# On the Linear Convergence of the ADMM in Decentralized Consensus Optimization

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin

Abstract—In decentralized consensus optimization, a connected network of agents collaboratively minimize the sum of their local objective functions over a common decision variable, where their information exchange is restricted between the neighbors. To this end, one can first obtain a problem reformulation and then apply the alternating direction method of multipliers (ADMM). The method applies iterative computation at the individual agents and information exchange between the neighbors. This approach has been observed to converge quickly and deemed powerful. This paper establishes its linear convergence rate for the decentralized consensus optimization problem with strongly convex local objective functions. The theoretical convergence rate is explicitly given in terms of the network topology, the properties of local objective functions, and the algorithm parameter. This result is not only a performance guarantee but also a guideline toward accelerating the ADMM convergence.

*Index Terms*—Decentralized consensus optimization, alternating direction method of multipliers (ADMM), linear convergence.

# I. INTRODUCTION

**R** ECENT advances in signal processing and control of networked multi-agent systems have led to much research interests in decentralized optimization [2]–[14]. Decentralized optimization problems arising in networked multi-agent systems include coordination of aircraft or vehicle networks [2]–[4], data processing of wireless sensor networks [5]–[10], spectrum sensing of cognitive radio networks [11], [12], state estimation and operation optimization of smart grids [13], [14], etc. In these scenarios, the data is collected and/or stored in a distributed manner; a fusion center is either disallowed or not economical. Consequently, any computing tasks must be

Manuscript received July 20, 2013; revised November 30, 2013; accepted January 20, 2014. Date of publication February 04, 2014; date of current version March 10, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuguang Cui. The work of W. Shi is supported by Chinese Scholarship Council grant 201306340046. The work of Q. Ling is supported by NSFC grant 61004137 and Chinese Scholarship Council grant 2011634506. The work of G. Wu is supported by MOF/MIIT/MOST grant BB2100100015. The work of W. Yin is supported by ARL and ARO grant W911NF-09-1-0383 and NSF grants DMS-0748839 and DMS-1317602. Part of this paper appeared in the Thiry-Eighth International Conference on Acoustics, Speech, and Signal Processing, Vancouver, Canada, May26–31, 2013. (Corresponding author: Q. Ling).

W. Shi, Q. Ling, K. Yuan, and G. Wu are with the Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China (e-mail: qingling@mail.ustc.edu.cn).

W. Yin is with the Department of Mathematics, University of California, Los Angeles, CA 90095 USA.

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.2014.2304432

accomplished in a decentralized and collaborative manner by the agents. This approach can be powerful and efficient, as the computing tasks are distributed over all the agents and information exchange occurs only between the agents with direct communication links. There is no risk of central computation overload or network congestion.

In this paper, we focus on *decentralized consensus optimization*, an important class of decentralized optimization in which a network of L agents cooperatively solve

$$\min_{\tilde{x}} \quad \sum_{i=1}^{L} f_i(\tilde{x}), \tag{1}$$

over a common optimization variable  $\tilde{x}$ , where  $f_i(\tilde{x}) : \mathbb{R}^N \to \mathbb{R}$ is the local objective function known by agent *i*. This formulation arises in averaging [4]–[6], learning [7], [8], and estimation [9]–[13] problems. Examples of  $f_i(\tilde{x})$  include least squares [4]–[6], regularized least squares [8], [10]–[12], as well as more general ones [7]. The values of  $\tilde{x}$  can stand for average temperature of a room [5], [6], frequency-domain occupancy of spectra [11], [12], states of a smart grid system [13], [14], and so on.

There exist several methods for decentralized consensus optimization, including distributed subgradient descent algorithms [15]–[17], dual averaging methods [18], [19], and the alternating direction method of multipliers (ADMM) [8]–[10], [20], [21]. Among these algorithms, the ADMM demonstrates fast convergence in many applications, e.g., [8]–[10]. However, how fast it converges and what factors affect the rate are both unknown. This paper addresses these issues.

#### A. Our Contributions

Firstly, we establish the linear convergence rate of the ADMM that is applied to decentralized consensus optimization with strongly convex local objective functions. This theoretical result gives a performance guarantee for the ADMM and validates the observation in prior literature.

Secondly, we study how the network topology, the properties of local objective functions, and the algorithm parameter affect the convergence rate. The analysis provide guidelines for networking strategies, objective-function splitting strategies, and algorithm parameter settings to achieve faster convergence.

# B. Related Work

Besides the ADMM, existing decentralized approaches for solving (1) include belief propagation [7], incremental optimization [22], subgradient descent [15]–[17], dual averaging [18], [19], etc. Belief propagation and incremental optimization require one to predefine a tree or loop structure in the network,

1053-587X © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information. whereas the advantage of the ADMM, subgradient descent, and dual averaging is that they do not rely on any predefined structures. Subgradient descent and dual averaging work well for asynchronous networks but suffer from slow convergence. Indeed, for subgradient descent algorithms [15] and [16] establish the convergence rate of O(1/k), where k is the number of iterations, to a neighborhood of the optimal solution when the local subgradients are bounded and the stepsize is fixed. Further assuming that the local objective functions are strongly convex, choosing a dynamic stepsize leads to a rate of  $O(\log(k)/k)$ [17]. Dual averaging methods using dynamic stepsizes also have sublinear rates, e.g.,  $O(\log(k)/\sqrt{k})$  as proved in [18] and [19].

The decentralized ADMM approaches use synchronous steps by all the agents but have much faster empirical convergence, as demonstrated in many applications [8]–[10]. However, existing convergence rate analysis of the ADMM is restricted to the classic, centralized computation. The centralized ADMM has a sublinear convergence rate O(1/k) for general convex optimization problems [23]. In [24] an ADMM with restricted stepsizes is proposed and proved to be linearly convergent for certain types of non-strongly convex objective functions. A recent paper [25] shows a linear convergence rate  $O(1/a^k)$  for some a > 1 under a strongly convex assumption, and our paper extends the analysis tools therein to the decentralized regime.

A notable work about convergence rate analysis is [20], which proves the linear convergence rate of the ADMM applied to the average consensus problem, a special case of (1) in which  $f_i(\tilde{x}) = ||\tilde{x} - y_i||_2^2$  with  $y_i$  being a local measurement vector of agent *i*. Its analysis takes a state-transition equation approach, which is not applicable to the more general local objective functions considered in this paper.

## C. Paper Organization and Notation

This paper is organized as follows. Section II reformulates the decentralized consensus optimization problem and develops an algorithm based on the ADMM. Section III analyzes the linear convergence rate of the ADMM and shows how to accelerate the convergence through tuning the algorithm parameter. Section IV provides extensive numerical experiments to validate the theoretical analysis in Section III. Section V concludes the paper.

In this paper we denote  $||x||_2$  as the Euclidean norm of a vector x and  $\langle x, y \rangle$  as the inner product of two vectors x and y. Given a semidefinite matrix G with proper dimensions, the G-norm of x is  $\sqrt{x^T G x}$ . We let  $\sigma_{\max}(G)$  be the operator that returns the largest singular value of G and  $\tilde{\sigma}_{\min}(G)$  be the one that returns the smallest nonzero singular value of G.

We use two kinds of definitions of convergence, Q-linear convergence and R-linear convergence. We say that a sequence  $y^k$ , where the superscript k stands for time index, Q-linearly converges to a point  $y^*$  if there exists a number  $\rho \in (0, 1)$  such that  $\lim_{k \to \infty} \frac{\|y^{k+1} - y^*\|}{\|y^k - y^*\|} = \rho$  with  $\|\cdot\|$  being a vector norm. We say that a sequence  $x^k$  R-linearly converges to a point  $x^*$  if for all k,  $\|x^k - x^*\| \le \|y^k - y^*\|$  where  $y^k$  Q-linearly converges to  $y^*$ .

# II. THE ADMM FOR DECENTRALIZED CONSENSUS OPTIMIZATION

In this section, we first reformulate the decentralized consensus optimization problem (1) such that it can be solved by the ADMM (see Section II-A). Then we develop the decentralized ADMM approach and provide a simplified decentralized algorithm (see Section II-B).

