of n-by-n symmetric matrices whose (i,j)-th elements are zeros if nodes i and j are neither neighbors nor the same. The matrix form of the weighted ADMM update is given by

$$\begin{aligned} X^{k+1} &= \arg\min_X f(X) + \left\langle X, \Lambda^k - (D+A)X^k \right\rangle \\ &\qquad + \langle X, DX\rangle, \\ \Lambda^{k+1} &= \Lambda^k + (D-A)X^{k+1}. \end{aligned} \tag{2}$$

In (2), $D \in \mathcal{D}$ is a diagonal matrix and its (i,i)-th element is positive and denoted by d_{ii}; $A \in \mathcal{A}$ is a symmetric matrix satisfying that its (i,j)-th element $a_{ij} = 0$ if nodes i and j are neither neighbors nor the same. Given a matrix A, define $\mathcal{C}_i = \{j | a_{ij} \neq 0 \text{ and } i \neq j\}$, which guarantees $\mathcal{C}_i \subseteq \mathcal{N}_i$.

Splitting the computation in the matrix form (2) to individual nodes, the update of node i is given by

$$\begin{aligned} x_i^{k+1} &= \arg\min_{x_i} f_i(x_i) + \left\langle x_i, \lambda_i^k - d_{ii}x_i^k - \sum_{j=1}^n a_{ij}x_j^k \right\rangle \\ &\qquad + d_{ii}\|x_i\|^2, \\ \lambda_i^{k+1} &= \lambda_i^k + d_{ii}x_i^{k+1} - \sum_{j=1}^n a_{ij}x_j^{k+1}. \end{aligned} \tag{3}$$

The algorithm can be implemented in a decentralized manner. In the update of x_i^{k+1}, node i needs to calculate the summation $\sum_{j=1}^n a_{ij}x_j^k$, which only requires the previous iterates x_i^k and $x_j^k, j \in \mathcal{C}_i$, as $a_{ij} = 0$ if $j \neq i$ and $j \notin \mathcal{C}_i$. The objective function $f_i(x_i)$ and the previous Lagrange multiplier λ_i^k are also locally available. Similarly, in the update of λ_i^{k+1}, node i calculates the weighted summation $\sum_{j=1}^n a_{ij}x_j^{k+1}$ of the current local estimates; this can be done through communication with its neighbors.

The weighted ADMM is outlined in Algorithm 1. At time $k = 0$ we initialize the local variables to $x_i^0 = 0$ and $\lambda_i^0 = 0$. For all subsequent times, node i runs the update (3), as shown in Step 2 and Step 4 of Algorithm 1. Implementation of Step 2 requires neighboring iterates x_j^k from the previous iteration. Implementation of Step 4 requires current neighboring iterates x_j^{k+1}, which become available through the exchange implemented in Step 3. This exchange step also makes the neighboring iterates available for the update in Step 2 with respect to the following time index. Note that in the exchange step, the communication scheme is broadcast, as we have assumed in Section II-A.

Algorithm 1: Weighted ADMM run by node i.

Require: Initialize local estimates to $x_i^0 = 0$ and $\lambda_i^0 = 0$.

1: **for** iterations $k = 1, 2, \ldots$ **do**

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

$$\begin{aligned} x_i^{k+1} &= \arg\min_{x_i} f_i(x_i) \\ &\qquad + \left\langle \lambda_i^k - d_{ii}x_i^k + \sum_{j=1}^n a_{ij}x_j^k, x \right\rangle + d_{ii}\|x_i\|^2. \end{aligned}$$

3: Transmit x_i^{k+1} & receive x_j^{k+1} from neighbors $j \in \mathcal{C}_i \subseteq \mathcal{N}_i$.

4: Update local Lagrange multiplier λ_i^{k+1} as

$$\lambda_i^{k+1} = \lambda_i^k + d_{ii}x_i^{k+1} - \sum_{j=1}^n a_{ij}x_j^{k+1}.$$

5: **end for**

C. Connection Between Weighted ADMM and Conventional ADMM

To unveil the connection between the proposed weighted ADMM and the conventional one, observe that [23] gives the matrix form of the conventional ADMM as

$$\begin{aligned} X^{k+1} &= \arg\min_X f(X) + \left\langle X, \Lambda^k - cUX^k \right\rangle \\ &\qquad + \left\langle X, c\frac{U+V}{2}X \right\rangle, \\ \Lambda^{k+1} &= \Lambda^k + cVX^{k+1}. \end{aligned} \tag{4}$$

Therein, c is the ADMM penalty factor for constraint violation, and also the stepsize of dual gradient ascent; U and V are the signless and signed Laplacian matrices of the network, respectively; $(U+V)/2$ is the diagonal node degree matrix whose i-th diagonal element is $|\mathcal{N}_i|$, the degree of node i. Such an algorithm is developed following the ADMM routine: introducing local copies of x at all the arcs and the nodes so as to form consensus constraints; minimizing the augmented Lagrangian function regarding the local copies at the arcs and those at the nodes, respectively; then moving a dual ascent step to update the Lagrange multipliers. Note that the local copies of the arcs are eliminated eventually due to the special structure of the problem formulation.

Comparing (2) and (4), we can find that the conventional ADMM is a special case of the weighted ADMM by setting $D = c(U+V)/2$ and $A = c(U-V)/2$. In this case, D is the degree matrix whose (i,i)-th element d_{ii} denotes the degree of node i, while A is the adjacent matrix whose (i,j)-th element a_{ij} equals to one if nodes i and j are connected and zero otherwise. Observe that such choices of D and A satisfy the requirements

of the weighted ADMM given by Section II-B. In fact, they correspond to assigning the same weights to all the consensus constraints. In contrast, the weighted ADMM essentially assigns different weights to the consensus constraints at different arcs.

This difference enables the weighted ADMM to achieve a better communication efficiency than the conventional one. First, in the conventional ADMM, updating the primal variable x_i and the Lagrange multiplier λ_i in node i involves communication with all the neighbors j in $\mathcal{N}_i$, because of the structures of U and V. Therefore, the conventional ADMM has a fixed communication cost per iteration given the network topology, which is pn for broadcast and $p\sum_{i=1}^n |\mathcal{N}_i|$ for unicast in the whole network. Contrarily, the weighted ADMM is able to reduce the communication cost by letting every node communicate with less neighbors; this can be done through wisely choosing the matrix A such that $\sum_{i=1}^n |\mathcal{C}_i|$ is significantly less than $\sum_{i=1}^n |\mathcal{N}_i|$. Second, the conventional ADMM can optimize its convergence speed through tuning the penalty factor c, since U and V are fixed given the network topology. Whereas, the weighted ADMM has the flexibility of tuning two matrices, A and D. Consequently, the weighted ADMM has the potential to achieve a faster convergence speed than its conventional counterpart.

In summary, comparing to the conventional ADMM, the weighted ADMM is able to both reduce the communication cost per iteration and accelerate the convergence speed. Thus, the weighted ADMM can achieve a target solution accuracy using less communication cost. In the following sections, we theoretically establish convergence and linear convergence rate of the weighted ADMM (Section III), minimize its communication cost based upon the analyses (Section IV), and demonstrate its effectiveness through numerical experiments (Section V).

III. Convergence and Linear Rate of Convergence

In this section, we give conditions on the weight matrices D and A under which the weighted ADMM converges to an optimal solution (Section III-B). We also establish its linear rate of convergence when every local objective functions have Lipschitz continuous gradients and are strongly convex (Section III-C). The convergence speed is dependent on condition numbers of the local objective functions and spectral properties of the weight matrices D and A. This fact motivates us to maximize the convergence speed through tuning D and A in the next section. Note that the tools of analyses used in this paper and in the conventional ADMM share similarities. However, the proof techniques have been adapted to fit the special algorithmic structures of the weighted ADMM.

A. Assumptions and Supporting Lemmas

We begin with several assumptions and supporting lemmas. Unless otherwise stated, the convergence results in this section are given under Assumptions from 1 through 4. Assumptions 1 and 2 are basic, requiring that the underlying network to be connected and the solution set not null, respectively.

Assumption 1 (Network connectivity): The network of n nodes are bidirectionally connected.

