DLM: Decentralized Linearized Alternating Direction Method of Multipliers

Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro

***Abstract*— This paper develops the Decentralized Linearized Alternating Direction Method of Multipliers (DLM) that minimizes a sum of local cost functions in a multiagent network. The algorithm mimics operation of the decentralized alternating direction method of multipliers (DADMM) except that it linearizes the optimization objective at each iteration. This results in iterations that, instead of successive minimizations, implement steps whose cost is akin to the much lower cost of the gradient descent step used in the distributed gradient method (DGM). The algorithm is proven to converge to the optimal solution when the local cost functions have Lipschitz continuous gradients. Its rate of convergence is shown to be linear if the local cost functions are further assumed to be strongly convex. Numerical experiments in least squares and logistic regression problems show that the number of iterations to achieve equivalent optimality gaps are similar for DLM and ADMM and both much smaller than those of DGM. In that sense, DLM combines the rapid convergence of ADMM with the low computational burden of DGM.**

***Index Terms*— multiagent network, decentralized optimization, linearized alternating direction method of multipliers**

I. Introduction

Consider a multiagent system composed of n networked agents whose goal is to solve a decentralized optimization problem with a separable cost of the form

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

The variable $\tilde{x} \in \mathbb{R}^p$ is common to all agents whose aim is to find an optimal argument $\tilde{x}^* = \operatorname{argmin} \sum_{i=1}^n f_i(\tilde{x})$. We say (1) is decentralized because, despite the common goal of finding $\tilde{x}^*$, the local cost function $f_i : \mathbb{R}^p \to \mathbb{R}$ is known to agent i only. The decentralized optimization problem (1) arises in various applications, such as event detection in wireless sensor networks [2]–[4], state estimation in smart grids [5], [6], spectrum sensing in cognitive radio networks [7]–[9], and decentralized machine learning in computer networks [10]–[12], to name a few.

While aggregating all functions at a common location is possible, it is more efficient to design decentralized optimization algorithms in which agents iterate through information exchanges with neighboring agents. Decentralized algorithms generating iterates that converge to an optimal argument $\tilde{x}^*$ of (1) can be divided into those operating in the primal domain and those operating in the dual domain. In the primal domain methods, each agent averages its local iterate with those of neighbors and descends along its local negative (sub)gradient direction. Typical primal domain methods include the distributed (sub)gradient method (DGM) [13]–[16] and the dual averaging method [17], [18]. The dual domain methods rewrite (1) to a constrained form where the constraints force local solutions to reach global consensus. The dual ascent method is hence applicable because (sub)gradients of the dual function depend on local and neighboring solutions only and can thereby be computed without global cooperation [19], [20]. The alternating direction method of multipliers (ADMM) modifies dual ascent by penalizing the constraints with a quadratic term and the resulting algorithm, the decentralized ADMM (DADMM), improves numerical stability as well as rate of convergence [21]–[24].

The main advantage of the primal domain methods is their low computation burden. The operation required at each iteration is akin to a (sub)gradient step and, hence, entails a small computation burden. However, existing primal domain methods suffer from either slow convergence or low accuracy. With time-varying stepsizes, the distributed gradient method and the dual averaging method converge to the optimal solution at sublinear rates [16], [17]. If the stepsize is constant, the distributed gradient method is able to achieve a linear rate of convergence under the assumption that the local cost functions have Lipschitz gradients and are strongly convex; however, the algorithm converges not to the actual optimizer but to a neighborhood of the optimal solution [15]. The dual domain methods, on the other hand, converge relatively fast to the exact optimal solution [24] but have high computation burden. Indeed, each agent needs to solve an optimization problem at each iteration, whose objective is the local cost function plus some other term – a linear term in the dual ascent method [19] and a quadratic term in DADMM [21]–[23]. Since this local optimization problem has no explicit solution unless the local cost function has a special structure, it has to be solved by a local iterative minimization method.

This paper develops the decentralized linearized ADMM (DLM) algorithm that enjoys the advantages of both the primal and the dual domain methods, i.e., low computation burden and fast convergence to the exact optimal solution. Besides developing DLM we prove its convergence to the optimal argument provided that the local cost functions have Lipschitz continuous gradients. If we further assume that the local cost functions are also strongly convex we show that the rate of convergence to the optimal solution is linear. This convergence guarantees are analogous to the ones for DADMM [24]. We point out that DLM is related to various (centralized)

Qing Ling, Wei Shi, and Gang Wu are with Department of Automation, University of Science and Technology of China, Hefei, Anhui, China 230026. Alejandro Ribeiro is with Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, USA 19104. Qing Ling is supported by NSFC grant 61004137. Alejandro Ribeiro is supported by NSF CAREER CCF-0952867, NSF CCF-1017454, and AFOSR MURI FA9550-10-1-0567. Part of the results in this paper appeared in [1].

algorithms that are known as linearized ADMM [25]–[27] but it is not identical to either of them; see Remark 2. More closely related approaches are the decentralized inexact ADMM [28] and ADMM+ [29]. Their differences with DLM and with respect to each other are relatively minor but lead to different analyses that provide complementary insights.

This paper begins by reformulating the decentralized optimization problem (1) to a constrained form, which is solved through alternating minimization of the augmented Lagrangian and linearization of the local cost functions (Section II). We further reorganize the iterations and introduce an initialization condition so that a simpler DLM algorithm is obtained (Proposition 1). We then proceed to analyze convergence properties of DLM (Section III) under the assumptions that the local objective functions have Lipschitz continuous gradients and are strongly convex (Section III-A). The assumption of Lipschitz continuous gradients guarantees convergence of the algorithm (Theorem 1 in Section III-B) while the addition of a strong convexity assumption establishes a linear rate of convergence (Theorem 2 in Section III-C). Numerical experiments are presented (Section IV) for least squares (Section IV-A) and logistic regression (Section IV-B) problems. The numerical results corroborate theoretical findings on DLM. They also show the number of iterations to achieve equivalent optimality gaps are similar for DLM and DADMM and both much smaller than those of DGM. In that sense, DLM combines the rapid convergence of DADMM with the low computational burden of DGM (Section V).

Notation For column vectors $v_1, \ldots, v_n$ we use the notation $v := [v_1; \ldots; v_n]$ to represent the stacked column vector v. For a block matrix M we use $(M)_{i,j}$ to denote the (i,j)th block. Given matrices $M_1, \ldots, M_n$ we use $\text{diag}(M_1, \ldots, M_n)$ to denote the block diagonal matrix whose ith diagonal block is M_i. Let $\|v\|$ be the Euclidean norm of v. For a positive definite matrix M, $\|v\|_M := \sqrt{v^T M v}$ is the norm of v with respect to M and $\langle v_1, v_2 \rangle_M := v_1^T M v_2$ is the inner product of v_1 and v_2 with respect to M.