# A. Problem Formulation

Throughout the paper, we consider a network consisting of L agents bidirectionally connected by E edges (and thus 2E arcs). We can describe the network as a symmetric directed graph  $\mathcal{G}_{d} = \{\mathcal{V}, \mathcal{A}\}$  or an undirected graph  $\mathcal{G}_{u} = \{\mathcal{V}, \mathcal{E}\}$ , where  $\mathcal{V}$  is the set of vertexes with cardinality  $|\mathcal{V}| = L$ ,  $\mathcal{A}$  is the set of arcs with  $|\mathcal{A}| = 2E$ , and  $\mathcal{E}$  is the set of edges with  $|\mathcal{E}| = E$ . Algorithms that solve the decentralized consensus optimization problem (1) are developed based on this graph.

Generally speaking, the ADMM applies to the convex optimization problem in the form of

$$\min_{y_1, y_2} \quad g_1(y_1) + g_2(y_2), \\ \text{s.t.} \quad C_1 y_1 + C_2 y_2 = b,$$
 (2)

where  $y_1$  and  $y_2$  are optimization variables,  $g_1$  and  $g_2$  are convex functions, and  $C_1y_1 + C_2y_2 = b$  is a linear constraint of  $y_1$  and  $y_2$ . The ADMM solves a sequence of subproblems involving  $g_1$ and  $g_2$  one at a time and iterates to converge as long as a saddle point exists.

To solve (1) with the ADMM in a decentralized manner, we reformulate it as

$$\min_{\{x_i\},\{z_{ij}\}} \sum_{i=1}^{L} f_i(x_i),$$
  
s.t.  $x_i = z_{ij}, x_j = z_{ij}, \forall (i,j) \in \mathcal{A}.$  (3)

Here  $x_i$  is the local copy of the common optimization variable  $\tilde{x}$  at agent *i* and  $z_{ij}$  is an auxiliary variable imposing the consensus constraint on neighboring agents *i* and *j*. In the constraints,  $\{x_i\}$  are separable when  $\{z_{ij}\}$  are fixed, and vice versa. Therefore, (3) lends itself to decentralized computation in the ADMM framework. Apparently, (3) is equivalent to (1) when the network is connected.

Defining  $x \in \mathbb{R}^{LN}$  as a vector concatenating all  $x_i, z \in \mathbb{R}^{2EN}$  as a vector concatenating all  $z_{ij}$ , and  $f(x) = \sum_{i=1}^{L} f_i(x_i)$ , (3) can be written in a matrix form as

$$\min_{\substack{x,z\\ \text{s.t.}}} f(x) + g(z),$$
s.t.  $Ax + Bz = 0,$  (4)

where g(z) = 0, which fits the form of (2), and is amenable to the ADMM. Here  $A = [A_1; A_2]; A_1, A_2 \in \mathbb{R}^{2EN \times LN}$  are both composed of  $2E \times L$  blocks of  $N \times N$  matrices. If  $(i, j) \in A$ and  $z_{ij}$  is the *q*th block of *z*, then the (q, i)th block of  $A_1$  and the (q, j)th block of  $A_2$  are  $N \times N$  identity matrices  $I_N$ ; otherwise the corresponding blocks are  $N \times N$  zero matrices  $0_N$ . Also, we have  $B = [-I_{2EN}; -I_{2EN}]$  with  $I_{2EN}$  being a  $2EN \times 2EN$ identity matrix.

# B. Algorithm Development

Now we apply the ADMM to solve (4). The augmented Lagrangian of (4) is

$$L_c(x, z, \lambda) = f(x) + \langle \lambda, Ax + Bz \rangle + \frac{c}{2} ||Ax + Bz||_2^2,$$

where  $\lambda \in \mathbb{R}^{4EN}$  is the Lagrange multiplier and c is a positive algorithm parameter. At iteration k + 1, the ADMM firstly minimizes  $L_c(x, z^k, \lambda^k)$  to obtain  $x^{k+1}$ , secondly minimizes  $L_c(x^{k+1}, z, \lambda^k)$  to obtain  $z^{k+1}$ , and finally updates  $\lambda^{k+1}$  from  $x^{k+1}$  and  $z^{k+1}$ . The updates are

$$\begin{aligned} x \text{-update} : & \nabla f(x^{k+1}) + A^T \lambda^k \\ & + c A^T (A x^{k+1} + B z^k) &= 0, \\ z \text{-update} : & B^T \lambda^k + c B^T (A x^{k+1} + B z^{k+1}) &= 0, \\ \lambda \text{-update} : & \lambda^{k+1} - \lambda^k - c (A x^{k+1} + B z^{k+1}) &= 0, \end{aligned}$$
(5)

where  $\nabla f(x^{k+1})$  is the gradient of f(x) at point  $x = x^{k+1}$  if f is differentiable, or is a subgradient if f is non-differentiable.

Next we show that if the initial values of z and  $\lambda$  are properly chosen the ADMM updates in (5) can be simplified (see also the derivation in [8]). Multiplying the two sides of the  $\lambda$ -update by  $A^T$  and adding it to the x-update, we have  $\nabla f(x^{k+1}) + A^T \lambda^{k+1} + cA^T B(z^k - z^{k+1}) = 0$ . Further, multiplying the two sides of the  $\lambda$ -update by  $B^T$  and adding it to the z-update we have  $B^T \lambda^{k+1} = 0$ . Therefore (5) can be equivalently expressed as

$$\nabla f(x^{k+1}) + A^T \lambda^{k+1} + c A^T B(z^k - z^{k+1}) = 0,$$
  

$$B^T \lambda^{k+1} = 0,$$
  

$$\lambda^{k+1} - \lambda^k - c(Ax^{k+1} + Bz^{k+1}) = 0.$$
 (6)

Letting  $\lambda = [\beta; \gamma]$  with  $\beta, \gamma \in \mathbb{R}^{2EN}$  and recalling  $B = [-I_{2EN}; -I_{2EN}]$ , we know  $\beta^{k+1} = -\gamma^{k+1}$  from the second equation of (6). Therefore, the first equation in (6) reduces to  $\nabla f(x^{k+1}) + M_{-}\beta^{k+1} - cM_{+}(z^{k} - z^{k+1}) = 0$  where  $M_{+} = A_{1}^{T} + A_{2}^{T}$  and  $M_{-} = A_{1}^{T} - A_{2}^{T}$ . The third equation in (6) splits to two equations  $\beta^{k+1} - \beta^{k} - cA_{1}x^{k+1} + cz^{k+1} = 0$  and  $\gamma^{k+1} - \gamma^{k} - cA_{2}x^{k+1} + cz^{k+1} = 0$ . If we choose the initial value of  $\lambda$  as  $\beta^{0} = -\gamma^{0}$  such that  $\beta^{k} = -\gamma^{k}$  holds for  $k = 0, 1, \cdots$ , summing and subtracting these two equations result in  $\frac{1}{2}M_{+}^{T}x^{k+1} - z^{k+1} = 0$  and  $\beta^{k+1} - \beta^{k} - \frac{c}{2}M_{-}^{T}x^{k+1} = 0$ , respectively. If we further choose the initial value of z as  $z^{0} = \frac{1}{2}M_{+}^{T}x^{0}$ ,  $\frac{1}{2}M_{+}^{T}x^{k} - z^{k} = 0$  holds for  $k = 0, 1, \cdots$ .

To summarize, with initialization  $\beta^0 = -\gamma^0$  and  $z^0 = \frac{1}{2}M_+^T x^0$ , (6) reduces to

$$\nabla f(x^{k+1}) + M_{-}\beta^{k+1} - cM_{+}(z^{k} - z^{k+1}) = 0,$$
  
$$\beta^{k+1} - \beta^{k} - \frac{c}{2}M_{-}^{T}x^{k+1} = 0,$$
  
$$\frac{1}{2}M_{+}^{T}x^{k} - z^{k} = 0.$$
(7)

In Section III we will analyze the convergence rate of the ADMM updates (7). The analysis requires an extra initialization condition that  $\beta^0$  lies in the column space of  $M_-^T$  (e.g.,  $\beta^0 = 0$ ) such that  $\beta^{k+1}$  also lies in the column space of  $M_-^T$ ; the reason will be given in Section III.