Assumption 2 (Solution existence): The solution set to (1), denoted by $\mathcal{X}^*$, is nonempty and has at least one bounded element.

Assumption 3 supposes that the local objective functions are convex and continuously differentiable. This assumption, along with Assumptions 1 and 2, is sufficient to prove convergence of the weighted ADMM.

Assumption 3 (Convexity and differentiability): The local objective functions f_i are convex and continuously differentiable.

To further establish linear rate of convergence, we require the local objective functions to have Lipschitz continuous gradients and be strongly convex, as stated in Assumption 4.

Assumption 4 (Lipschitz continuous gradients and strong convexity): The local objective functions f_i have Lipschitz continuous gradients. For node i, there is a positive constant $L_i > 0$ such that for any pair of points $\hat{x}$ and $\bar{x}$ it holds $\|\nabla f_i(\hat{x}) - \nabla f_i(\bar{x})\| \leq L_i\|\hat{x} - \bar{x}\|$. The maximum Lipschitz constant is $L = \max_i L_i$. The local objective functions f_i are strongly convex. For node i, there is a positive constant $\mu_i > 0$ such that for any pair of points $\hat{x}$ and $\bar{x}$ it holds $\langle \hat{x} - \bar{x}, \nabla f_i(\hat{x}) - \nabla f_i(\bar{x})\rangle \geq \mu_i\|\hat{x} - \bar{x}\|^2$. The minimum strong convexity constant is $\mu = \min_i \mu_i$.

Define two diagonal matrices $P \in \mathcal{R}^{n\times n}$ and $Q \in \mathcal{R}^{n\times n}$, whose i-th diagonal elements are L_i^2 and μ_i, respectively. Recall the definition of $f(X) = \sum_{i=1}^n f_i(x_i)$ where $X = [x_1^T; x_2^T; \cdots; x_n^T]$. For any pair of points $\hat{X} \in \mathcal{R}^{n\times n}$ and $\bar{X} \in \mathcal{R}^{n\times n}$, Assumption 4 yields

$$\|\nabla f(\hat{X}) - \nabla f(\bar{X})\|_{\mathrm{F}}^2 \leq \|\hat{X} - \bar{X}\|_P^2,$$

and

$$\langle \nabla f(\hat{X}) - \nabla f(\bar{X}), \hat{X} - \bar{X}\rangle \geq \|\hat{X} - \bar{X}\|_Q^2.$$

To facilitate analysis of the weighted ADMM, we rewrite its update in (2) to another form. Suppose that $D - A$ is positive semidefinite, which is necessary to the convergence of the weighted ADMM. Introducing a new series of matrices $\{Y^k \in \mathcal{R}^{n\times p}\}$ and observing the differentiability of f, we know that (2) is equivalent to

$$\begin{aligned} &\nabla f(X^{k+1}) + \sqrt{D-A}Y^k + 2DX^{k+1} - (D+A)X^k \\ &\qquad = 0, \\ &Y^{k+1} = Y^k + \sqrt{D-A}X^{k+1}. \end{aligned} \tag{5}$$

The equivalence of (2) and (5) is given by $\Lambda^k = \sqrt{D-A}Y^k$. To guarantee that such Y^k exists for every k, Λ^k must be in the column space of $D - A$ (also that of $\sqrt{D-A}$). This is not difficult to satisfy because if Λ^0 is initialized in the column space of $D - A$ (to be specific, $\Lambda^0 = 0$), then every Λ^k stays inside due to the recursion $\Lambda^{k+1} = \Lambda^k + (D - A)X^{k+1}$.

In the convergence analyses, we shall show that the weighted ADMM converges to an optimal solution of (1). This is done by showing that (X^k, Y^k) converges to an optimal pair (X^*, Y^*) defined by the following first-order optimality condition of (1).

Lemma 1 (First-order optimality condition): Suppose that $D - A \succeq 0$ and $\mathrm{Null}(D - A) = e$. Under Assumptions 2 and 3, the following two statements are equivalent.

- $X^* \triangleq \left[(x_1^*)^T; (x_2^*)^T; \cdots; (x_n^*)^T\right] \in \mathbb{R}^{n\times p}$ is consensual (namely, $x_1^* = x_2^* = \cdots = x_n^*$) and every x_i^* is optimal to (1);
- There exists an optimal pair (X^*, Y^*) with $Y^* = \sqrt{D-A}P$ for some $P \in \mathbb{R}^{n\times p}$ such that

$$\nabla f(X^*) + \sqrt{D-A}Y^* = 0,$$
$$\sqrt{D-A}X^* = 0. \tag{6}$$

Proof: First, by Null$(D-A) = e$, we have Null$(\sqrt{D-A}) = e$. Thus, $\sqrt{D-A}X^* = 0$ is equivalent to that X^* is consensual, namely, $x_1^* = x_2^* = \cdots = x_n^*$.

Second, given that X^* is consensual, every x_i^* is optimal to (1) if and only if $e^T\nabla f(X^*) = 0$. Because Null$(D-A) = e$, $e^T\nabla f(X^*) = 0$ is equivalent to that $\nabla f(X^*)$ stays in Span$(D-A)$. That is, there exists some $P \in \mathbb{R}^{n\times p}$ such that $\nabla f(X^*) + \sqrt{D-A}Y^* = 0$ with $Y^* = \sqrt{D-A}P$. This completes the proof. ■

Observing (5) and (6), we can see that the convergence of (X^k, Y^k) to (X^*, Y^*) is possible. Suppose that the convergence is true such that $\lim_{k\to\infty}X^k = X^*$ and $\lim_{k\to\infty}Y^k = Y^*$. Then in the first line of (5), $\lim_{k\to\infty}\nabla f(X^{k+1}) + \sqrt{D-A}Y^k = \nabla f(X^*) + \sqrt{D-A}Y^*$ and $\lim_{k\to\infty}2DX^{k+1} - (D+A)X^k = (D-A)X^*$; the latter term is zero because the null space of $D-A$ is e and X^* is consensual. In the second line of (5), $\lim_{k\to\infty}Y^{k+1} - Y^k - \sqrt{D-A}X^{k+1} = \sqrt{D-A}X^*$. Therefore, the optimal pair (X^*, Y^*) is a possible fixed point of the weighted ADMM iterate (X^k, Y^k).

In the convergence analyses, we are interested in the distance between (X^k, Y^k) and (X^*, Y^*). This is investigated through the weighted ADMM recursion in terms of X^k, Y^k, X^*, and Y^* given in Lemma 2. The recursion is obtained through subtracting the corresponding lines of (5) and (6).

Lemma 2 (Recursion of the weighted ADMM): Suppose $D - A \succeq 0$ and Null$(D-A) = e$. Under Assumptions 2 and 3, the quadruple sequence $\{X^k, Y^k, X^*, Y^*\}$ in the weighted ADMM obeys

$$\nabla f(X^{k+1}) - \nabla f(X^*) + \sqrt{D-A}(Y^{k+1} - Y^*) + (D+A)(X^{k+1} - X^k) = 0,$$
$$Y^{k+1} = Y^k + \sqrt{D-A}(X^{k+1} - X^*). \tag{7}$$

for any $k = 0, 1, \cdots$.

B. Convergence

To facilitate convergence analysis of the weighted ADMM, define

$$Z^k \triangleq \begin{pmatrix} Y^k \\ X^k \end{pmatrix}, \quad Z^* \triangleq \begin{pmatrix} Y^* \\ X^* \end{pmatrix}, \quad G \triangleq \begin{pmatrix} I & \mathbf{0} \\ \mathbf{0} & D+A \end{pmatrix}.$$

In Theorem 1, we shall show that Z^k converges to Z^*, or equivalently, Y^k converges to Y^* and X^k converges to X^*.

Theorem 1: Under Assumptions 2 and 3 and given that the weight matrices $D \in \mathcal{D}$ and $A \in \mathcal{A}$ are chosen such that $D + A \succeq 0$, $D - A \succeq 0$ and Null$(D-A) = e$, the iterate (X^k, Y^k) generated by the weighted ADMM converges to an optimal pair (X^*, Y^*).