II. Algorithm Development

Consider a connected network composed of a set of n agents $\mathcal{V} = \{1, \ldots, n\}$ and a set of m arcs $\mathcal{A} = \{1, \ldots, m\}$, where each arc $e \sim (i,j)$ is associated with an ordered pair (i,j) indicating that i can communicate to j. Assume communication is bidirectional so that if arc $e \sim (i,j) \in \mathcal{A}$ the opposite arc $e' \sim (j,i) \in \mathcal{A}$. We refer to agents adjacent to i as the neighbors of i and denote their set as $\mathcal{N}_i := \{j : (i,j) \in \mathcal{A}\}$. The cardinality of this set is represented by $d_i : |\mathcal{N}_i|$ and referred to as the degree of agent i. Further define the block arc source matrix $A_s \in \mathbb{R}^{mp \times np}$ containing $m \times n$ square blocks $(A_s)_{e,i} \in \mathbb{R}^{p \times p}$ of dimension p. The block $(A_s)_{e,i}$ is not identically null if and only if the arc $e \sim (i,j)$ *originates* at node i in which case $(A_s)_{e,i} = I_p$ is given by the $p \times p$ identity matrix. Likewise, define the block arc destination matrix $A_d \in \mathbb{R}^{mp \times np}$ containing $m \times n$ square blocks $(A_d)_{e,i} \in \mathbb{R}^{p \times p}$. The block $(A_d)_{e,j} = I_p \in \mathbb{R}^{p \times p}$ if the arc $e \sim (i,j)$ *terminates* at node j and is null otherwise. The extended oriented incidence matrix is then written as $E_o = A_s - A_d$ and the unoriented incidence matrix as $E_u = A_s + A_d$. The extended oriented (signed) Laplacian is given by $L_o = (1/2) E_o^T E_o$, the unoriented (unsigned) Laplacian by $L_u = (1/2) E_u^T E_u$, and the degree matrix containing degrees d_i in the diagonal blocks by $D = (1/2)(L_o + L_u)$. Let Γ_u and γ_u be the largest and smallest eigenvalues of L_u, respectively, and γ_o be the smallest nonzero eigenvalue of L_o. The eigenvalues Γ_u, γ_u, and γ_o are measures of network connectedness [30]. We make the following assumptions on the local cost functions f_i.

Assumption 1 The local cost functions f_i are proper closed convex and differentiable.

Assumption 2 The local cost functions f_i have Lipschitz continuous gradients. There is a positive constant $M_f > 0$ such that for all agents i and for any pair of points $\tilde{x}_a$ and $\tilde{x}_b$ it holds $\|\nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b)\| \leq M_f \|\tilde{x}_a - \tilde{x}_b\|$.

Assumption 3 The local cost functions f_i are strongly convex. There is a positive constant $m_f > 0$ such that for all agents i for any pair of points $\tilde{x}_a$ and $\tilde{x}_b$ it holds $[\tilde{x}_a - \tilde{x}_b]^T [\nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b)] \geq m_f \|\tilde{x}_a - \tilde{x}_b\|^2$.

A. DADMM: Decentralized ADMM

To develop DLM for a problem having the form of (1) we introduce variables $x_i \in \mathbb{R}^p$ representing local copies of the variable $\tilde{x}$, auxiliary variables $z_{ij} \in \mathbb{R}^p$ associated with each arc $(i,j) \in \mathcal{A}$, and reformulate (1) as

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i(x_i), \\ \text{s.t.} \quad & x_i = z_{ij}, \ x_j = z_{ij}, \ \text{for all } (i,j) \in \mathcal{A}. \end{aligned} \tag{2}$$

The constraints $x_i = z_{ij}$ and $x_j = z_{ij}$ force neighboring agents i and j to reach consensus on their local copies x_i and x_j. Since the edges are bidirectional, we have that $x_i = z_{ij} = z_{ji} = x_j$, for all$(i,j) \in \mathcal{A}$. Insofar as the network is connected, this local consensus implies that the variables in the feasible set of (2) must be $x_i = x_j$ for all, not necessarily neighboring, agents $i, j \in \mathcal{V}$. Thus, (2) is equivalent to (1) for connected networks in the sense that for all i and j the optimal arguments of (2) must satisfy $x_i^* = \tilde{x}^*$ and $z_{ij}^* = \tilde{x}^*$ where, we recall, $\tilde{x}^*$ is an optimal solution of (1).

Using the definitions of the arc source and arc destination matrices A_s and A_d we can rewrite (2) in a more compact form. To do so define the vector $x := [x_1; \ldots; x_n] \in \mathbb{R}^{np}$ concatenating all local variables x_i and the vector $z = [z_1; \ldots; z_m] \in \mathbb{R}^{mp}$ concatenating all auxiliary variables $z_e = z_{ij}$. Introduce the aggregate function $f : \mathbb{R}^{np} \to \mathbb{R}$ defined as $f(x) := \sum_{i=1}^{n} f_i(x_i)$ and rewrite (2) in matrix form as

$$\min \ f(x), \quad \text{s.t.} \ A_s x - z = 0, \ A_d x - z = 0. \tag{3}$$

Further define the matrix $A = [A_s; A_d] \in \mathbb{R}^{2mp \times np}$ stacking the arc source and destination matrices A_s and A_d and the

matrix $B = [-I_{mp}; -I_{mp}]$ stacking the opposite of two identity matrices so that (3) reduces to

$$\min \ f(x), \quad \text{s.t. } Ax + Bz = 0. \tag{4}$$

Introduce now Lagrange multipliers $\alpha_e = \alpha_{ij}$ associated with the constraints $x_i = z_{ij}$ and Lagrange multipliers $\beta_e = \beta_{ij}$ associated with the constraints $x_j = z_{ij}$. Group the multipliers α_e in the vector $\alpha = [\alpha_1; \dots; \alpha_m] \in \mathbb{R}^{mp}$ and the multipliers β_e in the vector $\beta = [\beta_1; \dots; \beta_m] \in \mathbb{R}^{mp}$ which are thus associated with the constraints $A_s x - z = 0$ and $A_d x - z = 0$, respectively. Grouping α and β in the multiplier $\lambda = [\alpha; \beta] \in \mathbb{R}^{2mp}$, which is therefore associated with the constraint $Ax + Bz = 0$, we define the augmented Lagrangian of (4) as

$$\mathcal{L}(x, z, \lambda) = f(x) + \lambda^T [Ax + Bz] + \frac{c}{2} \|Ax + Bz\|^2,$$

where $c > 0$ is an arbitrary strictly positive constant.