Indeed, (7) also leads to a simple decentralized algorithm that involves only an x-update and a new multiplier update. To see this, substituting  $\frac{1}{2}M_+^T x^k - z^k = 0$  into the first two equations of (7) we have

$$\nabla f(x^{k+1}) + M_{-}\beta^{k+1} - \frac{c}{2}M_{+}M_{+}^{T}x^{k} + \frac{c}{2}M_{+}M_{+}^{T}x^{k+1} = 0, \qquad (8)$$
$$\beta^{k+1} - \beta^{k} - \frac{c}{2}M_{-}^{T}x^{k+1} = 0,$$

which is irrelevant with z. Note that in the first equation of (8) the x-update relies on  $M_{-}\beta^{k+1}$  other than  $\beta^{k}$ . Therefore, multiplying the second equation with  $M_{-}$  we have  $M_{-}\beta^{k+1} - M_{-}\beta^{k} - \frac{c}{2}M_{-}M_{-}^{T}x^{k+1} = 0$ . Substituting it to the first equation of (8) we obtain the x-update where  $x^{k+1}$  is decided by  $x^{k}$  and  $M_{-}\beta^{k}$ , i.e.,  $\nabla f(x^{k+1}) + M_{-}\beta^{k} + (\frac{c}{2}M_{+}M_{+}^{T} + \frac{c}{2}M_{-}M_{-}^{T})x^{k+1} - \frac{c}{2}M_{+}M_{+}^{T}x^{k} = 0$ . Letting  $W \in \mathbb{R}^{LN \times LN}$  be a block diagonal matrix with its (i, i)th block being the degree of agent i multiplying  $I_{N}$  and other blocks being  $0_{N}$ ,  $L_{+} = \frac{1}{2}M_{+}M_{+}^{T}$ ,  $L_{-} = \frac{1}{2}M_{-}M_{-}^{T}$ , we know  $W = \frac{1}{2}(L_{+} + L_{-})$ . Defining a new multiplier  $\alpha = M_{-}\beta \in \mathbb{R}^{LN}$ , we obtain a simplified decentralized algorithm

x-update: 
$$\nabla f(x^{k+1}) + \alpha^k + 2cWx^{k+1} - cL_+x^k = 0,$$
 (9)  
 $\alpha$ -update:  $\alpha^{k+1} - \alpha^k - cL_-x^{k+1} = 0.$ 

The introduced matrices  $M_+$ ,  $M_-$ ,  $L_+$ ,  $L_-$ , and W are related to the underlying network topology. With regard to the undirected graph  $\mathcal{G}_u$ ,  $M_+$  and  $M_-$  are the extended unoriented and oriented incidence matrices, respectively;  $L_+$  and  $L_-$  are the extended signless and signed Laplacian matrices, respectively; and W is the extended degree matrix. By "extended", we mean replacing every 1 by  $I_N$ , -1 by  $-I_N$ , and 0 by  $0_N$  in the original definitions of these matrices [26]–[29].

The updates in (9) are distributed to agents. Note that  $x = [x_1; \dots; x_L]$  where  $x_i$  is the local solution of agent i and  $\alpha = [\alpha_1; \dots; \alpha_L]$  where  $\alpha_i \in \mathbb{R}^N$  is the local Lagrange multiplier of agent i. Recalling the definitions of W,  $L_+$  and  $L_-$ , (9) translates the update of agent i by

$$x_{i}^{k+1} = (\nabla f_{i} + 2c|\mathcal{N}_{i}|I)^{-1} \left( c|\mathcal{N}_{i}|x_{i}^{k} + c\sum_{j\in\mathcal{N}_{i}} x_{j}^{k} - \alpha_{i}^{k} \right),$$
  
$$\alpha_{i}^{k+1} = \alpha_{i}^{k} + c \left( |\mathcal{N}_{i}|x_{i}^{k+1} - \sum_{j\in\mathcal{N}_{i}} x_{j}^{k+1} \right),$$
(10)

where  $N_i$  denotes the set of neighbors of agent *i*. The algorithm is fully decentralized since the updates of  $x_i$  and  $\alpha_i$  only rely on local and neighboring information. The decentralized consensus optimization algorithm based on the ADMM is outlined in Table I.

#### III. CONVERGENCE RATE ANALYSIS

This section first establishes the linear convergence rate of the ADMM in decentralized consensus optimization with strongly convex local objective functions (see Section III-A); the detailed proof of the main theoretical result is placed in Appendix. We then discuss how to tune the parameter and accelerate the convergence (see Section III-B).

 TABLE I

 Algorithm 1: Decentralized Consensus Optimization Based on the ADMM

Input functions 
$$f_i$$
; initialize variables  $x_i^0 = 0$ ,  $\alpha_i^0 = 0$ ; set algorithm parameter  $c > 0$ ;  
For  $k = 0, 1, \cdots$ , every agent *i* do  
Update  $x_i^{k+1}$  by solving  $\nabla f_i(x_i^{k+1}) + \alpha_i^k + 2c|\mathcal{N}_i|x_i^{k+1} - c\left(|\mathcal{N}_i|x_i^k + \sum_{j \in \mathcal{N}_i} x_j^k\right) = 0$ ;  
Update  $\alpha_i^{k+1} = \alpha_i^k + c\left(|\mathcal{N}_i|x_i^{k+1} - \sum_{j \in \mathcal{N}_i} x_j^{k+1}\right)$ ;  
End for

# A. Main Theoretical Result

Throughout this paper, we make the following assumption that the local objective functions are strongly convex and have Lipschitz continuous gradients; note that the latter implies differentiability.

Assumption 1: The local objective functions are strongly convex. For each agent *i* and given any  $\tilde{x}_a, \tilde{x}_b \in \mathbb{R}^N$  $\langle \nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b), \tilde{x}_a - \tilde{x}_b \rangle \geq m_{f_i} || \tilde{x}_a - \tilde{x}_b ||_2^2$  with  $m_{f_i} > 0$ . The gradients of the local objective functions are Lipschitz continuous. For each agent *i* and given any  $\tilde{x}_a, \tilde{x}_b \in \mathbb{R}^N$ ,  $|| \nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b) ||_2 \leq M_{f_i} || \tilde{x}_a - \tilde{x}_b ||_2$  with  $M_{f_i} > 0$ .

$$\begin{split} \|\nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b)\|_2 &\leq M_{f_i} \|\tilde{x}_a - \tilde{x}_b\|_2 \text{ with } M_{f_i} > 0. \\ \text{Recall the definition } f(x) &= \sum_{i=1}^L f_i(x_i). \text{ Assumption 1 directly indicates that } f(x) \text{ is strongly convex (i.e., } \langle \nabla f(x_a) - \nabla f(x_b), x_a - x_b \rangle \geq m_f \|x_a - x_b\|_2^2 \text{ given any } x_a, x_b \in \mathbb{R}^{LN} \\ \text{with } m_f &= \min_i m_{f_i} \text{ ) and the gradient of } f(x) \text{ is Lipschitz continuous (i.e., } \|\nabla f(x_a) - \nabla f(x_b)\|_2 \leq M_f \|x_a - x_b\|_2 \text{ for any } \\ x_a, x_b \in \mathbb{R}^{LN} \text{ with } M_f = \max_i M_{f_i} \text{).} \end{split}$$

Although the convergence of Algorithm 1 to the optimal solution of (4) can be shown based on the convergence property of the ADMM (see e.g., [21]), establishing its linear convergence is nontrivial. In [25] the linear convergence of the centralized ADMM is proved given that either g(z) is strongly convex or *B* is full row-rank in (8). However, the decentralized consensus optimization problem does not satisfy these conditions. The function g(z) = 0 is not strongly convex, and the matrix  $B = [-I_{2EN}; -I_{2EN}]$  is row-rank deficient.

Next we will analyze the convergence rate of the ADMM iteration (7). The analysis requires an extra initialization condition that  $\beta^0$  lies in the column space of  $M_-^T$  such that  $\beta^{k+1}$  also lies in the column space of  $M_-^T$ , which is necessary in the analysis. Note that there is a unique optimal multiplier  $\beta^*$  lying in the column space of  $M_-^T$ . To see so, taking  $k \to +\infty$  in (7) yields the KKT conditions of (4)

$$\nabla f(x^*) + M_{-}\beta^* = 0,$$
  

$$M_{-}^T x^* = 0,$$
  

$$\frac{1}{2}M_{+}^T x^* - z^* = 0,$$
(11)