Proof: See Appendix A. ■

Note that the weight matrices D and A that satisfy the requirements $D \in \mathcal{D}$, $A \in \mathcal{A}$, $D + A \succeq 0$, $D - A \succeq 0$ and Null$(D-A) = e$ do exist when Assumption 1 holds, namely, the network is connected. One possible choice is that $D = (U+V)/2$ and $A = (U-V)/2$ where U and V are the signless and signed Laplacian matrices of the network, respectively; see Section II-C.

C. Linear Convergence Rate

Suppose that the local objective functions have Lipschitz continuous gradients and are strongly convex as in Assumption 4. Recall that P and Q are diagonal matrices containing the squared Lipschitz gradient constants and the strong convexity constants, respectively. Theorem 2 further establishes linear rate of convergence for the weighted ADMM. In particular, we obtain the convergence speed that is explicitly determined by the Lipschitz continuous gradient and strong convexity constants of the local objective functions, as well as the spectral properties of the weight matrices D and A. Note that the optimal solution to (1) is unique due to the strong convexity of the objective function.

Theorem 2: Under Assumptions 2, 3 and 4 and given that the weight matrices $D \in \mathcal{D}$ and $A \in \mathcal{A}$ are chosen such that $D + A \succeq 0$, $D - A \succeq 0$ and Null$(D-A) = e$, the iterate (X^k, Y^k) generated by the weighted ADMM converges linearly to the optimal pair (X^*, Y^*). Specifically, for any positive δ satisfying

$$1 \geq \frac{\delta\theta\sigma_{\max}(D+A)}{\widetilde{\sigma}_{\min}(D-A)}, \tag{8}$$

and

$$2Q + (D-A) - \delta(D+A) \succeq \frac{\delta\theta}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}P, \tag{9}$$

where θ is a constant satisfying $\theta > 1$, it holds

$$\|Z^k - Z^*\|_G^2 \geq (1+\delta)\|Z^{k+1} - Z^*\|_G^2. \tag{10}$$

That is, $\|Z^k - Z^*\|_G^2$ converges to zero at the Q-linear rate $O\big((1+\delta)^{-k}\big)$ and consequently, $\|X^k - X^*\|_{D+A}^2$ converges to zero at the R-linear rate $O\big((1+\delta)^{-k}\big)$.

Proof: See Appendix B. ■

Theorem 2 shows that the weighted ADMM converges linearly and its theoretically achievable speed is given by the maximum constant δ satisfying (8) and (9), which is determined by the Lipschitz gradient and strong convexity constants of the local objective functions (P and Q) and the weight matrices (D and A). However, the matrix constraint (9) hinders us from obtaining an explicit expression of δ. In the corollary below, we give a sufficient condition of (8) and (9), which provides a clearer indication that how the weight matrices D and A affects the achievable convergence speed.

Corollary 1: Under Assumptions 2, 3 and 4 and given that the weight matrices $D \in \mathcal{D}$ and $A \in \mathcal{A}$ are chosen such that $D + A \succeq 0$, $D - A \succeq 0$ and Null$(D-A) = e$, the iterate (X^k, Y^k) generated by the weighted ADMM converges linearly to the optimal pair (X^*, Y^*) in the sense of

$$\left\|Z^k - Z^*\right\|_G^2 \geq (1+\delta)\left\|Z^{k+1} - Z^*\right\|_G^2.$$

The achievable convergence speed δ satisfies

$$\delta \leq \min\left\{\frac{\widetilde{\sigma}_{\min}(D-A)}{\theta\sigma_{\max}(D+A)}, \frac{2\mu}{\sigma_{\max}(D+A)+\frac{\theta L^2}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}}\right\}, \tag{11}$$

where θ is a constant satisfying $\theta > 1$, L and μ denote the largest Lipschitz continuity and the smallest strong convexity constants, respectively.

Proof: For (9), observe that the smallest eigenvalues of Q and $D-A$ are μ and 0, respectively, while the largest eigenvalues of $D+A$ and P are $\sigma_{\max}(D+A)$ and L^2, respectively. Therefore, a sufficient condition of (9) is

$$2\mu - \delta\sigma_{\max}(D+A) \geq \frac{\delta\theta L^2}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}. \tag{12}$$

It is easy to check that (8) and (12) simultaneously hold for any δ satisfying (11). ■

Note that when the matrices D and A are chosen such that the weighted ADMM turns to the conventional ADMM (see Section II-C), the theoretical bound δ also degenerates to the one of the conventional ADMM [23]. Also observe that (12) requires the smallest strong convexity constant μ is positive and sufficiently large, meaning that all local objective functions are strongly convex. This is actually not necessary since (9) only implies that $2Q+(D-A)$ must be sufficiently positive definite.

Based on the theoretical analyses in this section, the next section investigates how to minimize the communication cost of the weighted ADMM.

IV. Minimizing Communication Cost

This section investigates how to minimize the communication cost of the weighted ADMM through optimizing the spectral properties of the weight matrices D and A. Observe that the diagonal elements of D and A correspond to the nodes and the off-diagonal elements of A correspond to the arcs. If a_{ij} and a_{ji} are both zero and $i \neq j$, then nodes i and j have no information exchange even though there exist communication arcs between nodes i and j. Therefore, given a predefined network topology $(\mathcal{V}, \mathcal{E})$, we propose two different strategies of tuning D and A. The first strategy allows every a_{ij} to be nonzero as long as $i \in \mathcal{N}_j$ (Section IV-A). The second strategy lets some a_{ij} be zeros even though $i \in \mathcal{N}_j$, which is equivalent to selecting a subset of neighbors $\mathcal{C}_j$ from $\mathcal{N}_j$ to communicate and hence reduces the amount of information exchange per iteration (Section IV-B). Discussions on the reduction of communication cost are given in Section IV-C.

A. Maximizing Convergence Speed

According to (8) and (9) in Theorem 2, to maximize the convergence speed of the weighted ADMM through tuning the weight matrices D and A, the optimization model is

$$\begin{aligned} \max\ & \delta, \\ s.t.\ & 1 \geq \frac{\delta\theta\sigma_{\max}(D+A)}{\widetilde{\sigma}_{\min}(D-A)}, \\ & 2Q+(D-A)-\delta(D+A) \succeq \frac{\delta\theta}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}P, \\ & \theta > 1. \end{aligned} \tag{13}$$

The optimized convergence speed is dependent with the local Lipschitz gradient and strong convexity constants, which are contained in the matrices P and Q. If P and Q are known in advance and solving this optimization problem is affordable, we can obtain task-dependent weight matrices D and A. This straightforward application of Theorem 2, however, is not flexible when the local objective functions change (for example, the nodes collect new data for fusion). In addition, we may prefer setting the weight matrices prior to starting the decentralized network optimization tasks, when the properties of the local objective functions are unknown. To address these issues, we have an alternative approach that is simple and independent with P and Q.

Suppose that the local objective functions are unknown but fixed. Under this circumstance, the theoretically achievable speed given by (12) in Corollary 1 is monotonically decreasing in $\sigma_{\max}(D+A)$ while increasing in $\widetilde{\sigma}_{\min}(D-A)$. Hence, to accelerate the convergence speed and reduce the communication cost, we have the flexibility of tuning the weight matrices D and A so as to minimize $\sigma_{\max}(D+A)$ and maximize $\widetilde{\sigma}_{\min}(D-A)$. Note that tuning the elements in D and A changes the weights of the individual nodes and arcs. This helps because in a given topology some nodes and arcs may contribute more to the information diffusion process while the others contribute less. Intuitively, we expect to identify those important nodes and arcs and give them higher weights, which expedites propagation of "useful" information and in turn reduces exchange of "less useful" messages.