The ADMM algorithm proceeds through alternating minimizations of the augmented Lagrangian with respect to x and z followed by a gradient ascent update of the Lagrange multiplier. Specifically, introduce iteration index k and let $x(k)$, $z(k)$, and $\lambda(k)$ be variable iterates at time k. At each iteration the augmented Lagrangian is minimized with respect to x and z in an alternating fashion

$$x(k+1) := \operatorname*{argmin}_x \mathcal{L}(x, z(k), \lambda(k)), \tag{5}$$

$$z(k+1) := \operatorname*{argmin}_z \mathcal{L}(x(k+1), z, \lambda(k)). \tag{6}$$

After updating variables $x(k+1)$ and $z(k+1)$ the Lagrange multiplier $\lambda(k+1)$ is updated through the dual ascent iteration

$$\lambda(k+1) = \lambda(k) + c\big[Ax(k+1) + Bz(k+1)\big]. \tag{7}$$

Observing the special structures of the cost function $f(x)$ and the matrices A and B, the iterations in (5)-(7) can be implemented in a decentralized manner. This implementation is the DADMM algorithm [22]–[24].

The computation cost of a DADMM iteration is dominated by the Lagrangian minimization with respect to x in (5). The dual update in (7) requires a few operations per agent because the matrices A and B are as sparse as the graph. The minimization with respect to z in (6) is a simple quadratic minimization that can be solved in a closed form. The minimization in (5), in general, requires implementation of an iterative minimization method. The idea of DLM is to avoid this minimization as we explain in the following.

B. DLM: Decentralized Linearized ADMM

Similar to DADMM, the proposed DLM algorithm also operates with alternating minimizations with respect to x and z followed by a dual ascent step on the multiplier λ. However, instead of minimizing exactly with respect to x we perform an inexact minimization in which the function $f(x)$ is replaced by a quadratic approximation centered at the current iterate. In particular, say that past iterates $x(k)$, $z(k)$ and $\lambda(k)$ are given. Then, the primal iterate $x(k+1)$ is defined as

$$x(k+1) := \operatorname*{argmin}_x \nabla f(x(k))^T (x - x(k)) + \frac{\rho}{2} \|x - x(k)\|^2 + \lambda(k)^T [Ax + Bz(k)] + \frac{c}{2} \|Ax + Bz(k)\|^2, \tag{8}$$

where the approximation parameter ρ is a constant. Comparing the DADMM iteration in (5) with the DLM iteration in (8) we see that the term $\nabla f(x(k))^T \big[x - x(k)\big] + \frac{\rho}{2} \|x - x(k)\|^2$ in the latter is a quadratic approximation of $f(x)$ at point $x(k)$. The steps in (6) and (7) remain unchanged with respect to DADMM. The DLM algorithm is therefore defined by recursive application of (8), (6) and (7).

Using first order optimality conditions for the minimization problems in (8) and (6) yields explicit expressions for $x(k+1)$ and $z(k+1)$. The resulting equation for $x(k+1)$ is

$$\nabla f(x(k)) + \rho\big[x(k+1) - x(k)\big] + A^T \lambda(k) + cA^T \big[Ax(k+1) + Bz(k)\big] = 0, \tag{9}$$

which can be solved explicitly for $x(k+1)$ by inverting the matrix $\rho I_{np} + cA^T A$. Likewise, the first order optimality condition for (6) yields

$$B^T \lambda(k) + cB^T \big[Ax(k+1) + Bz(k+1)\big] = 0, \tag{10}$$

which can be solved for $z(k+1)$ if we invert the matrix $cB^T B$.

By exploiting the sparse structure of $A^T A$ and $B^T B$ it is possible to see that the variable components $x_i(t)$ and $z_{ij}(t)$ can be updated by agent i using its own local iterates and iterates of neighboring agents. Instead of developing that decomposition we first notice that, similar to DADMM, the iterations in (9), (10) and (7) can be replaced by a simpler set of iterations if the variables are properly initialized. Such initialization is adopted henceforth and specified in the following assumption.

Assumption 4 We require the initial Lagrange multiplier $\lambda(0) = [\alpha(0); \beta(0)]$ to satisfy $\alpha(0) = -\beta(0)$ and the initial auxiliary variable $z(0)$ to be such that $E_o x(0) = 2z(0)$. We further define variables $\phi(k) = [\phi_1(k); \dots; \phi_n(k)] := E_o^T \alpha(k) \in \mathbb{R}^{np}$.

If the initialization condition in Assumption 4 holds, the auxiliary variable $z(k)$ can be eliminated and the Lagrange multipliers $\lambda(k) \in \mathbb{R}^{2mp}$ replaced by the smaller dimension vector $\phi(k) \in \mathbb{R}^{np}$. We summarize the simplified algorithm in the following proposition; see [23], [31].

Proposition 1 *Consider the sequence of variables $x(k)$ generated by* (9), (10) *and* (7)*. If Assumption 4 holds, iterates $x(k+1)$ can be alternatively generated by the recursion*

$$x(k+1) = \tilde{D}^{-1} \Big[\tilde{L}_u x(k) - \nabla f(x(k)) - \phi(k)\Big], \tag{11}$$

$$\phi(k+1) = \phi(k) + cL_o x(k+1), \tag{12}$$

where we define the weighted degree matrix $\tilde{D} := 2cD + \rho I_{np}$ and the weighted unoriented Laplacian $\tilde{L}_u = cL_u + \rho I_{np}$.

Proof: See Appendix I. ∎

The initial conditions in Assumption 4 are not difficult to satisfy; e.g., it suffices to set $\alpha_e(0) = \beta_e(0) = 0$ and $x_i(0) = z_e(0) = 0$ for all arcs $e \sim (i, j)$ and agents i. On the other hand, implementation of (11) and (12) does not rely on Assumption 4. Since the iterations in (11) and (12) are equivalent to the iterations in (9), (10) and (7) with

Algorithm 1 DLM algorithm run by agent i

Require: Initialize local variables to $x_i(0)$ and $\phi_i(0) = 0$.