where  $(x^*, z^*)$  is the unique primal optimal solution and the uniqueness follows from the strong convexity of f(x) as well as the consensus constraint Ax + Bz = 0. Since the consensus constraints Ax + Bz = 0 are feasible, there is at least one optimal multiplier  $\tilde{\beta}$  exists such that  $\nabla f(x^*) + M_{-}\tilde{\beta} = 0$ . We show that its projection onto the column space of  $M_{-}^T$ , denoted by  $\beta^*$ , is also an optimal multiplier. According to the property of projection,  $M_{-}(\tilde{\beta} - \beta^*) = 0$  and hence  $M_{-}\tilde{\beta} = M_{-}\beta^*$ . Therefore, the projection  $\beta^*$  that lies in the column space of  $M_{-}^{T}$  also satisfies  $\nabla f(x^{*}) + M_{-}\beta^{*} = 0$ . Next we show the uniqueness of such a  $\beta^{*}$  by contradiction. Consider two different vectors  $M_{-}^{T}v_{1}, M_{-}^{T}v_{2} \in \mathbb{R}^{2EN}$  that both lie in the column space of  $M_{-}^{T}$  and satisfy the equation. Therefore, we have  $\nabla f(x^{*}) + M_{-}M_{-}^{T}v_{1} = 0$  and  $\nabla f(x^{*}) + M_{-}M_{-}^{T}v_{2} = 0$ . Subtracting them yields  $M_{-}M_{-}^{T}(v_{1} - v_{2}) = 0$ . Since  $\|M_{-}M_{-}^{T}(v_{1} - v_{2})\|_{2} \geq \tilde{\sigma}_{\min}(M_{-})\|M_{-}^{T}(v_{1} - v_{2})\|_{2}$  where  $\tilde{\sigma}_{\min}(M_{-})$  is the smallest nonzero singular value of  $M_{-}$ , we conclude that  $\|M_{-}^{T}(v_{1} - v_{2})\|_{2} = 0$  and consequently  $M_{-}^{T}v_{1} = M_{-}^{T}v_{2}$  which contradicts with the assumption of  $M_{-}^{T}v_{1}$  and  $M_{-}^{T}v_{2}$  being different. Hence,  $\beta^{*}$  is the unique dual optimal solution that lies in the column space of  $M_{-}^{T}$ .

Our main theoretical result considers the convergence of a vector that concatenating the primal variable z and the dual variable  $\beta$ , which is common in the convergence rate analysis of the ADMM [23]–[25]. Let us introduce

$$u = \begin{pmatrix} z \\ \beta \end{pmatrix}, \quad G = \begin{pmatrix} cI_{2EN} & 0_{2EN} \\ 0_{2EN} & \frac{1}{c}I_{2EN} \end{pmatrix}.$$
 (12)

We will show that  $u^k = [z^k; \beta^k]$  is Q-linearly convergent to its optimal  $u^* = [z^*; \beta^*]$  with respect to the *G*-norm. Further, the Q-linear convergence of  $u^k = [z^k; \beta^k]$  to  $u^* = [z^*; \beta^*]$  implies that  $x^k$  is R-linearly convergent to its optimal  $x^*$ .

Theorem 1: Consider the ADMM iteration (7) that solves (4). The primal variables x and z have their unique optimal values  $x^*$  and  $z^*$ , respectively; the dual variable  $\beta$  has its unique optimal value  $\beta^*$  that lies in the column space of  $M_{-}^T$ . Recall the definition of u and G defined in (12). If the local objective functions satisfy Assumption 1 and the dual variable  $\beta$  is initialized such that  $\beta^0$  lies in the column space of  $M_{-}^T$ , then for any  $\mu > 1, u^k = [z^k; \beta^k]$  is Q-linearly convergent to its optimal  $u^* = [z^*; \beta^*]$  with respect to the G-norm

$$\|u^{k+1} - u^*\|_G^2 \le \frac{1}{1+\delta} \|u^k - u^*\|_G^2,$$
(13)

where

$$\delta = \min\left\{\frac{(\mu - 1)\tilde{\sigma}_{\min}^2(M_-)}{\mu \sigma_{\max}^2(M_+)}, \frac{m_f}{\frac{c}{4}\sigma_{\max}^2(M_+) + \frac{\mu}{c}M_f^2\tilde{\sigma}_{\min}^{-2}(M_-)}\right\} > 0.$$
(14)

Further,  $x^k$  is R-linearly convergent to  $x^*$  following from

$$\|x^{k+1} - x^*\|_2^2 \le \frac{1}{m_f} \|u^k - u^*\|_G^2.$$
(15)

Proof: See Appendix.

In Theorem 1, (14) shows that  $||u^{k+1} - u^*||_G^2$  is no greater than  $\frac{1}{1+\delta}||u^k - u^*||_G^2$  and hence  $u^k$  converges to  $u^*$  Q-linearly at a rate

$$\rho \le \sqrt{\frac{1}{1+\delta}}.$$

A larger  $\delta$  guarantees faster convergence. On the other hand,  $\frac{1}{1+\delta}$  is a theoretical upper bound of the convergence rate, probably not tight. The Q-linear convergence of  $u^k$  to  $u^*$  translates to the R-linear convergence of x to  $x^*$  as shown in (15).

#### B. Accelerating the Convergence

From (14) we can find that the theoretical convergence rate (more precisely, its upper bound) is given in terms of the network topology, the properties of local objective functions, and the algorithm parameter. The value of  $\delta$  is related with the free parameter  $\mu > 1$ ,  $\sigma_{\max}(M_+)$ ,  $\tilde{\sigma}_{\min}(M_-)$ , the strongly convexity constant  $m_f$  of f, the Lipschitz constant  $M_f$  of  $\nabla f$ , and the algorithm parameter c.

Now we consider tuning the free parameter  $\mu$  and the algorithm parameter c to maximize  $\delta$  and thus accelerate the convergence (i.e., through minimizing  $\frac{1}{1+\delta}$  that is actually an upper bound). From the analysis we will see more clearly how the convergence rate is influenced by the network topology and the local objective functions. For convenience, we define the condition number of f as

$$\kappa_f = \frac{M_f}{m_f}.$$

Recall that  $m_f = \min_i m_{f_i}$  and  $M_f = \max_i M_{f_i}$ . Therefore,  $\kappa_f$  is an upper bound of the condition numbers of the local objective functions. We also define the condition number of the underlying graph  $\mathcal{G}_d$  or  $\mathcal{G}_u$  as

$$\kappa_{\rm G} = \frac{\sigma_{\rm max}(M_+)}{\tilde{\sigma}_{\rm min}(M_-)} = \sqrt{\frac{\sigma_{\rm max}(L_+)}{\tilde{\sigma}_{\rm min}(L_-)}}$$

With regard to the underlying graph, the minimum nonzero singular value of the extended signed Laplacian matrix  $L_-$ , denoted as  $\tilde{\sigma}_{\min}(L_-)$ , is known as its algebraic connectivity [26], [27]. The maximum singular value of the extended signless Laplacian matrix  $L_+$ , denoted as  $\sigma_{\max}(L_+)$ , has also drawn research interests recently [28], [29]. Both  $\sigma_{\max}(L_+)$  and  $\tilde{\sigma}_{\min}(L_-)$  are measures of network connectedness but the former is weaker. Roughly speaking, larger  $\sigma_{\max}(L_+)$  and  $\tilde{\sigma}_{\min}(L_-)$  mean stronger connectedness, and a larger  $\kappa_{\rm G}$  means weaker connectedness.

Keeping the definitions of  $\kappa_f$  and  $\kappa_G$  in mind, the following theorem shows how to choose the free parameter  $\mu$  and the algorithm parameter c to maximize  $\delta$  and accelerate the convergence.

*Theorem 2:* If the algorithm parameter c in (14) is chosen as

$$c = c_{\rm t} = \frac{2\mu^{\frac{1}{2}}M_f}{\sigma_{\rm max}(M_+)\tilde{\sigma}_{\rm min}(M_-)} \tag{16}$$

where

$$\mu = \left(1 + \frac{\kappa_{\rm G}^2}{2\kappa_f^2} - \frac{\kappa_{\rm G}}{2\kappa_f} \sqrt{\frac{\kappa_{\rm G}^2}{\kappa_f^2} + 4}\right)^{-1} > 1, \qquad (17)$$

then

$$\delta = \delta_{\rm t} = \frac{1}{2\kappa_f} \sqrt{\frac{1}{\kappa_f^2} + \frac{4}{\kappa_{\rm G}^2}} - \frac{1}{2\kappa_f^2}$$
(18)

maximizes the value of  $\delta$  in (14) and ensures that (15) holds.