According to the discussions above, a simplified way of maximizing the convergence speed of the weighted ADMM through tuning the weight matrices D and A is to solve

$$\begin{aligned} \min_{D,A}\ & \left\{\sigma_{\max}(D+A), -\widetilde{\sigma}_{\min}(D-A)\right\}, \\ s.t.\ & D \in \mathcal{D}, A \in \mathcal{A}, \\ & D+A \succeq 0, D-A \succeq 0, \\ & \text{Null}(D-A) = e. \end{aligned} \tag{14}$$

However, the multi-objective optimization problem (14) is difficult to handle. Therefore, we propose a single-objective variant that keeps the value of $\sigma_{\max}(D+A)$ less than a positive constant ρ while minimizes $-\widetilde{\sigma}_{\min}(D-A)$. Thus, we have a

single-objective optimization formulation

$$\begin{aligned}\min_{D,A} \quad & -\widetilde{\sigma}_{\min}(D-A),\\ s.t. \quad & D\in\mathcal{D}, A\in\mathcal{A},\\ & D+A\succeq 0, D-A\succeq 0,\\ & \text{Null}(D-A)=e,\\ & \sigma_{\max}(D+A)\le\rho. \end{aligned} \tag{15}$$

The optimization problem (14) is convex since the objective function $-\widetilde{\sigma}_{\min}(D-A)$ is convex in (D,A) given that the smallest eigenvalue of $D-A$ is 0 with multiplicity 1, which is guaranteed by the constraint $\text{Null}(D-A)=e$ and $D-A\succeq 0$, while the set of constraints

$$\begin{aligned}\Omega\triangleq\{(D,A)|D\in\mathcal{D}, A\in\mathcal{A}, D+A\succeq 0, D-A\succeq 0,\\ \text{Null}(D-A)=e, \sigma_{\max}(D+A)\le\rho\}\end{aligned}$$

is also convex [36]. We solve (14) with CVX, a popular optimization toolbox [37]. Of particular note, in the implementation of CVX, we change the objective function of (15) to the negative summation of the two smallest eigenvalues of $D-A$ (namely, 0 and $\widetilde{\sigma}_{\min}(D-A)$), which is easier to handle in CVX. To reach an ϵ-optimal solution of (15), the number of iterations is in the order of $\mathcal{O}(\sqrt{n})\ln(1/\epsilon)$ and the per-iteration computation cost is in the order of $\mathcal{O}(n^4)$ [35].

Though the convergence speed derived in Theorem 2 is for the strongly convex case, our numerical experiments demonstrate that the idea of minimizing $\sigma_{\max}(D+A)$ and maximizing $\widetilde{\sigma}_{\min}(D-A)$ also works when the local objective functions are not strongly convex.

B. Maximizing Convergence Speed Using Limited Communication Arcs

As we have discussed in Sections II and III, overall communication cost of the weighted ADMM is determined by the product of the number of iterations and the communication cost per iteration. On a fixed topology, utilizing all the available communication arcs shall definitely achieve the fastest convergence speed, and hence reduce the number of iterations to reach a given accuracy. However, this strategy brings high communication cost per iteration for the unicast scheme. Indeed, some communication arcs are less important than the others and can be disconnected to reduce the amount of information exchange per iteration, while not significantly affecting the convergence speed as we have pointed out in Section IV-A. Therefore, below we propose another strategy that maximizes the convergence speed under the constraint of limited communication arcs.

Observe that the number of communication arcs required in the weighted ADMM equals to the number of nonzero off-diagonal elements in A. Denote $\text{OffDiag}(A)$ as a matrix whose off-diagonal elements are identical to those of A and diagonal elements are zeros. Also denote the pseudo ℓ_0 norm $\|\text{OffDiag}(A)\|_0$ as the number of nonzero elements of $\text{OffDiag}(A)$. Suppose that we expect to use at most $2s$ arcs (namely, at most s edges due to the symmetry of A), the optimization of D and A turns to

$$\begin{aligned}\min_{D,A} \quad & -\widetilde{\sigma}_{\min}(D-A),\\ s.t. \quad & (D,A)\in\Omega,\\ & \|\text{OffDiag}(A)\|_0\le 2s. \end{aligned} \tag{16}$$

Note that we can also consider a more complicated optimization task in a similar form of (13) by appending the constraint on the number of edges.

The new formulation (16) is nonconvex due to the ℓ_0 norm constraint. We propose to utilize ADMM to find a suboptimal solution of (16) because ADMM has had successful applications in many optimization problems with ℓ_0 norm constraints. Note that here ADMM is used to split the nonconvex constraint and the rest convex part, while in decentralized optimization ADMM is used to split the computation of the nodes. Following the ADMM routine, we introduce an auxiliary variable $\widetilde{A}\in\mathcal{R}^{n\times n}$ and reformulate (16) to

$$\begin{aligned}\min_{D,\widetilde{A},A} \quad & -\widetilde{\sigma}_{\min}(D-A),\\ s.t. \quad & (D,A)\in\Omega,\\ & \left\|\text{OffDiag}(\widetilde{A})\right\|_0\le 2s,\\ & \widetilde{A}=A. \end{aligned} \tag{17}$$

Denote $\Gamma\in\mathcal{R}^{n\times n}$ as the Lagrange multiplier corresponding to the constraint $\widetilde{A}=A$ and let a positive constant β be the ADMM penalty factor. The augmented Lagrangian of (17) is

$$-\widetilde{\sigma}_{\min}(D-A)+\left\langle\Gamma,\widetilde{A}-A\right\rangle+\frac{\beta}{2}\left\|\widetilde{A}-A\right\|_{\text{F}}^2, \tag{18}$$

where $(D,A)\in\Omega$ and $\|\text{OffDiag}(\widetilde{A})\|_0\le 2s$.

At every iteration, the ADMM first fixes $\widetilde{A}$ and Γ and minimizes the augmented Lagrangian with respect to (D,A), then fixes (D,A) and Γ and minimizes the augmented Lagrangian with respect to $\widetilde{A}$, and finally updates Γ from the calculated (D,A) and $\widetilde{A}$. At the t-th iteration, the update of (D,A) is

$$\begin{aligned}\left(D^{t+1},A^{t+1}\right)=\arg\min_{D,A} \quad & -\widetilde{\sigma}_{\min}(D-A)\\ & +\frac{\beta}{2}\left\|A-\widetilde{A}^t-\frac{\Gamma^t}{\beta}\right\|_{\text{F}}^2,\\ s.t. \quad & (D,A)\in\Omega. \end{aligned} \tag{19}$$

This is a convex program and can be solved by, for example, CVX. The update of $\widetilde{A}$ is

$$\begin{aligned}\widetilde{A}^{t+1}=\arg\min_{\widetilde{A}} \quad & \frac{\beta}{2}\left\|\widetilde{A}-A^{t+1}+\frac{\Gamma^t}{\beta}\right\|_{\text{F}}^2,\\ s.t. \quad & \left\|\text{OffDiag}(\widetilde{A})\right\|_0\le 2s. \end{aligned} \tag{20}$$

The explicit solution of (20) is the projection of $A^{t+1}-\Gamma^t/\beta$ onto the set $\{A|\|\text{OffDiag}(\widetilde{A})\|_0\le 2s\}$. This step is computationally cheap because we just need to keep the $2s$ largest-in-magnitude off-diagonal elements of $A^{t+1}-\Gamma^t/\beta$ and set the rest as zeros. For the sake of computational stability, we also

keep $\widetilde{A}^{t+1}$ symmetric in the optimization process. The update of Γ is

$$\Gamma^{t+1} = \Gamma^t + \beta\left(\widetilde{A}^{t+1} - A^{t+1}\right). \tag{21}$$

In preliminary experiments we tried to relax the nonconvex ℓ_0 norm constraint to the convex ℓ_1 norm constraint. However, numerical results on this convex approximation showed that the off-diagonal part of A is not as sparse as we expected. Thus, we resorted to the nonconvex formulation (16) and its ADMM algorithm, which yields favorable solutions as we will demonstrate in the next section. Indeed, the use of ADMM to handle ℓ_0 norm constraints has already found successful applications even though it cannot guarantee global convergence to the optimal solution; for this topic readers are referred to [34] and [38].

C. Discussions

In Section II we have assumed that the network follows either a broadcast or a unicast communication scheme. Suppose that at every iteration, every node exchanges its local estimate, whose communication cost is p, with a subset $\mathcal{C}_i$ of its neighbor set $\mathcal{N}_i$. For every iteration, it has been shown in Section II that the cost of broadcast is pn and the cost of unicast is $p\sum_{i=1}^n |\mathcal{C}_i|$.