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

2: Compute local solution $x_i(k+1)$ from [cf. (13)]

$$x_i(k+1) = x_i(k) - \frac{1}{\tilde{d}_i}\Big[\nabla f_i(x_i(k)) + c \sum_{j\in\mathcal{N}_i} \big[x_i(k) - x_j(k)\big] + \phi_i(k)\Big],$$

3: Transmit $x_i(k+1)$ / receive $x_j(k+1)$ from neighbors $j \in \mathcal{N}_i$.

4: Update local dual variable $\phi_i(k+1)$ as [cf. (14)]

$$\phi_i(k+1) = \phi_i(k) + c \sum_{j\in\mathcal{N}_i} \big[x_i(k+1) - x_j(k+1)\big].$$

5: **end for**

proper initialization, we implement (11) and (12), not (9), (10) and (7). To implement (11) and (12) and have Proposition 1 hold we just need to make sure that $\phi(0)$ lies in the column space of E_o^T. Further observe that the matrices $\tilde{D}$ and $\tilde{L}_u$ are linear combinations of the degree matrix D and the unoriented Laplacian L_u with identity matrices I_{np}. The coefficients in these linear combinations are $2c$, c, and ρ, which are parameters of DLM.

As is the case of (9), (10) and (7), the operations in (11) and (12) can be implemented in a decentralized manner. Consider the component of the update for $x(k+1)$ corresponding to the variable $x_i(k+1)$. Using the definitions of the weighted degree matrix $\tilde{D}$, the weighted unoriented Laplacian $\tilde{L}_u$, and the oriented incidence matrix E_o, we can write this component of (11) as

$$x_i(k+1) = x_i(k) - \frac{1}{\tilde{d}_i}\Big[\nabla f_i(x_i(k)) + c \sum_{j\in\mathcal{N}_i} \big[x_i(k) - x_j(k)\big] + \phi_i(k)\Big], \tag{13}$$

where we define the weighted degree $\tilde{d}_i := 2cd_i + \rho$ such that $\tilde{D} = \text{diag}(\tilde{d}_1, \ldots, \tilde{d}_n)$. Likewise, using the definition of the oriented Laplacian L_o the update in (12) can be written as

$$\phi_i(k+1) = \phi_i(k) + c \sum_{j\in\mathcal{N}_i} \big[x_i(k+1) - x_j(k+1)\big]. \tag{14}$$

The iterations in (13) and (14) have intuitive appeal. The iteration in (13) is reminiscent of gradient descent with stepsize $1/\tilde{d}_i$. The gradient $\nabla f_i(x_i(k))$ is corrected by the sum $c\sum_{j\in\mathcal{N}_i}\big[x_i(k) - x_j(k)\big]$ which accounts for the disagreement between local variable $x_i(k)$ and neighboring variables $x_j(k)$ and the dual variable $\phi_i(k)$. In turn, the dual variable $\phi_i(k)$ is just an integration device for past disagreements $c\sum_{j\in\mathcal{N}_i}\big[x_i(l+1) - x_j(l+1)\big]$ between local variables $x_i(l)$ and neighboring variables $x_j(l)$ for all times $l \leq k$.

An algorithmic summary of DLM is shown in Algorithm 1. At time $k = 0$ we initialize local variables to arbitrary $x_i(0)$ and $\phi_i(0) = 0$. The latter is one out of many possible selections to ensure that the vector $\phi(0)$ is in the column space of E_0^T. For all subsequent times agent i goes through successive steps implementing the primal and dual iterations in (13) and (14) as shown in Step 2 and Step 4 of Algorithm 1, respectively. Implementation of Step 2 requires neighboring variables $x_j(k)$ from the previous iteration. Implementation of Step 4 requires current neighboring variables $x_j(k+1)$, which become available through the exchange implemented in Step 3. This variable exchange also makes variables available for the update in Step 2 corresponding to the following time index.

We proceed to analyze the convergence properties of DLM after two pertinent remarks.

Remark 1 As intended, DLM is advantageous over DADMM due to its lower computation burden. The iterations in (13) and (14) contain simple algebraic operations and a gradient descent step. The counterpart of (13) in DADMM is a, most often nontrivial, minimization of $f_i(x_i)$ augmented by a quadratic term. It is also interesting to compare DLM and DGM [13]. In the latter, agent i updates its local variable $x_i(k)$ as

$$x_i(k+1) = \sum_{j\in\mathcal{N}_i\cup i} w_{ij}x_j(k) - \epsilon(k)\nabla f_i(x_i(k)), \tag{15}$$

where $\epsilon(k)$ is a stepsize sequence that can be chosen as constant or nonsummable vanishing and the weights w_{ij} are elements of a doubly stochastic matrix $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ [32]. The idea of the update in (15) is to descent along the negative gradient direction $-\nabla f_i(x_i(k))$ while mixing local and neighboring iterates. This idea can be also construed as an interpretation of (13) with the difference being the addition of the memory term $\phi_i(k)$. In that sense we can think of DLM as a primal method with memory – as opposed to a dual method with inexact Lagrangian minimization. Irrespective of interpretation, the computation cost of DLM is of the same order as DGM.

Remark 2 DLM differs from the centralized linearized ADMM in [25], [26] in that the latter linearizes the quadratic term $(c/2)\|Ax + Bz\|^2$ in the augmented Lagrangian in (5) – while DLM linearizes the objective function $f(x)$. The centralized linearized ADMM in [27] applies to objectives of the form $f(x) + g(z)$ and uses linearized versions of these functions in both, the $x(k)$ iteration in (5) and the $z(k)$ iteration in (6). This, however, yields an algorithm that is not guaranteed to converge to optimal arguments. An extra gradient step is added to overcome this limitation. The special structure of the cost function in (4), namely, that the objective does not contain a term of the form $g(z)$ precludes this problem. This yields the simpler linearized algorithm defined by (9), (10) and (7) which we transform into the DLM algorithm defined by (11) and (12).

III. Convergence Analysis

This section analyzes convergence and rate of convergence of the proposed DLM algorithm. We analyze the iterations (9), (10) and (7), instead of the iterations (11) and (12) (see also Algorithm 1). Recall that they are equivalent under Assumption 4 as shown in Proposition 1. Section III-A provides basic assumptions and supporting lemmas. Section III-B proves convergence of DLM, while Section III-C establishes a linear rate of convergence under stronger assumptions.