*Proof:* Observing the two values inside the minimization operator in (14), we find that only the second term is relevant with c. It is easy to check that the value of c in (16), no matter how  $\mu$  is chosen, maximizes  $\delta$  as

$$\delta = \min\left\{\frac{(\mu - 1)\tilde{\sigma}_{\min}^2(M_-)}{\mu\sigma_{\max}^2(M_+)}, \frac{m_f\tilde{\sigma}_{\min}(M_-)}{\mu^{\frac{1}{2}}M_f\sigma_{\max}(M_+)}\right\}.$$
 (19)

Inside the minimization operator in (19), the first and second terms are monotonically increasing and decreasing with regard to  $\mu > 1$ , respectively. To maximize  $\delta$ , we choose a value of  $\mu$  such that the two terms are equal. Simple calculations show that the value of  $\mu$  in (17), which is larger than 1, satisfies this condition. The resulting maximum value of  $\delta$  is the one in (18).

The value of  $\delta$  in (18) is monotonically decreasing with regard to  $\kappa_f \geq 1$  and  $\kappa_G > 0$ . This conclusion suggests that a smaller condition number  $\kappa_f$  of f(x) and a smaller condition number  $\kappa_G$  of the graph lead to faster convergence. On the other hand, if these condition numbers keep increasing, the convergence can go arbitrarily slow. In fact, the limit of  $\delta$  in (18) is 0 as  $\kappa_f \to \infty$  or  $\kappa_G \to \infty$ . Given  $\delta_t$ , the upper bound of  $\delta$ , we define the upper bound of the convergence rate as

$$\rho_{\rm t} = \sqrt{\frac{1}{1+\delta_{\rm t}}}.$$

#### **IV. NUMERICAL EXPERIMENTS**

In this section, we provide extensive numerical experiments and supplement to validate our theoretical analysis. We introduce experimental settings in Section IV-A and then study the influence of different factors on the convergence rate in Sections IV-B through IV-E.

# A. Experimental Settings

We generate a network consisting of L agents and possessing at most  $\frac{L(L-1)}{2}$  edges. If the network is randomly generated, we define p, the connectivity ratio of the network, as its actual number of edges divided by  $\frac{L(L-1)}{2}$ . Such a random network is generated with  $\frac{L(L-1)}{2}p$  edges that are uniformly randomly chosen, while ensuring the network connected.

Section	Factor	Conclusion		
IV-B	p, connectivity ratio	Larger $p$ leads to faster convergence		
IV-C	c, algorithm parameter	$c \simeq 0.5 c_{ m t}$ works well		
IV-D	$\kappa_f$ , condition number of objective function	Larger $\kappa_f$ leads to slower convergence		
IV-E	$\kappa_{\rm G}$ , condition number of network	Larger $\kappa_{\rm G}$ leads to slower convergence		
IV-E	D, network diameter	Larger D leads to slower convergence		
IV-E	$d_s$ , geometric average degree	Larger $d_s$ leads to faster convergence		
IV-E	$L_d$ , imbalance of bipartite graphs	Larger $L_d$ leads to faster convergence		

 TABLE II

 SUMMARY OF THE NUMERICAL EXPERIMENTS

We apply the ADMM to a decentralized consensus least squares problem

$$\min_{\tilde{x}} \sum_{i=1}^{L} \frac{1}{2} \|v_i - U_i \tilde{x}\|_2^2.$$
(20)

Here  $\tilde{x} \in \mathbb{R}^3$  is the unknown signal to estimate and its true values follow the normal distribution  $\mathcal{N}(0, I), U_i \in \mathbb{R}^{3\times 3}$  is the linear measurement matrix of agent *i* whose elements follow  $\mathcal{N}(0, 1)$  by default, and  $v_i \in \mathbb{R}^3$  is the measurement vector of agent *i* whose elements are polluted by random noise following  $\mathcal{N}(0, 0.1)$ . In Section IV-D the elements of the matrices  $U_i$  need to be further manipulated to produce different condition numbers  $\kappa_f$  of the objective functions. We reformulate (20) into the form of (3) as

$$\min_{\{x_i\},\{z_{ij}\}} \sum_{i=1}^{L} \frac{1}{2} \|v_i - U_i x_i\|_2^2,$$
  
s.t.  $x_i = z_{ij}, x_j = z_{ij}, \forall (i,j) \in \mathcal{A}.$  (21)

The solution to (20) is denoted by  $x^*$  in which the part of agent *i* is denoted by  $x_i^*$ . The algorithm is stopped once  $||x^k - x^*||_2$  reaches  $10^{-15}$  or the number of iterations *k* reaches 4000, whichever is earlier.

In the numerical experiments, we choose to record the primal error  $||x^k - x^*||_2$  instead of  $||u^k - u^*||_G$  as the latter incurs significant extra computation when the number of agents L is large. But note that  $||x^k - x^*||_2$  is not necessarily monotonic in k. Let the transient convergence rate be  $\rho_k = \frac{||x^k - x^*||_2}{||x^{k-1} - x^*||_2}$ . As  $\rho_k$  fluctuates, we report the *running geometric-average* rate of convergence  $\bar{\rho}_k$  given by

$$\bar{\rho}_{k} := \left(\prod_{s=1}^{k} \rho_{s}\right)^{1/k} \\ = \left(\frac{\|x^{k} - x^{*}\|_{2}}{\|x^{0} - x^{*}\|_{2}}\right)^{1/k} \\ \leq \sqrt{\frac{1}{1+\delta}} \left(\sqrt{\frac{1+\delta}{m_{f}}} \frac{\|u^{0} - u^{*}\|_{G}}{\|x^{0} - x^{*}\|_{2}}\right)^{1/k}, \quad (22)$$

which follows from (13) and (15). While  $u^0$ ,  $u^*$ ,  $x^0 x^*$ , and  $m_f$  influence  $\bar{\rho}_k$ , observing

$$\bar{\rho} := \lim_{k \to +\infty} \bar{\rho}_k \le \sqrt{\frac{1}{1+\delta}}$$

we see that their influence diminishes and the steady state  $\bar{\rho}$  is upper bounded by  $\sqrt{\frac{1}{1+\delta}}$  as  $\rho$  is. Throughout the numerical experiments, we report  $\bar{\rho}_k$  and  $\bar{\rho}$ .

In the following subsections, we demonstrate how different factors influence the convergence rate. We firstly show the evidence of linear convergence, and along the way, the influence of the connectivity ratio p on the convergence rate (see Section IV-B). Secondly, we compare the practical convergence rate using the best theoretical algorithm parameter  $c = c_t$  in (16) and that using the best hand-tuned parameter  $c = c^*$  (see Section IV-C). Thirdly, we check the effect of  $\kappa_f$ , the condition number of the objective function (see Section IV-D). Finally, we show how  $\kappa_f$ , the condition number of the network, as well as other network parameters, influence the convergence rate (see Section IV-E). The numerical experiments are summarized in Table II.

## B. Linear Convergence

To illustrate linear convergence of the ADMM for decentralized consensus optimization, we generate random networks consisting of L = 200 agents. The connectivity ratio of the networks, p, is set to different values. The ADMM parameter is set as  $c = c_t$  (16).

Fig. 1 depicts how the relative error,  $\frac{||x^k - x^*||_2}{||x^*||_2}$ , varies in k. Obviously the convergence rates are linear for all p; a higher connectivity ratio leads to faster convergence. Fig. 2 plots  $\bar{\rho}_k$ , which stabilizes within 10 iterations. From Figs. 1 and 2, one can observe that for such randomly generated networks, varying the connectivity ratio p within the range [0.08, 1] does not significantly change the convergence rate. The reason is that when p is larger than a certain threshold, its value makes little influence on  $\kappa_G$  (see Table III in Section IV-C). We will discuss more about the influence of  $\kappa_G$  in Section IV-D.

As a comparison, we also demonstrate the convergence of the distributed gradient descent (DGD) method in Figs. 1 and 2. Using a diminishing stepsize  $1/k^{1/3}$  [30], the DGD shows sublinear convergence that is slow even for a complete graph (i.e., p = 1).

## C. Algorithm Parameter

Here we discuss the influence of the ADMM parameter c on the convergence rate. The best theoretical value  $c = c_t$  in (16), though optimizing the upper bound of the convergence rate, does not give best practical performance. We vary c, and plot the steady-state running geometric-average rates of convergence  $\bar{\rho}$ in Fig. 3. For each curve that corresponds to a unique p, we mark

 TABLE III

 Settings and Convergence Rates Corresponding to Figs. 1, 2, 4, and 5

Connectivity ratio p		Best theoretical $c_{\rm t}$		Best practical $c^*$		Theoretical rate
(L = 200  agents)	$\kappa_{ m G}$	С	$\bar{ ho}$	c	$\bar{ ho}$	$ ho_{ m t}$
0.01	33.00	123.8	0.9960	3.110	0.9189	0.9908
0.02	7.032	3.477	0.9314	0.5510	0.7014	0.9806
0.04	3.500	0.6714	0.8358	0.1687	0.5624	0.9295
0.08	2.221	0.1677	0.7088	0.05303	0.4297	0.8526
1.00	1.411	0.006837	0.5348	0.002722	0.2714	0.7313



Fig. 2. Running geometric-average rate of convergence  $\bar{\rho}_k$  versus iteration k.

the best theoretical value  $c_t$  and the best practical value  $c^*$ . Consistently,  $c_t$  are larger than  $c^*$ .