When we only maximize the convergence speed of the weighted ADMM as in Section IV-A, very likely the weight matrix A is dense; that is, $a_{ij} \neq 0$ if and only if nodes i and j are neighbors or identical. In this case, $\mathcal{C}_i = \mathcal{N}_i$ for every node i. Therefore, saving of the communication cost is determined by the improvement of convergence speed.

If instead we maximize the convergence speed under the constraint of communication arcs as in Section IV-B, then the decentralized algorithm requires at most $2s$ arcs. In this case, $\sum_{i=1}^n |\mathcal{C}_i| \leq 2s$. Thus, we know that every iteration requires pn broadcast cost and at most $2ps$ unicast cost. Meanwhile, optimizing convergence speed also contributes to the reducing of the communication cost.

The optimization of D and A in this section can be done offline in a centralized manner. However, for a large-scale or slowly time-varying network, decentralized computation of D and A is preferred. Note that for the average consensus problem, which is a very special case of the consensus optimization problem (1), there have been relevant works about tuning weighted Laplacian matrices in average consensus algorithms to accelerate convergence speeds and reduce communication costs [39]. Decentralized optimization algorithms of calculating such weighted Laplacian matrices (for example, the one in [40]) can potentially enlighten the decentralized computation of D and A in our future research, since the matrix $D - A$ in this paper is essentially a weighted Laplacian matrix. Intuitively, the connection between our work and the previous ones on average consensus is understandable because they are all relevant to information diffusion over networks.

Two papers tightly related to our work are [41] and [42], which consider applying a weighted version of ADMM in the average consensus problem. Write the average consensus problem in the form of (1). For every node i, its local objective function has the simplest quadratic forms $f_i(x) = (1/2)\|y_i - x\|^2$, where y_i is a constant measurement vector to be averaged. Applying the weighted ADMM in this problem yields a linear system, whose convergence speed can be strictly characterized by the weight matrices and optimized based on the theoretical results. Note that the convergence analyses in [41] and [42] are essentially different from those in our paper, and cannot be easily extended to the general decentralized consensus optimization problem (1).

The optimization problems (15) and (16) have a common parameter ρ, while (16) has an additional parameter s. The value of ρ determines the scale of $\sigma_{\max}(D + A)$. Large ρ in general leads to large diagonal elements of D and A (namely, a_{ii} and d_{ii}). We suggest to use large ρ when the local cost functions are not strongly convex, since large values of d_{ii} improve the condition numbers of the objectives in (3). Otherwise, ρ can be chosen as a moderate value. The value of s represents the expected number of working communication edges. For the sake of reducing the unicast cost per iteration, s should be chosen as a small value. Nevertheless, s must be large enough such that the convergence speed (which determines the number of required iterations) is reasonable.

V. Numerical Experiments

In the numerical experiments, we compare the weighted ADMM and the conventional ADMM with respect to the communication costs. We first show that through maximizing the convergence speed, the weighted ADMM achieves better communication efficiency than the conventional one on unbalanced graphs (Section V-A). The saving of the weighted ADMM on the communication cost becomes more significant when we maximize the convergence speed under the constraint of communication arcs (Section V-B). In the comparison, we hand-tune the penalty factor c of the conventional ADMM and the parameter ρ (scale of $\sigma_{\max}(D + A)$) of the the weighted ADMM to their optimal values. We conduct numerical experiments on the following two local objective functions.

- *Quadratic functions*: for every node i, let
$$f_i(x) = \frac{1}{2}\|y_i - M_i x\|^2.$$
- *Huber functions*: for every node i, let
$$f_i(x) = h(y_i - M_i x).$$
Here $h(\cdot)$ is the Huber function. For a scalar a, $h(a)$ is given by $a^2/2$ if $|a| \leq 1$ and $|a| - (1/2)$ otherwise. It extends to the vector case as the sum of the Huber functions of the components [34]. In both functions, $M_i \in \mathcal{R}^{m\times p}$ and $y_i \in \mathcal{R}^m$ for every i and their elements are generated following the standard normal distribution.

In the numerical experiments, let $p = 3$ (length of x), $m = 3$ (length of y_i) and $n = 50$ (number of nodes). Quality of the local estimates at time k is evaluated by accuracy, which is defined as the maximum distance from the local estimates to the optimal solution x^*, namely, $\max_i \|x_i^k - x^*\|^2$. We demonstrate how the accuracies evolve with respect to the broadcast and unicast costs. As we have discussed in Section II-A, at every iteration the unicast costs are $p\sum_{i=1}^n |\mathcal{C}_i|$ for the weighted ADMM and $p\sum_{i=1}^n |\mathcal{N}_i|$ for the conventional ADMM, while the broadcast costs are pn for both. Therefore, the curves of broadcast costs also depict the convergence speeds of the two algorithms in terms of the number of iterations.

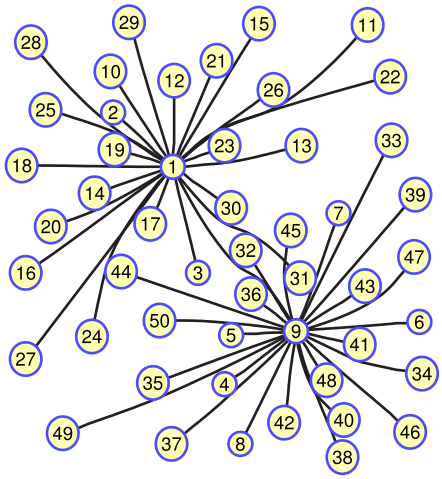

Fig. 1. A graph with two clusters of nodes. The weighted ADMM outperforms the conventional ADMM with respect to communication efficiency on this kind of graphs.

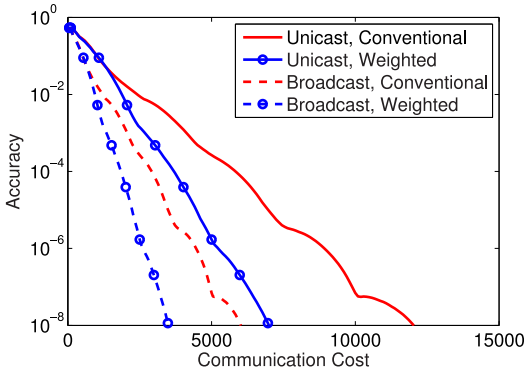

Fig. 2. Communication costs on the graph with two clusters. The local objective functions are quadratic.

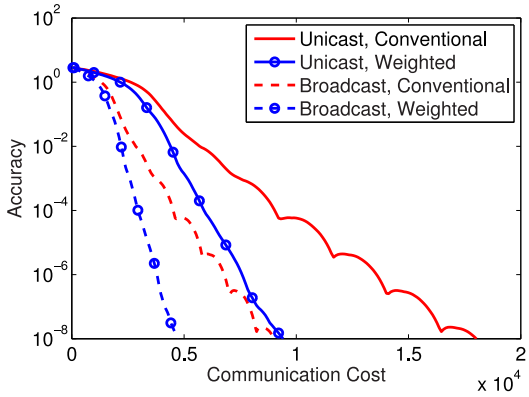

Fig. 3. Communication costs on the graph with two clusters. The local objective functions are Huber.

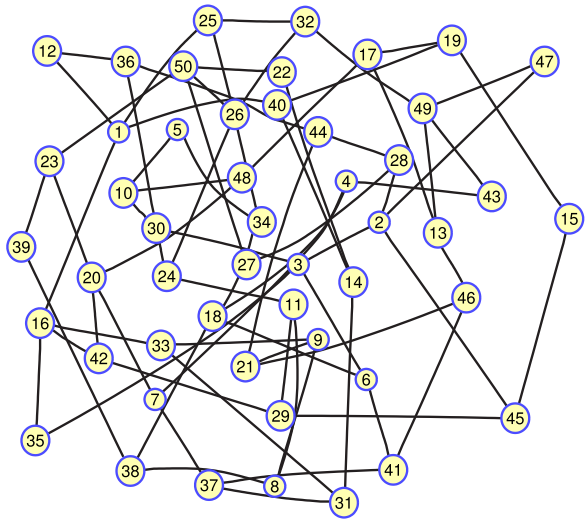

Fig. 4. The subgraph optimized through maximizing the convergence speed of the ADMM while the number of communication arcs is limited to $2s = 150$.