A. Preliminaries

The analyses of convergence and rate of convergence are based on Assumptions 1–3. Assumption 1 implies that the aggregate function $f(x) := \sum_{i=1}^{n} f_i(x_i)$ is proper closed convex and differentiable. Assumption 2 implies that the aggregate function $f(x)$ has Lipschitz continuous gradients with constant M_f. For any pair of points x_a and x_b it holds

$$\|\nabla f(x_a) - \nabla f(x_b)\| \leq M_f \|x_a - x_b\|. \tag{16}$$

Assumption 3 implies that the aggregate function $f(x)$ is strongly convex with constant m_f. For any pair of points x_a and x_b it holds

$$\big[x_a - x_b\big]^T \big[\nabla f(x_a) - \nabla f(x_b)\big] \geq m_f \|x_a - x_b\|^2. \tag{17}$$

Assumptions 1 and 2 are common in proving convergence of descent algorithms. Assumption 3 is also a common requirement to establish linear convergence rates.

We investigate convergence the primal variable $x(k)$ and the dual variable $\alpha(k)$, which is a part of the Lagrange multiplier $\lambda(k) = [\alpha(k); \beta(k)]$, to their optimal values. Observe that due to the consensus constraints, an optimal primal solution has the form $x^* = [\tilde{x}^*; \cdots; \tilde{x}^*]$ where $\tilde{x}^*$ is an optimal solution of (1). If the local cost functions are not strongly convex, then there may exist multiple optimal primal solutions; instead, if the local cost functions are strongly convex (i.e., Assumption 3 holds), the optimal primal solution is unique. For each optimal primal solution x^*, there exist multiple optimal Lagrange multipliers $\lambda^* = [\alpha^*; \beta^*]$ where $\alpha^* = -\beta^*$ as we will prove in Lemma 1. In the analysis of convergence (Section III-B), we show that $\alpha(k)$ converges to one of the optimal dual solutions α^* whose value depends on the initial dual variable $\alpha(0)$. In establishing a linear rate of convergence (Section III-C), we require that the dual variable is initialized such that $\alpha(0)$ lies in the column space of E_o and consider its convergence to a unique dual solution α^* that also lies in the column space of E_o. Existence and uniqueness of such an α^* are also proved in Lemma 1 as we state next.

Lemma 1 *Given an optimal primal solution x^* of* (4)*, there exist multiple optimal multipliers $\lambda^* = [\alpha^*; \beta^*]$ where $\alpha^* = -\beta^*$ such that every (x^*, λ^*) is a primal-dual optimal pair. Among all these optimal duals λ^*, there exists a unique $\lambda^* = [\alpha^*; \beta^*]$ such that $\alpha^* = -\beta^*$ lies in the column space of E_o.*

Proof: See Appendix II. ■

In the subsequent analyses of convergence and rate of convergence, we need a couple of equalities that connect $x(k+1)$, $x(k)$, $\alpha(k+1)$ and $\alpha(k)$ with a pair of optimal primal and dual solutions x^* and α^*. These equalities are technical and provided in the following lemma.

Lemma 2 *Consider iterations* (9), (10) *and* (7) *initialized as in Assumption 4. Let x^* and α^* be optimal for* (4) *and recall the definition $\tilde{L}_u = cL_u + \rho I_{np}$. Then, for all times $k \geq 0$, we write the gradient difference $\nabla f(x(k)) - \nabla f(x^*)$ as*

$$\begin{aligned} &\nabla f(x(k)) - \nabla f(x^*) \\ &\qquad = \tilde{L}_u \big[x(k) - x(k+1)\big] - E_o^T \big[\alpha(k+1) - \alpha^*\big], \end{aligned} \tag{18}$$

Likewise, the primal variable difference $x(k+1) - x^*$ satisfies

$$\frac{c}{2} E_o \big[x(k+1) - x^*\big] = \alpha(k+1) - \alpha(k). \tag{19}$$

Proof: See Appendix III. ■

B. Convergence

To prove convergence of DLM iterates to an optimal pair x^* and α^* of (4) we show that the primal and dual variables $x(k+1)$ and $\alpha(k+1)$ are closer to x^* and α^* than the previous iterates $x(k)$ and $\alpha(k)$. More to the point, for a given optimal pair (x^*, α^*) define the energy function

$$V_{x^*,\alpha^*}(x, \alpha) := \frac{1}{2} \big\|x - x^*\big\|_{\tilde{L}_u}^2 + \frac{1}{c} \big\|\alpha - \alpha^*\big\|^2. \tag{20}$$

Recall that the positive definite matrix $\tilde{L}_u$ is defined as $\tilde{L}_u = cL_u + \rho I_{np}$. We show that the energy function $V_{x^*,\alpha^*}(k) = V_{x^*,\alpha^*}(x(k), \alpha(k))$ is monotonically decreasing with an improvement that is related to the squared distance between the two successive points $(x(k+1), \alpha(k+1))$ and $(x(k), \alpha(k))$ as we formally state next.

Lemma 3 *Consider iterations* (9), (10) *and* (7) *with the initial conditions in Assumption 4. With γ_u denoting the smallest eigenvalue of the unoriented Laplacian L_u and M_f the Lipschitz continuity constant of the local cost functions' gradients define the constant*

$$\xi := (c\gamma_u + \rho - M_f/2)/(c\gamma_u + \rho). \tag{21}$$

Assume that the DLM parameters c and ρ are chosen so that $\xi > 0$ and that Assumptions 1 and 2 hold. Then, for all times $k \geq 0$ we have that the energy function $V_{x^*,\alpha^*}(k) = V_{x^*,\alpha^*}(x(k), \alpha(k))$ in (20) *is monotonically decreasing and satisfies*

$$\begin{aligned} V_{x^*,\alpha^*}(k+1) \;\leq\; & V_{x^*,\alpha^*}(k) \\ & - \frac{\xi}{2} \big\|x(k+1) - x(k)\big\|_{\tilde{L}_u}^2 - \frac{\xi}{c} \big\|\alpha(k+1) - \alpha(k)\big\|^2. \end{aligned} \tag{22}$$

Proof: See Appendix IV. ■

To guarantee the condition $\xi > 0$ we just need to make c or ρ sufficiently large. For future reference further note that we must have $\xi < 1$ for any choice of DLM parameters c and ρ.