Now we set  $c = c^*$ , the hand-tuned optimal value, and plot  $\frac{\|x^k - x^*\|_2}{\|x^*\|_2}$  in Fig. 4 as per Fig. 1 and  $\bar{\rho}_k$  in Fig. 5 as per Fig. 2. Comparing to those using  $c = c_t$ , the best theoretical value, in Figs. 1 and 2, the convergence improves significantly. The numerical quantities of Figs. 1, 2, 4, and 5 are given in Table III.

It appears that  $c_t$  is a stable overestimate of  $c^*$ . Therefore, we recommend  $c = \theta c_t$  for nearly optimal convergence using some  $\theta \in (0, 1)$ . Fig. 6 illustrates the convergence corresponding to different values of  $\theta$ . We randomly generate 4000 connected networks with L = 200 agents whose connectivity ratios are



Fig. 3. Steady-state running geometric-average rate of convergence  $\bar{\rho}$  versus algorithm parameter c.



uniformly distributed on  $[\frac{2}{L}, 1]$ . The random networks are divided into 20 groups according to their condition numbers  $\kappa_{\rm G}$ . For each group of the random networks, the values of  $\bar{\rho}$  are plotted with error bars, and compared with the theoretical upper bound  $\rho_{\rm t}$ . For this dataset,  $\theta \simeq 0.5$  appear to be a good overall choice. A smaller  $\theta$  imposes a risk of slower convergence when  $\kappa_{\rm G}$  is small.

# D. Condition Number of the Objective Function

Now we study how  $\kappa_f$ , the condition number of the objective function, affects the convergence rate. We generate random



Fig. 5. Running geometric-average rate of convergence  $\bar{\rho}_k$  versus iteration k.



Fig. 6. Convergence performance obtained with  $c = \theta c_t$  for varying  $\theta$ , where  $c_t$  is analytically given in (16).

networks consisting of L = 200 agents with different connectivity ratios p. We set  $c = c_t$ . To produce different  $\kappa_f$ , we first generate a linear measurement matrix  $U_i$  with its elements following  $\mathcal{N}(0, 1)$ . Second, we apply singular value decompositions to  $U_i$ , scale the singular values to the range  $[\sqrt{\frac{1}{\kappa_f}}, 1]$ , and rebuild  $U_i$ .

Fig. 7 shows that the theoretical convergence rates  $\rho_t$  are monotonically increasing as  $\kappa_f$  increases, which is consistent with Theorem 2. When the connectivity ratios p are small, the trend of  $\bar{\rho}$  disobeys the theoretical analysis. It is because that our upper bound of the convergence rate, becomes loose when the network connectedness is poor. When the network is well-connected (say p = 1), we can observe a positive correlation between  $\bar{\rho}$  and  $\kappa_f$ , which coincides with the theoretical analysis.

# E. Network Topology

Last we study how the network topology affects the convergence rate. Besides the condition number  $\kappa_{G}$  of the network that is relevant, we also consider other network parameters including the network diameter, geometric average degree, as well



Fig. 7. Convergence performance versus the condition number  $\kappa_f$  of the objective function at different connectivity ratios p.



Fig. 8. Convergence performance versus the condition number of the network  $\kappa_{\rm G}$  obtained with networks of different topologies (random, line, cycle, star, and complete) and of different sizes (L = 50, 200, 500).

as imbalance of bipartite networks. In the numerical experiments, the local objective functions are generated as described in Section IV-A. The algorithm parameter is set as  $c = c_t$ .

1) Condition Number of the Network: As it is difficult to precisely design  $\kappa_{\rm G}$ , the condition number of the network, we run a large number of trials to sample  $\kappa_{\rm G}$ . We randomly generate 4000 connected networks with L = 50, 200, 500 agents, 12000 networks in total. Their connectivity ratios are uniformly distributed on  $[\frac{2}{L}, 1]$ . In addition, we generate special networks with topologies of the line, cycle, star, complete, and grid types. The grid networks are generated in a 3D space  $(2 \times 5 \times 5, 5 \times 5 \times 8, \text{ and } 5 \times 10 \times 10)$ .

Fig. 8 depicts the effect of  $\kappa_{\rm G}$  on the convergence rate. In Fig. 8, the dashed curve with error bars correspond to the random networks, and the individual points correspond to the special networks. There is only one dashed curve in the plot since L = 50,200,500 do not make significant differences. The networks of the line, cycle, complete, and grid topologies 1758



Fig. 9. Convergence performance versus the condition number of the network  $\kappa_{\rm G}$  obtained with networks of different topologies (random, line, cycle, star, complete, and grid) and of different sizes (L = 50, 200, 500).

generate points in the plot that are nearly on the dashed curve, which indicates that  $\kappa_{\rm G}$  is a good indicator for convergence rate. In addition, the trends of  $\bar{\rho}$ , the steady-state running geometric-average rate of convergence, and  $\rho_{\rm t}$ , the theoretical rate of convergence, are consistent. The points corresponding to the three networks of the star topology are away from the dashed side.

We observe that the convergence rate is closely related to  $\kappa_{\rm G}$ , less to L. To reach a target convergence rate, one therefore shall have a sufficiently small  $\kappa_{\rm G}$ , which in turn depends on L and p, as well as other factors. To obtain a sufficiently small  $\kappa_{\rm G}$ , typically, p needs to be large if L is small, but not as large if L is large. In other words, if one has a network with a large number of agents (say L = 200), a small connectivity ratio (say p = 0.1) will lead to a small  $\kappa_{\rm G}$  and thus fast convergence.

With the same  $\kappa_{G}$ , the networks with the star topology have much faster convergence than random networks. We shall discuss this special topology at the end of this subsection.

2) Network Diameter: The network diameter D is defined as the longest distance between any pair of agents in the network. In decentralized consensus optimization, D is related to how many iterations the information from one agent will reach all the other agents.

To discuss the effect of the network diameter on the convergence rate, we randomly generated 4000 connected networks with L = 200 agents and connectivity ratios uniformly distributed on  $[\frac{2}{L}, 1]$ . We also generate the networks of the line, cycle, star, complete, and grid topologies. Most randomly generated networks possess small diameters. In this experiment, the numbers of those with D = 2,  $3 \le D \le 4$  and  $5 \le D \le 198$ are 3141, 717 and 142, respectively. From Fig. 9, we conclude that in general a larger diameter tends to cause a worse condition number of the network and thus slower convergence, though this relationship is interfered by network properties.

3) Geometric Average Degree: Define  $d_{\min}$  and  $d_{\max}$  as the largest and smallest degrees of the agents in the network, respectively. The geometric average degree  $d_s = \sqrt{d_{\min}d_{\max}}$  reflects the agents' number of neighbors in a geometric average sense.



Fig. 10. Convergence performance versus the condition number  $\kappa_{\rm G}$  of the network and the network diameter *D* obtained with networks of different topologies (random, line, cycle, star, complete, and grid) and of size L = 200.



Fig. 11. Convergence performance versus the condition number  $\kappa_{\rm G}$  of the network and the imbalance of bipartite networks  $L_d$  obtained with networks of random and star topologies and of size L = 200.

Its value reaches maximum at L-1 if the topology is complete; and reaches minimum  $\sqrt{2}$  when the topology is a line.

To discuss the effect of the network diameter on the convergence rate, we randomly generated 4000 connected networks with L = 200 agents and connectivity ratios uniformly distributed on  $[\frac{2}{L}, 1]$ . We also generate the networks of the line, cycle, star, complete, and grid topologies. Most randomly generated networks possess small diameters. In this experiment, the numbers of those with D = 2,  $3 \le D \le 4$  and  $5 \le D \le 198$ are 3141, 717 and 142, respectively. From Fig. 9, we conclude that in general a larger diameter tends to cause a worse condition number of the network and thus slower convergence, though this relationship is interfered by network properties.