A. Best Convergence Speed

In the first set of numerical experiments, we consider maximizing the convergence speed of the weighted ADMM. Interestingly, the conventional ADMM performs almost the same as the weighted one on most regular (such as line, circle, star, and complete) graphs as well as on many random graphs according to extensive preliminary simulations. But we observe that if the graph has several clusters of nodes (see Fig. 1 for an example of two clusters), then the weighted ADMM outperforms its conventional counterpart. Figs. 2 and 3 compare the two algorithms for the cases that the local objective functions are quadratic and Huber, respectively. For both cases, the weighted ADMM only needs nearly a half number of iterations to reach an accuracy of 10^{-8}, comparing to the conventional ADMM. Because the two algorithms work on the same underlying graph, they require the same communication costs per iteration for both broadcast and unicast. Therefore, the weighted ADMM reduces 50% communication costs of both broadcast and unicast than the conventional ADMM, as demonstrated by Figs. 2 and 3.

Such a significant gap of communication efficiency is reasonable because in the conventional ADMM, the cluster heads do not distinguish the neighboring ordinary nodes and the neighboring cluster heads. Through optimizing the weight matrices, the weighted ADMM properly emphasizes the importance of the cluster heads to their neighboring peers, and hence achieves better communication efficiency.

B. Best Convergence Speed Using Limited Communication Arcs

In the second set of numerical experiments, we let the conventional ADMM run on a complete graph, but limit the number of communication arcs for the weighted ADMM. Optimally picking 150 arcs (letting $s = 75$) out of 2450 possible ones through solving (16), the resultant subgraph is given by Fig. 4. The communication costs of the two algorithms, in terms of broadcast and unicast, are demonstrated in Figs. 5 and 6 for quadratic and Huber local objective functions, respectively. Regarding the communication cost of broadcast, which is proportional to the number of iterations and irrelevant with the number of communication arcs, the conventional ADMM is nearly twice faster than the weighted ADMM when the target accuracy is set as 10^{-8}. The reason is that the conventional ADMM runs on a complete graph and guarantees to have the fastest information diffusion speed. However, considering the communication cost of unicast that is proportional to the product of the number of

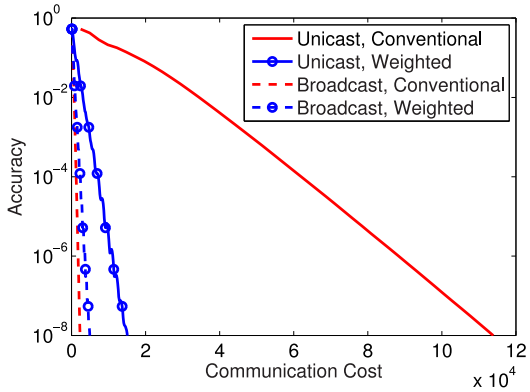

Fig. 5. Communication costs of the weighted ADMM on the optimized subgraph and the conventional one on the complete graph. The local objective functions are quadratic.

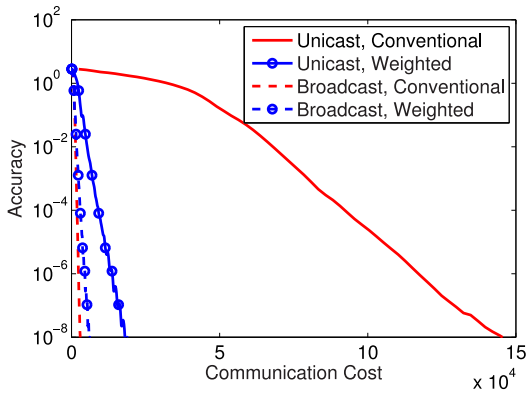

Fig. 6. Communication costs of the weighted ADMM on the optimized subgraph and the conventional one on the complete graph. The local objective functions are Huber.

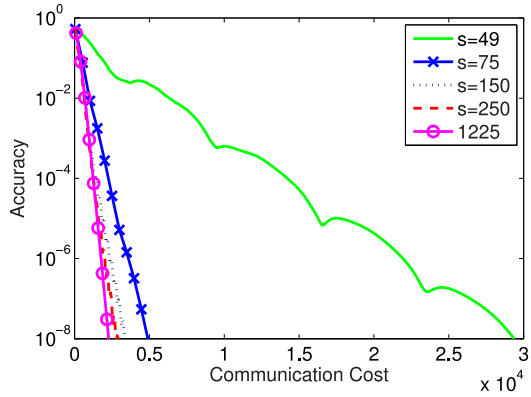

Fig. 7. The impact of the number of communication arcs $2s$ on the communication cost of broadcast in the weighted ADMM. The local objective functions are quadratic.

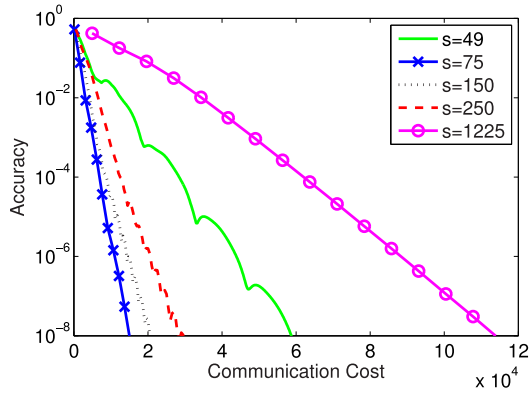

Fig. 8. The impact of the number of communication arcs $2s$ on the communication cost of unicast in the weighted ADMM. The local objective functions are quadratic.

iterations and the number of communication arcs, the weighted ADMM achieves more than 85% saving than the conventional ADMM, since the average number of neighbors reduces from 49 in the complete graph to 3 in the optimized subgraph. This demonstrates the importance of dropping out redundant communication arcs, which does not significantly slow down the convergence speed but effectively improves the communication efficiency.

A noticeable byproduct of the weighted ADMM is that the selected subgraph is naturally load-balanced, meaning that all the nodes have similar numbers of neighbors, though we do not explicitly consider this metric in (16). Most of the nodes have 3 neighbors and some have 2 or 4, which is beneficial to the robustness of the network. This is relevant with the properties of expander graphs [43], [44]. We shall investigate this phenomenon in our future research.

In the last set of numerical experiments, we demonstrate the impact of the number of communication arcs on the communication costs in the weighted ADMM. Letting every node hold a quadratic local objective function, we choose different values of $2s$ in (16). Figs. 7 and 8 show the communication costs of broadcast and unicast, respectively, and compare $s = 49$ (line graph), $s = 75$ (average node degree equals to 3), $s = 150$ (average node degree equals to 6), $s = 250$ (average node degree equals to 10), and $s = 1225$ (complete graph). Since adding the number of communication arcs generally helps information diffusion over the network, larger s leads to faster convergence, and thus smaller communication cost of broadcast (see Fig. 7). However, allowing too many communication arcs to work incurs large communication cost of unicast at every iteration. Hence, we can observe a tradeoff between the communication cost per iteration and the convergence speed (see Fig. 8). The rule of thumb is to choose a moderate number of the communication arcs, such that the convergence speed is reasonably fast, and meanwhile, the network is not unnecessarily dense.

VI. Conclusion

This paper is dedicated to improving the communication efficiency of ADMM, a powerful decentralized network optimization algorithm. We propose the weighted ADMM by assigning every communication arc and every node a weight, which determines the speed of network information diffusion. Compared with the conventional ADMM, the weighted ADMM is able to tune its weight matrices for the purpose of reducing the communication costs. We prove convergence and establish linear convergence rate of the weighted ADMM, and then maximize the derived convergence speed to obtain the best weight matrices on a given topology. Moreover, observing that exchanging information with all the neighbors is expensive, we maximize the

convergence speed while limit the number of communication arcs. This strategy finds a subset of arcs within the underlying topology to fulfill the optimization task and leads to a favorable tradeoff between the number of iterations and the communication cost per iteration. Numerical experiments show that the weighted ADMM outperforms its conventional counterpart in expediting the convergence speed and reducing the communication cost.