Since the energy function $V_{x^*,\alpha^*}(k)$ is monotonically decreasing and nonnegative, Lemma 3 implies that it must eventually converge. To prove convergence of the sequence $x(k)$ to an optimal solution x^* we need to show not only that the energy function $V_{x^*,\alpha^*}(k)$ converges but that it converges to zero for some optimal pair (x^*, α^*). Constructing this argument is not difficult if we follow analogous proofs for the centralized ADMM; see e.g., [26], [33]. With particular note, recently [34] also proves convergence of the centralized ADMM for nonconvex sharing and consensus problems. We give the convergence result in the following theorem.

Theorem 1 *Consider iterations* (9), (10) *and* (7) *with the initial conditions in Assumption 4. Let Assumptions 1 and 2 hold and the constant ξ in* (21) *be positive. Then, the sequences*

$x(k)$ and $\alpha(k)$ generated by the DLM algorithm converge to an optimal pair of (4)*. I.e., there exist optimal x^* and $\lambda^* = [\alpha^*; \beta^*]$ such that*

$$\lim_{k\to\infty} x(k) = x^* \quad \textit{and} \quad \lim_{k\to\infty} \alpha(k) = \alpha^*. \tag{23}$$

Proof: See Appendix V. ■

We emphasize that Theorem 1 does not specify which optimal primal and dual solutions $x(k)$ and $\alpha(k)$ converge to. Indeed, $x(k)$ can converge to one of the optimal primal solutions x^* and $\alpha(k)$ can converge to one of the corresponding optimal dual solutions. However, if let $\alpha(0)$ be initialized in the column space of E_o, from the dual iterate $\alpha(k+1) = \alpha(k) + (c/2)E_o x(k+1)$ in (37), we know that $\alpha(k)$ always lies in the column space of E_o for all times $k \geq 0$. Therefore, $\alpha(k)$ converges to the unique optimal dual solution that corresponds to x^* and lies in the column space of E_o; existence and uniqueness of such a dual solution have been proved in Lemma 1.

Note that we need $c > 0$ to have a proper energy function in (22) but that it is possible to have $\rho < 0$ without violating the hypotheses of Theorem 1. If we choose a negative ρ we just need to make c larger to guarantee $\xi > 0$ in (21). As long as $\xi > 0$, DLM converges to a pair of optimal (x^*, α^*), which shows robustness of DLM to the parameters c and ρ. It implies that the cost function of (8), which is the Taylor expansion of the augmented Lagrangian in (5), must have a positive definite Hessian. The possibility of selecting negative ρ notwithstanding, our numerical analyses suggest that fastest convergence is achieved with a constant c that is slightly larger than the equivalent constant used in DADMM [24] and positive constant ρ of similar order to c – see Section IV.

C. Linear Rate of Convergence

If we add the strong convexity condition in Assumption 3 to the hypotheses in Theorem 1 we can establish a linear rate of convergence for DLM. To do so we use the strong convexity of the local cost functions f_i to develop a contraction inequality for the energy function $V_{x^*,\alpha^*}(x,\alpha)$ in (20) for a properly selected optimal pair (x^*, α^*) and a properly selected initial multiplier $\alpha(0)$. The particular optimal pair is formed by the unique optimal primal argument x^* – which is unique because the local cost functions f_i are strongly convex – and the unique dual optimal solution lying in the column space of E_o – which is unique because we prove so in Lemma 1. The initial multiplier $\alpha(0)$ must be selected in the column space of E_0. This is needed because our analysis holds in the column space of E_0 and selecting $\alpha(0)$ in that space ensures that $\alpha(k)$ stays in it for all times k. We emphasize that this setting is different to Lemma 3 that holds for any pair of optimal primal and dual solutions and any initial condition that satisfies Assumption 4. We present this result in the following theorem.

Theorem 2 *Consider iterations* (9), (10) *and* (7) *with the initial condition in Assumption 4 and the further requirement that $\alpha(0)$ lies in the column space of E_o. Further assume that assumptions 1-3 hold, that the constant in* (21) *is $\xi > 0$, and let x^* and α^* be the unique optimal pair of* (4) *for which α^* lies in the column space of E_o. Then, there exists a contraction parameter $\delta > 0$ such the energy function in* (20) *satisfies*

$$V_{x^*,\alpha^*}(k+1) \;\leq\; \frac{1}{1+\delta} V_{x^*,\alpha^*}(k) \tag{24}$$

Proof: See Appendix VI. ■

The constant δ has an explicit expression in terms of the unoriented Laplacian eigenvalues Γ_u and γ_u, the oriented Laplacian eigenvalue γ_o, the strong convexity and Lipschitz constants m_f and M_f, and the DLM parameters c and ρ – see (68) in Appendix VI. The result in Theorem 2 is analogous to similar results that hold for ADMM [35] and DADMM [24]. The result is also related to the linear convergence results of the centralized multi-block ADMM [38], the centralized ADMM on quadratic problems [40], as well as the asymptotic linear convergence rate of DADMM [39].

In the contraction inequality in (24) we have that $V_{x^*,\alpha^*}(k)$ shrinks by a factor strictly smaller than 1 at all iterations. Therefore, Theorem 2 indicates linear convergence of $V_{x^*,\alpha^*}(k)$ to 0. Since $V_{x^*,\alpha^*}(k) = (1/2)\|x(k) - x^*\|^2_{\tilde{L}_u} + (1/c)\|\alpha(k) - \alpha^*\|^2$, it follows that $\|x(k) - x^*\|^2_{\tilde{L}_u}$ also converges linearly to 0 because we have that

$$\begin{aligned} \|x(k) - x^*\|^2_{\tilde{L}_u} &\leq V_{x^*,\alpha^*}(k+1) \\ &\leq \left(\frac{1}{1+\delta}\right)^{k+1} V_{x^*,\alpha^*}(0). \end{aligned} \tag{25}$$

To establish that $\|x(k) - x^*\|$ converges linearly to 0 it suffices to write the conventional Euclidean norm $\|x(k) - x^*\|$ in terms of the $\tilde{L}_u$ norm $\|x(k) - x^*\|_{\tilde{L}_u}$ and take the square root of both sides of (25). Substituting the inequality $\|x(k) - x^*\|^2_{\tilde{L}_u} \geq (c\gamma_u + \rho)\|x(k) - x^*\|^2$ into (25), we have the following corollary of Theorem 2.

Corollary 1 *Consider iterations* (9), (10) *and* (7) *and assume the same hypotheses of Theorem 2. Then, there exists a contraction parameter $\delta > 0$ such that*

$$\|x(k) - x^*\| \;\leq\; \left(\frac{1}{\sqrt{1+\delta}}\right)^{k+1} \left(\frac{V_{x^*,\alpha^*}(0)}{c\gamma_u + \rho}\right)^{1/2}. \tag{26}$$

I.e., the primal variable $x(k)$ converges linearly to the unique optimal primal variable x^*.