4) Imbalance of Bipartite Networks: Let  $\mathcal{B}(\mathcal{L}_A, \mathcal{L}_B)$  denote the class of bipartite networks with  $|\mathcal{L}_A|$  agents in one group and  $|\mathcal{L}_B|$  agents in another group. Agents within either group cannot directly communicate with each other. For a bipartite network consisting of  $L = |\mathcal{L}_A| + |\mathcal{L}_B|$  agents, its imbalance is defined as  $L_d = |\mathcal{L}_A| - |\mathcal{L}_B|$ , which can vary between 0 and L - 2. We randomly generate 1000 bipartite graphs of size L = 200, whose connectivity ratios p are uniformly distributed on  $\left[\frac{2}{L}, \frac{(L+L_d)(L-L_d)}{2L(L-1)}\right]$ , for each of the cases  $L_d = 196, 180, 140, 0$ . The star topology corresponds to a special bipartite network with  $L_d = L - 2 = 198$ . From Fig. 11, we find that for the same  $\kappa_G$ , the networks with larger  $L_d$  have faster convergence. An extreme example is the network of the star topology. This observation suggests us to assign few "hot spots" to relay information for fast convergence, if  $\kappa_G$  is fixed in advance. However, this approach may cause robustness or scalability issues because the relaying agents are subject to extensive communication burden. Hence there is a tradeoff between fast convergence and robustness or scalability in network design.

# V. CONCLUSION

We apply the ADMM to a reformulation of a general decentralized consensus optimization problem. We show that if the objective function is strongly convex, the decentralized ADMM converges at a globally linear rate, which can be given explicitly. It is revealed that several factors affect the convergence rate that include the topology-related properties of the network, the condition number of the objective function, and the algorithm parameter. Numerical experiments corroborate and supplement our theoretical findings. Our analysis sheds light on how to construct a network and tune the algorithm parameter for fast convergence.

# APPENDIX

*Proof:* Consider the ADMM updates (7) and the KKT conditions (11). Subtracting the three equations in (11) from the corresponding equations in (7) yields

$$\nabla f(x^{k+1}) - \nabla f(x^*)$$
  
=  $cM_+(z^k - z^{k+1}) - M_-(\beta^{k+1} - \beta^*),$  (23)

$$\frac{c}{2}M_{-}^{T}(x^{k+1} - x^{*}) = \beta^{k+1} - \beta^{k}, \qquad (24)$$

$$\frac{1}{2}M_{+}^{T}(x^{k+1} - x^{*}) = z^{k+1} - z^{*},$$
(25)

respectively.

To prove the Q-linear convergence of  $||u^{k+1} - u^*||_G^2$  we use  $m_f ||x^{k+1} - x^*||_2^2$  as an intermediate. Based on Assumption 1, f(x) is strongly convex with a constant  $m_f$  such that

$$m_f \|x^{k+1} - x^*\|_2^2 \le \langle x^{k+1} - x^*, \nabla f(x^{k+1}) - \nabla f(x^*) \rangle.$$
 (26)

Using (23), we can split the right-hand side of (26) to two terms

$$\langle x^{k+1} - x^*, \nabla f(x^{k+1}) - \nabla f(x^*) \rangle = \langle x^{k+1} - x^*, cM_+(z^k - z^{k+1}) - M_-(\beta^{k+1} - \beta^*) \rangle = \langle x^{k+1} - x^*, cM_+(z^k - z^{k+1}) \rangle + \langle x^{k+1} - x^*, -M_-(\beta^{k+1} - \beta^*) \rangle = c \langle M_+^T(x^{k+1} - x^*), z^k - z^{k+1} \rangle + \langle -M_-^T(x^{k+1} - x^*), \beta^{k+1} - \beta^* \rangle.$$
 (27)

Substituting (24) and (25) to (27) we can eliminate the term  $x^{k+1} - x^*$  and obtain

$$\langle x^{k+1} - x^*, \nabla f(x^{k+1}) - \nabla f(x^*) \rangle = 2c \langle z^k - z^{k+1}, z^{k+1} - z^* \rangle + \frac{2}{c} \langle \beta^k - \beta^{k+1}, \beta^{k+1} - \beta^* \rangle.$$
 (28)

Recall the definition of u and G defined in (12). It is obvious that the right-hand side of (28) can be written as a compact form  $2(u^k - u^*)^T G(u^{k+1} - u^k)$ . Using the equality  $2(u^k - u^*)^T G(u^{k+1} - u^k) = ||u^k - u^*||_G^2 - ||u^{k+1} - u^*||_G^2 - ||u^k - u^{k+1}||_G^2$ , (28) is equivalent to

$$\langle x^{k+1} - x^*, \nabla f(x^{k+1}) - \nabla f(x^*) \rangle$$
  
=  $\| u^k - u^* \|_G^2 - \| u^{k+1} - u^* \|_G^2 - \| u^k - u^{k+1} \|_G^2,$  (29)

and consequently using (26)

$$m_f \|x^{k+1} - x^*\|_2^2 \le \|u^k - u^*\|_G^2 - \|u^{k+1} - u^*\|_G^2 - \|u^k - u^{k+1}\|_G^2.$$
(30)

Having (30) at hand, to prove (13) we only need to show

$$\|u^{k} - u^{k+1}\|_{G}^{2} + m_{f}\|x^{k+1} - x^{*}\|_{2}^{2} \ge \delta \|u^{k+1} - u^{*}\|_{G}^{2},$$
(31)

which is equivalent to

$$c\|z^{k+1} - z^{k}\|_{2}^{2} + \frac{1}{c}\|\beta^{k+1} - \beta^{k}\|_{2}^{2} + m_{f}\|x^{k+1} - x^{*}\|_{2}^{2}$$
$$\geq \delta c\|z^{k+1} - z^{*}\|_{2}^{2} + \frac{\delta}{c}\|\beta^{k+1} - \beta^{*}\|_{2}^{2}.$$
 (32)

The idea of proof is to show that  $\delta c \|z^{k+1} - z^*\|_2^2$  and  $\frac{\delta}{c} \|\beta^{k+1} - \beta^*\|_2^2$  are upper bounded by two non-overlapping parts of the left-hand side of (32), respectively.

The upper bound of  $||z^{k+1} - z^*||_2^2$  follows from (25) that shows  $\frac{1}{2}M_+^T(x^{k+1} - x^*) = z^{k+1} - z^*$ . Hence we have

$$||z^{k+1} - z^*||_2^2$$
  
=  $\frac{1}{4} ||M_+^T (x^{k+1} - x^*)||_2^2$   
 $\leq \frac{1}{4} \sigma_{\max}^2 (M_+) ||x^{k+1} - x^*||_2^2,$  (33)

where  $\sigma_{\max}(M_+)$  is the largest singular value of  $M_+$ . To find the upper bound of  $\|\beta^{k+1} - \beta^*\|_2^2$ , we use two inequalities  $\sigma_{\max}^2(M_+)\|z^{k+1} - z^k\|_2^2 \ge \|M_+^T(z^k - z^{k+1})\|_2^2$  and  $M_f^2\|x^{k+1} - x^*\|_2^2 \ge \|\nabla f(x^{k+1}) - \nabla f(x^*)\|_2^2$ ; the latter holds since f(x) has Lipschitz continuous gradients with a constant  $M_f$ . Therefore, given the positive algorithm parameter c and any  $\mu > 1$  it holds

$$c^{2}\sigma_{\max}^{2}(M_{+})\|z^{k+1} - z^{k}\|_{2}^{2} + (\mu - 1)M_{f}^{2}\|x^{k+1} - x^{*}\|_{2}^{2}$$
  

$$\geq \|cM_{+}^{T}(z^{k} - z^{k+1})\|_{2}^{2} + (\mu - 1)\|\nabla f(x^{k+1}) - \nabla f(x^{*})\|_{2}^{2}.$$
 (34)

Recall that from (23)  $cM_+(z^k - z^{k+1})$  is the summation of  $\nabla f(x^{k+1}) - \nabla f(x^*)$  and  $M_-(\beta^{k+1} - \beta^*)$ . Hence we can apply

the basic inequality  $||a + b||_2^2 + (\mu - 1)||a||_2^2 \ge (1 - \frac{1}{\mu})||b||_2^2$ , which holds for any  $\mu > 0$ , to (34) and obtain

$$c^{2}\sigma_{\max}^{2}(M_{+})\|z^{k+1} - z^{k}\|_{2}^{2} + (\mu - 1)M_{f}^{2}\|x^{k+1} - x^{*}\|_{2}^{2}$$
  
$$\geq \left(1 - \frac{1}{\mu}\right)\|M_{-}(\beta^{k+1} - \beta^{*})\|_{2}^{2}.$$
 (35)