One of the future research topics is designing decentralized algorithms to optimize the weight matrices, which enables efficient implementation of the weighted ADMM. Another topic is investigating the load-balancing property of the weight tuning strategy. Through addressing these two issues, we expect to reach the goal of autonomous, robust and communication-efficient decentralized network optimization.

APPENDIX A
PROOF OF THEOREM 1

Proof: Given any bounded optimal pair (X^*, Y^*), by convexity of $f(X)$ in Assumption 2, it holds

$$\left\langle X^{k+1} - X^*, \nabla f\left(X^{k+1}\right) - \nabla f(X^*)\right\rangle \geq 0. \tag{22}$$

Substituting the equality containing $\nabla f(X^{k+1}) - \nabla f(X^*)$ in (7) into (22) yields

$$\left\langle X^{k+1} - X^*, -\sqrt{D-A}\left(Y^{k+1} - Y^*\right) \right. \\ \left. -(D+A)\left(X^{k+1} - X^k\right)\right\rangle \geq 0. \tag{23}$$

Rearranging the terms at the left-hand side of (23) and noticing $Y^{k+1} = Y^k + \sqrt{D-A}(X^{k+1} - X^*)$ in (7), we have

$$\left\langle Y^k - Y^{k+1}, Y^{k+1} - Y^*\right\rangle \\ + \left\langle (D+A)\left(X^k - X^{k+1}\right), X^{k+1} - X^*\right\rangle \geq 0. \tag{24}$$

According to the definition of Z^k, Z^* and G, we rewrite (24) to

$$\left\langle G\left(Z^k - Z^{k+1}\right), Z^{k+1} - Z^*\right\rangle \\ = \frac{1}{2}\left(\left\|Z^k - Z^*\right\|_G^2 - \left\|Z^{k+1} - Z^*\right\|_G^2 - \left\|Z^k - Z^{k+1}\right\|_G^2\right) \\ \geq 0. \tag{25}$$

Here the equality comes from expanding the squared norms.

Observing (25), we know that $\|Z^k - Z^*\|_G^2$ is nonincreasing. Because Z^0 and Z^* are bounded, it follows that $\|Z^0 - Z^*\|_G^2$ is bounded and consequently, any $\|Z^k - Z^*\|_G^2$ is bounded. Summing (25) over $k = 0$ through ∞ and applying the telescope cancellation, we have

$$\sum_{k=0}^{\infty}\left\|Z^k - Z^{k+1}\right\|_G^2 \leq \left\|Z^0 - Z^*\right\|_G^2 - \left\|Z^\infty - Z^*\right\|_G^2.$$

The summation is bounded because $\|Z^0 - Z^*\|_G^2$ and $\|Z^\infty - Z^*\|_G^2$ are both bounded. Thus we must have $\lim_{k\to\infty}\|Z^k - Z^{k+1}\|_G^2 = 0$.

According to the definitions of Z^k, Z^{k+1} and G, $\lim_{k\to\infty}\|Z^k - Z^{k+1}\|_G^2 = 0$ implies that $\lim_{k\to\infty}\|Y^k - Y^{k+1}\|_{\mathrm{F}}^2 = 0$ and $\lim_{k\to\infty}\|X^k - X^{k+1}\|_{D+A}^2 = 0$. The first result $\lim_{k\to\infty}\|Y^k - Y^{k+1}\|_{\mathrm{F}}^2 = 0$ immediately shows that Y^k converges to a stationary point $\hat{Y}$. Now we proceed to show that the second result $\lim_{k\to\infty}\|X^k - X^{k+1}\|_{D+A}^2 = 0$, along with the conditions $D + A \succeq 0$, $D - A \succeq 0$ and $\mathrm{Null}(D-A) = e$, guarantees that X^k also reaches a stationary point in limit.

Due to the convergence of Y^k to $\hat{Y}$ and by the second line of (5), we know that $\sqrt{D-A}X^{k+1}$ converges to zero. Since $\mathrm{Null}(D-A) = e$, X^{k+1} must stay in the subspace spanned by e in the limit. Hence, to satisfy $\lim_{k\to\infty}\|X^k - X^{k+1}\|_{D+A}^2 = 0$, we have either $(D+A)e = 0$ or $X^k - X^{k+1}$ converges to zero. But $(D+A)e \neq 0$ as can be proved by contradiction. If $(D+A)e = 0$, by $\mathrm{Null}(D-A) = e$ we also have $(D-A)e = 0$, then it holds $De = 0$. But by hypothesis $D \in \mathcal{D}$ is a diagonal matrix whose diagonal elements are positive such that $De \neq 0$. Thus we conclude that X^k converges to a stationary point, denoted by $\hat{X}$.

Given that (X^k, Y^k) converges to the stationary pair $(\hat{X}, \hat{Y})$, we shall show that $(\hat{X}, \hat{Y})$ is an optimal pair. Substituted into the weighted ADMM recursion (5), the stationary pair $(\hat{X}, \hat{Y})$ satisfies

$$\nabla f(\hat{X}) + \sqrt{D-A}\hat{Y} + (D-A)\hat{X} = 0, \tag{26}$$
$$\sqrt{D-A}\hat{X} = 0.$$

Comparing (26) with (6) in Lemma 1, we know $(\hat{X}, \hat{Y})$ satisfies the optimality condition and is thus an optimal pair, which completes the proof. ■

Note that in proving convergence of the weighted ADMM, we do not need f to be differentiable. If f is nondifferentiable but continuous, replacing the gradient ∇f by one of the subgradients ∂f, the proof still holds true. Here we assume that f is differentiable to keep the presentation simple. However, the assumption of differentiability is necessary in the proof of linear convergence.

APPENDIX B
PROOF OF THEOREM 2

Proof: By Assumption 4, we have

$$2\left\|X^* - X^{k+1}\right\|_Q^2 \\ \leq 2\left\langle X^* - X^{k+1}, \nabla f(X^*) - \nabla f(X^{k+1})\right\rangle. \tag{27}$$

Substituting (7) in Lemma 2 into the right-hand side of (27) yields

$$2\left\|X^* - X^{k+1}\right\|_Q^2 \\ \leq 2\left\langle X^* - X^{k+1}, \sqrt{D-A}\left(Y^{k+1} - Y^*\right)\right. \\ \left. + (D+A)\left(X^{k+1} - X^k\right)\right\rangle \\ = 2\left\langle Y^k - Y^{k+1}, Y^{k+1} - Y^*\right\rangle \\ + 2\left\langle X^* - X^{k+1}, (D+A)\left(X^{k+1} - X^k\right)\right\rangle. \tag{28}$$

Using the definitions of Z^k, Z^{k+1}, Z^* and G, we have

$$\begin{aligned} & 2\left\|X^* - X^{k+1}\right\|_Q^2 \\ \leq\ & 2\left\langle Z^k - Z^{k+1}, G\left(Z^{k+1} - Z^*\right)\right\rangle \\ =\ & \left\|Z^k - Z^*\right\|_G^2 - \left\|Z^{k+1} - Z^*\right\|_G^2 - \left\|Z^k - Z^{k+1}\right\|_G^2. \end{aligned} \quad (29)$$

A critical inequality: Having (29) at hand, in order to establish (10) it remains to show

$$\begin{aligned} & 2\left\|X^{k+1} - X^*\right\|_Q^2 + \left\|Z^k - Z^{k+1}\right\|_G^2 \\ \geq\ & \delta\left\|Z^{k+1} - Z^*\right\|_G^2. \end{aligned} \quad (30)$$

Observing the equality $Y^{k+1} = Y^k + \sqrt{D-A}(X^{k+1} - X^*)$ from (7) in Lemma 2, it holds that $\|Y^k - Y^{k+1}\|_{\mathrm{F}}^2 = \|X^{k+1} - X^*\|_{D-A}^2$. Hence (30) is equivalent to

$$\begin{aligned} & \left\|X^{k+1} - X^*\right\|_{2Q+(D-A)-\delta(D+A)}^2 + \left\|X^k - X^{k+1}\right\|_{D+A}^2 \\ & \geq \delta\left\|Y^{k+1} - Y^*\right\|_{\mathrm{F}}^2, \end{aligned} \quad (31)$$

which shall be proved below. The proof is based on finding upper bounds for $\|Y^{k+1} - Y^*\|_{\mathrm{F}}^2$ in terms of $\|X^{k+1} - X^*\|_{\mathrm{F}}^2$ and $\|X^k - X^{k+1}\|_{\mathrm{F}}^2$. We split it into the next two steps.