Corollary 1 shows that $\|x(k) - x^*\|$ linearly converges to 0 if the initial energy function $V_{x^*,\alpha^*}(0)$ is finite and the weighted Laplacian $\tilde{L}_u = cL_u + \rho I_{np}$ is positive definite. Note that $\|x(k) - x^*\|$ is not necessarily monotonically decreasing as $V_{x^*,\alpha^*}(k)$ is (see Theorem 2).

Remark 3 In Theorem 2 and Corollary 1, we require that $\alpha(0)$ lies in the column space of E_o in addition to the initial condition in Assumption 4. Translating the initial condition of $\alpha(0)$ to that of $\phi(0)$ in Algorithm 1, we can see that the initial Lagrange multiplier $\phi(0)$ determines where the dual solution converges (Section III-B). To achieve linear rate of convergence, $\phi(0)$ must be chosen in the column space of L_o (e.g., $\phi(0) = 0$) because $\phi(0) = E_o^T \alpha(0)$ and $L_o =$

$(1/2)E_o^T E_o$ (Section III-C). This is equivalent to choosing $\lambda(0) = [\alpha(0); \beta(0)]$ such that both $\alpha(0)$ and $\beta(0)$ are in the column space of E_o.

IV. Numerical Experiments

This section provides numerical experiments to study the convergence times of DLM as defined by Algorithm 1 for a least squares problem (Section IV-A) and a logistic regression problem (Section IV-B). The local cost functions in the least squares problem are strongly convex whereas the local functions in the logistic regression example are convex but not strongly convex. We consider various network topologies – random, line, star, complete, and small world graphs – as well as the effect of growing the number of agents in the network. We also compare the performance of DLM with that of the decentralized ADMM (DADMM) of [22] as defined by (5)-(7) and the distributed gradient method (DGM) of [13] as defined by (15), and an accelerated variant of DGM, the distributed Nesterov gradient (DNG) method of [16], as defined by

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup i} w_{ij} y_j(k) - \epsilon(k) \nabla f_i(y_i(k)),$$
$$y_i(k+1) = x_i(k+1) + \eta(k) \left(x_i(k+1) - x_i(k) \right),$$

where $\eta(k) = (k-1)/(k+2)$ is the parameter of Nesterov acceleration. In DGM and DNG, the weight matrix W is chosen following the maximum-degree rule [32]. Convergence is studied in terms of the average absolute error

$$e(k) := \frac{1}{n} \sum_{i=1}^{n} \|x_i(k) - \tilde{x}^*\|.$$

The average absolute error $e(k)$ is the average of the local errors $\|x_i(k) - \tilde{x}^*\|$ observed at each agent i.

A. Least Squares Regression

Agent i measures a true signal $\tilde{x}^o \in \mathbb{R}^3$ through the noisy linear transformation $y_i = U_i \tilde{x}^o + \omega_i$ where $U_i \in \mathbb{R}^{3\times 3}$ is the measurement matrix and $\omega_i \in \mathbb{R}^3$ is the noise vector. To run global least squares regression taking advantage of the information collected by all agents we formulate a problem as in (1) with the local cost function of agent i given by $f_i(\tilde{x}) = \|U_i \tilde{x} - y_i\|^2/2$. With this particular choice of functions the iteration in (13) and, equivalently, Step 2 of Algorithm 1, becomes

$$x_i(k+1) = x_i(k) \qquad (27)$$
$$- \frac{1}{\tilde{d}_i} \left[U_i^T U_i x_i(k) - U_i^T y_i + c \sum_{j \in \mathcal{N}_i} \left[x_i(k) - x_j(k) \right] + \phi_i(k) \right].$$

The iteration in (14) and Step 4 of Algorithm 1 is independent of the specific form of $f_i(\tilde{x})$. Elements of the matrix U_i are chosen at random from a normal distribution with zero mean and variance 1. Matrices U_i are checked for invertibility by requiring $U_i^T U_i \succeq 10^{-4} \times I_3$ so that the local functions are strongly convex with strong convexity parameter $m_f = 10^{-4}$. A different U_i matrix is chosen if this is not satisfied. The noise vectors $\omega_i \in \mathbb{R}^3$ follow from a zero-mean Gaussian distribution with covariance matrix $\mathbb{E}\left[\omega_i \omega_i^T\right] = 10^{-2} \times I_3$.

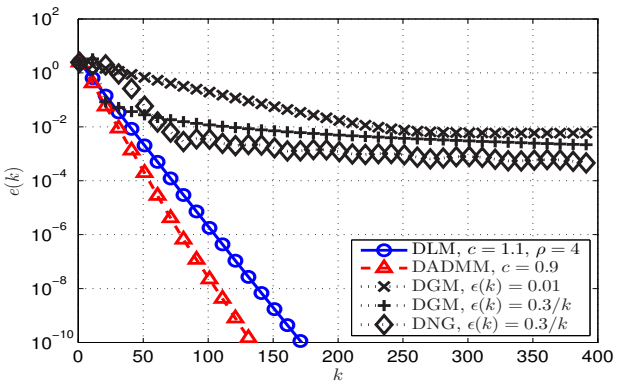

Fig. 1. Comparison of different decentralized optimization algorithms in a least squares problem. DLM, DADMM, DNG, and DGM with constant and vanishing stepsizes are shown for a random network with $n = 100$ agents and 384 edges. DLM has a slope close to DADMM but a much smaller computation cost per iteration. DLM has similar computation cost per iteration as DNG and DGM but converges much faster.

Algorithm comparison. Given $n = 100$ agents select bidirectional edges at random until obtaining a connected network. An example run of DLM, DADMM, DNG, and DGM with constant and vanishing stepsizes are shown for one such network in Fig. 1. In this example the network contains 384 edges ($m = 768$ arcs) out of the 4950 possible. Different parameter combinations are chosen for each algorithm and results are reported for the one that minimizes the average absolute error $e(k)$ in (27) after $k = 100$ iterations. These parameters are $c = 1.1$ and $\rho = 4$ for DLM, $c = 0.9$ for DADMM, $\epsilon(k) = 0.3/k$ for DNG, $\epsilon(k) = 0.01$ for DGM with constant stepsize and $\epsilon(k) = 0.3/k$ for DGM with decreasing stepsize. Note that for least squares regression, local optimization of DADMM boils down to a matrix inversion. The convergence rate of DLM is linear as proven in Section III. More interesting, the difference in the slopes of DLM and DADMM are minimal. The latter requires between 20% to 30% less iterations to achieve a target accuracy. This penalty in convergence rate is small given that the computation cost of each DLM iteration involves $O(p)$ operations (cf. steps 2 and 4 of Algorithm 1) – in this experiment $p = 3$ – whereas DADMM requires solution of local optimization problems at each iteration [22]. Further observe that DLM converges much faster than DNG and DGM. This is consistent with earlier theoretical and numerical comparisons of DADMM, DNG, and DGM [15], [16], [24].