Since by assumption  $\beta^0$  is initialized such that it lies in the column space of  $M_{-}^T$ , we know that  $\beta^{k+1}$  lies in the column space of  $M_{-}^T$  too; see the ADMM updates (7). Because  $\beta^*$  also lies in the column space of  $M_{-}^T$ ,  $\|M_{-}(\beta^{k+1} - \beta^*)\|_2^2 \geq \tilde{\sigma}_{\min}^2(M_{-})\|\beta^{k+1} - \beta^*\|_2^2$  where  $\tilde{\sigma}_{\min}(M_{-})$  is the smallest nonzero singular value of  $M_{-}$ . Therefore from (35) we can upper bound  $\|\beta^{k+1} - \beta^*\|_2^2$  by

$$c^{2}\sigma_{\max}^{2}(M_{+})\|z^{k+1} - z^{k}\|_{2}^{2} + (\mu - 1)M_{f}^{2}\|x^{k+1} - x^{*}\|_{2}^{2}$$
  
$$\geq \left(1 - \frac{1}{\mu}\right)\tilde{\sigma}_{\min}^{2}(M_{-})\|\beta^{k+1} - \beta^{*}\|_{2}^{2}.$$
 (36)

Combining (33) and (36), we prove (32). From (33) we have

$$\frac{c}{4}\sigma_{\max}^2(M_+)\|x^{k+1} - x^*\|_2^2 \ge c\|z^{k+1} - z^*\|_2^2.$$
(37)

From (36) we have

$$\frac{c\mu\sigma_{\max}^2(M_+)}{(\mu-1)\tilde{\sigma}_{\min}^2(M_-)} \|z^{k+1} - z^k\|_2^2 + \frac{\mu M_f^2}{c\tilde{\sigma}_{\min}^2(M_-)} \|x^{k+1} - x^*\|_2^2 \\ \ge \frac{1}{c} \|\beta^{k+1} - \beta^*\|_2^2.$$
(38)

Summing up (37) and (38) yields

$$\frac{c\mu\sigma_{\max}^{2}(M_{+})}{(\mu-1)\tilde{\sigma}_{\min}^{2}(M_{-})} \|z^{k+1} - z^{k}\|_{2}^{2} + \left(\frac{\mu M_{f}^{2}}{c\tilde{\sigma}_{\min}^{2}(M_{-})} + \frac{c}{4}\sigma_{\max}^{2}(M_{+})\right) \|x^{k+1} - x^{*}\|_{2}^{2} \\ \geq c\|z^{k+1} - z^{*}\|_{2}^{2} + \frac{1}{c}\|\beta^{k+1} - \beta^{*}\|_{2}^{2}.$$
(39)

Apparently,  $\delta$  in (14) satisfies

$$c\|z^{k+1} - z^{k}\|_{2}^{2} + m_{f}\|x^{k+1} - x^{*}\|_{2}^{2}$$
  

$$\geq \delta c\|z^{k+1} - z^{*}\|_{2}^{2} + \frac{\delta}{c}\|\beta^{k+1} - \beta^{*}\|_{2}^{2}, \quad (40)$$

and consequently (32), which proves (13).

To prove the R-linear convergence of  $x^k$  to  $x^*$ , we observe that (30) implies  $m_f ||x^{k+1} - x^*||_2^2 \le ||u^k - u^*||_G^2$ , which proves (15).

# REFERENCES

 W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "Linearly convergent decentralized consensus optimization with the alternating direction method of multipliers," in *Proc. 38th Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 4613–4617.

- [2] G. Inalhany, D. Stipanovic, and C. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination," in *Proc. 41st IEEE Conf. Dec. Control*, 2002, pp. 1147–1155.
- [3] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control: Collective group behavior through local interaction," *IEEE Control Syst. Mag.*, vol. 27, pp. 71–82, Apr. 2007.
- [4] B. Johansson, "On Distributed Optimization in Networked Systems," Ph.D. dissertation, Electr. Eng. Dept., KTH, Stockholm, Sweden, 2008.
- [5] L. Xiao, S. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.
- [6] A. Dimakis, S. Kar, M. R. J. Moura, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [7] J. Predd, S. Kulkarni, and H. Poor, "A collaborative training algorithm for distributed learning," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1856–1871, 2009.
- [8] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.
- [9] 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, 2008.
- [10] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3816–3827, 2010.
  [11] J. Bazerque and G. Giannakis, "Distributed spectrum sensing for
- [11] J. Bazerque and G. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, 2010.
- [12] J. Bazerque, G. Mateos, and G. Giannakis, "Group-Lasso on splines for spectrum cartograph," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, 2011.
- [13] V. Kekatos and G. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, 2013.
- [14] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, 2013.
- [15] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [16] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, pp. 516–545, 2010.
- [17] K. Tsianos and M. Rabbat, "Distributed strongly convex optimization," in Proc. 50th Ann. Allerton Conf. Commun., Control Comput., 2012, pp. 593–600.
- [18] 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, 2012.
- [19] K. Tsianos, S. Lawlor, and M. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. 51st IEEE Ann. Conf. Dec. Control*, 2012, pp. 5453–5458.
- [20] 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, 2011.
- [21] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Nashua, NH, USA: Athena Scientific, 1997.
- [22] M. Rabbat and R. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, 2006.
- [23] B. He and X. Yuan, "On the O(1/t) convergence rate of the Douglas-Rachford alternating direction method," SIAMJ. Numer. Anal., vol. 50, no. 2, pp. 700–709, 2012.
- [24] M. Hong and Z. Luo, "On the linear convergence of the alternating direction method of multipliers," 2013 [Online]. Available: http://arxiv. org/ pdf/1208.3922v3.pdf
- [25] W. Deng and W. Yin, "On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers," Rice Univ., Houston, TX, USA, Tech. Rep. TR12-14, 2012.
- [26] F. Chung, Spectral Graph Theory, ser. CBMS Regional Conf. Series Mathematics, No. 92. Providence, RI, USA: Amer. Math. Soc., 1996.
- [27] M. Fiedler, "Algebra connectivity of graphs," *Czechoslovake Math. J.*, vol. 23, no. 98, pp. 298–305, 1973.
- [28] D. Cvetkovic, P. Rowlinson, and S. Simic, "Signless Laplacians of finite graphs," *Linear Algebra Appl.*, vol. 423, pp. 155–171, 2007.

- [29] Y. Chen and L. Wang, "Sharp bounds for the largest eigenvalue of the signless Laplacian of a graph," *Linear Algebra Appl.*, vol. 433, pp. 908–913, 2010.
- [30] D. Jakovetic, J. Xavier, and J. M. Moura, "Convergence rate analysis of distributed gradient methods for smooth optimization," in *Proc. 20th IEEE Telecommun. Forum (TELFOR)*, 2012, pp. 867–870.



Kun Yuan received the B.E. degree in telecommunication engineering from Xidian University in 2011. He currently working toward the M.S. degree in control theory and control engineering in the Department of Automation, University of Science and Technology of China. His current research focuses on decentralized optimization of networked multi-agent systems.



Wei Shi received the B.E. degree in automation from the University of Science and Technology of China in 2010. He is currently working toward the Ph.D. degree in control theory and control engineering in the Department of Automation, University of Science and Technology of China. His current research focuses on decentralized optimization of networked multi-agent systems.



**Gang Wu** received the B.E. degree in automation and the M.S. degree in control theory and control engineering from University of Science and Technology of China in 1986 and 1989, respectively. Since 1991, he has been in the Department of Automation, University of Science and Technology of China, where he is now a Professor. His current research interests are advanced control and optimization of industrial processes.



Qing Ling received the B.E. degree in automation and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China in 2001 and 2006, respectively. From 2006 to 2009, he was a Post-Doctoral 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. His current research focuses on decentralized optimiza-

tion of networked multi-agent systems.



**Wotao Yin** received the B.S. degree in mathematics and applied mathematics from Nanjing University in 2001 and the M.S. and Ph.D. degrees in operations research from Columbia University in 2003 and 2006, respectively. From 2006 to 2013, he was an Assistant Professor and then an Associate Professor in the Department of Computational and Applied Mathematics, Rice University. Since 2013, he has been a Professor in the Department of Mathematics, University of California, Los Angeles, CA, USA. His current research interest is large-scale de-

centralized/distributed optimization.