Establishing (31), step 1: From (7) in Lemma 2, we have

$$\begin{aligned} & \left\|\sqrt{D-A}\left(Y^{k+1} - Y^*\right)\right\|_{\mathrm{F}}^2 \\ =\ & \left\|(D+A)\left(X^{k+1} - X^k\right) + \nabla f\left(X^{k+1}\right) - \nabla f(X^*)\right\|_{\mathrm{F}}^2. \end{aligned} \quad (32)$$

From the inequality $\|C_1 + C_2\|_{\mathrm{F}}^2 \leq \theta\|C_1\|_{\mathrm{F}}^2 + \frac{\theta}{\theta-1}\|C_2\|_{\mathrm{F}}^2$, which holds for any $\theta > 1$ and any matrices C_1 and C_2 of the same dimensions, it follows that

$$\begin{aligned} \left\|Y^{k+1} - Y^*\right\|_{D-A}^2 \leq\ & \theta\left\|X^k - X^{k+1}\right\|_{(D+A)^2}^2 \\ & + \frac{\theta}{\theta-1}\left\|\nabla f\left(X^{k+1}\right) - \nabla f(X^*)\right\|_{\mathrm{F}}^2. \end{aligned} \quad (33)$$

By Lemmas 1 and 2, all the columns of Y^* and Y^{k+1} lie in the column space of $\sqrt{D-A}$. This fact, together with the Lipschitz continuity of ∇f in Assumption 4, turns (33) into

$$\begin{aligned} & \widetilde{\sigma}_{\min}(D-A)\left\|Y^{k+1} - Y^*\right\|_{\mathrm{F}}^2 \\ \leq\ & \theta\sigma_{\max}(D+A)\left\|X^k - X^{k+1}\right\|_{D+A}^2 \\ & + \frac{\theta}{\theta-1}\left\|X^{k+1} - X^*\right\|_P^2, \end{aligned} \quad (34)$$

where $\tilde{\sigma}_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ give the smallest *nonzero* and the largest eigenvalues, respectively.

Establishing (31), step 2: In order to establish (31), with (34), it only remains to show

$$\begin{aligned} & \left\|X^k - X^{k+1}\right\|_{D+A}^2 + \left\|X^{k+1} - X^*\right\|_{2Q+(D-A)-\delta(D+A)}^2 \\ \geq\ & \frac{\delta\theta\sigma_{\max}(D+A)}{\widetilde{\sigma}_{\min}(D-A)}\left\|X^k - X^{k+1}\right\|_{D+A}^2 \\ & + \frac{\delta\theta}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}\left\|X^{k+1} - X^*\right\|_P^2. \end{aligned} \quad (35)$$

A sufficient condition for (35) being valid is that

$$1 \geq \frac{\delta\theta\sigma_{\max}(D+A)}{\widetilde{\sigma}_{\min}(D-A)},$$

and

$$2Q + (D-A) - \delta(D+A) \succeq \frac{\delta\theta}{(\theta-1)\widetilde{\sigma}_{\min}(D-A)}P,$$

given that $D - A \succeq 0$. This finishes the proof of (31) and consequently, the proof of the main result (10). Observe that when $D + A \succeq 0$, $D - A \succeq 0$ and $\mathrm{Null}(D-A) = e$, (10) implies the linear convergence of (X^k, Y^k) to (X^*, Y^*); the derivation is the same as that in the proof of Theorem 1. ■

REFERENCES

[1] Q. Ling, Y. Liu, W. Shi, and Z. Tian, "Communication-efficient weighted ADMM for decentralized network optimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, Paper P15.4, pp. 4832-4825.

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

[3] G. Giannakis, Q. Ling, G. Mateos, I. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," 2015. [Online]. Available: http://arxiv.org/pdf/1503.08855v1.pdf

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

[5] J. Meng, W. Yin, H. Li, E. Hossain, and Z. Han, "Collaborative spectrum sensing from sparse observations in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 327–337, 2011.

[6] G. Giannakis, N. Gatsis, V. Kekatos, S. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power systems: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, pp. 107–128, Sep. 2013.

[7] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping," *IEEE J. Sel. Areas Commun.*, vol. 31, pp. 1184–1194, Jul. 2013.

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

[9] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 678–695, Aug. 2011.

[10] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1185–1197, Mar. 2014.

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

[12] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," 2013. [Online]. Available: http://arxiv.org/pdf/1310.7063v3.pdf

[13] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, pp. 1131–1146, May 2014.

[14] A. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, pp. 311–801, 2014.

[15] S. Tu and A. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, Mar. 2014.

[16] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.
[17] K. Tsianos and M. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. Amer. Control. Conf.*, 2012, pp. 1067–1072.
[18] A. Mokhtari, Q. Ling, and A. Ribeiro, "An approximate Newton method for distributed optimization," in *Proc. Int. Conf. Acoust, Speech Signal Process.*, 2015, pp. 2959–2963.
[19] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5158–5173, 2016.
[20] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1997.
[21] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.
[22] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," 2014. [Online]. Available: http://arxiv.org/pdf/1312.1085v3.pdf
[23] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
[24] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.
[25] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.
[26] T. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
[27] P. Bianchi, W. Hachem, and F. Iutzeler, "A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization," 2014. [Online]. Available: http://arxiv.org/pdf/1407.0898v2.pdf
[28] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, pp. 17–40, 1976.
[29] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, pp. 293–318, 1992.
[30] B. He and X. Yuan, "On the $O(1/t)$ convergence rate of the alternating dirction method," *SIAM J. Numer. Anal.*, vol. 50, pp. 700–709, 2012.
[31] W. Deng, M. Lai, and W. Yin, "On the $o(1/k)$ convergence and parallelization of the alternating direction method of multipliers," 2013. [Online]. Available: http://arxiv.org/pdf/1312.3040v2.pdf
[32] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, pp. 889–916, 2016.
[33] M. Hong and Z. Luo, "On the linear convergence of the alternating direction method of multipliers," 2012. [Online]. Available: http://arxiv.org/pdf/1208.3922v3.pdf
[34] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, 2011.
[35] K. Wang, A. So, T. Chang, W. Ma, and C. Chi, "Outrage constrained robust transmit optimization for multiuse MISO downlinks: Tractable approximations by conic optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5690–5705, Nov. 2014.
[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[37] M. Grant and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming, vers. 2.0 Beta. (2013) [Online]. Available: http://cvxr.com/cvx
[38] G. Li and T. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25. pp. 2434–2460, 2015.
[39] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, pp. 667–689, 2004.
[40] A. Bertrand and M. Moonen, "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks," *Signal Process.*, vol. 93, pp. 1106–1117, 2013.
[41] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5523–5537, Nov. 2011.
[42] E. Ghadimi, A. Teixeira, M. Rabbat, and M. Johansson, "The ADMM algorithm for distributed averaging: Convergence rates and optimal parameter selection," in *Proc. ASILOMAR Conf. Signals, Syst. Comput.*, 2014, pp. 783–787.
[43] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1996.
[44] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bull. Amer. Math. Soc.*, vol. 43, pp. 439–561, 2006.

Qing Ling received the B.E. degree in automation and the Ph.D. degree in control theory and control engineering both from the University of Science and Technology of China, in 2001 and 2006, respectively. From 2006 to 2009, he was a Postdoctoral Research Fellow in the Department of Electrical and Computer Engineering, Michigan Technological University. Since 2009, he has been an Associate Professor in the Department of Automation, University of Science and Technology of China, Hefei, China. His research interest includes decentralized optimization of networked multiagent systems.

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

Wei Shi received the B.E. degree in automation and the Ph.D. degree in control science and engineering both from the University of Science and Technology of China, Hefei, China, in 2010 and 2015, respectively. In 2015, he joined Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, as a Postdoctoral Research Associate. His research interests include optimization and its applications in signal processing and control, with a special focus on distributed optimization and power networks.

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