Network topology. The slope of the convergence curve of DLM varies with the choice of network topology. Convergence curves for line, star, complete, and small world topologies are shown in figs. 2 and 3. The random small world topologies are constructed through first forming a cycle topology and then adding random edges. In all cases we consider $n = 100$ agents and choose parameters c and ρ in Algorithm 1 to minimize the average absolute error $e(k)$ in (27) after $k = 100$ iterations.

As seen in Fig. 2 the fastest and slowest convergence are

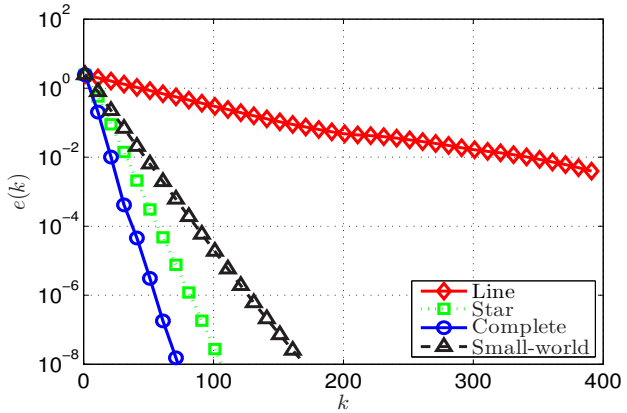

Fig. 2. Convergence of DLM on line, star, complete, and small world graphs. All networks have $n = 100$ agents. The small world graph is formed by a cycle to which 100 extra random edges are added. Convergence is slowest for the line and fastest for the complete graph. The star graph is good at diffusing information with small average degree but large maximum degree. Small world graphs diffuse information efficiently with small maximum degree.

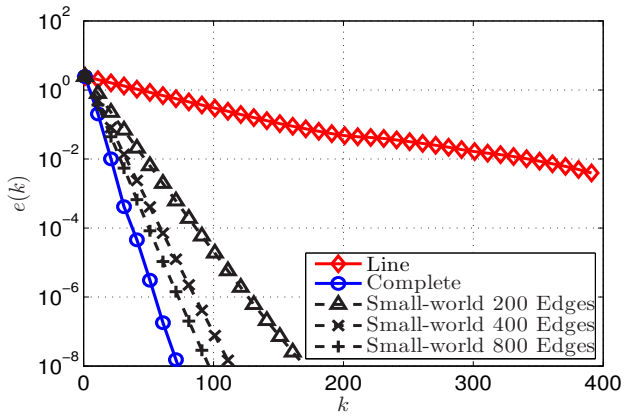

Fig. 3. Convergence of DLM on small world networks with different average degree. Convergence slopes for networks with $n = 100$ agents and 100, 400, or 700 random edges added to the cycle are shown. Convergence of the line and complete graphs are also depicted for reference. Adding more random edges to small world networks increases the agents' average degree but expedites convergence.

exhibited by complete (with $c = 0.05$ and $\rho = 3$) and line graphs (with $c = 30$ and $\rho = 8$), respectively. This is reasonable because these are the graphs for which it takes the longest and shortest time possible for the observations of one agent to affect all other agents. The faster convergence rate of complete graphs comes at the expense of communication cost. Agents in the line graph exchange information with one or two neighbors only, whereas in the complete graph each agent communicates with all other $n - 1$ agents. Small communication cost and steep convergence slope are achieved by the star topology also shown in Fig. 2 (with $c = 3.6$ and $\rho = 5$). While aggregate communication cost is small for star topologies, the center agent is a communication bottleneck. A structure that avoids this problem is a small world network

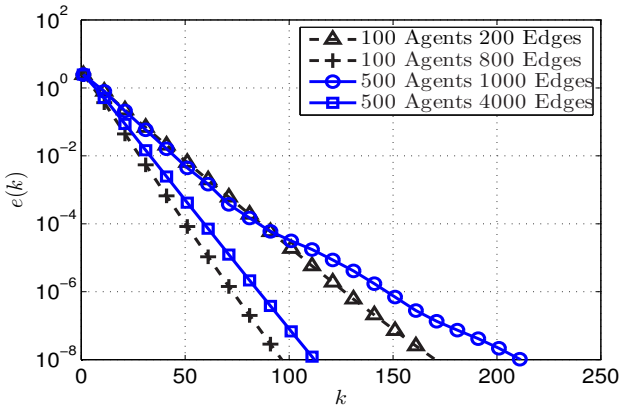

Fig. 4. Convergence of DLM for small world networks of different sizes and average degrees. Networks with $n = 100$ and $n = 500$ agents are shown. In each case we add 1 or 7 random edges per agent. The convergence rate of DLM is more sensitive to the average degree than to the network size.

formed by a cycle plus a given number of random edges. The convergence behavior for a small world network with 100 random edges (200 edges in total) is also shown in Fig. 2 (with $c = 2.8$ and $\rho = 2$). Each agent in this network communicates with an average of 4 neighbors. Convergence slope improves substantially over the line graph while avoiding the excessive communication cost of a complete graph or the bottleneck of the star topology. This is, again, not surprising. Small world networks are good at diffusing information with small degrees because the addition of random edges decreases the network's diameter.

The convergence slope of small world networks depends on the number of random edges added. In Fig. 3 we show convergence curves when we add 100, 300, and 700 random edges. This corresponds to networks whose average degrees are 4, 8, and 16, respectively; the parameters are $c = 2.8$ and $\rho = 2$, $c = 1$ and $\rho = 2$, and $c = 0.4$ and $\rho = 3$, respectively. The curves corresponding to line and complete graphs are also shown for reference. Adding random edges to small world networks increases the agents' average degree and expedites convergence. Observe that to reduce communication cost needed to achieve a target accuracy, there is a tradeoff in setting the average degree. Increasing the average degree requires higher communication cost per iteration, but its gain in expediting convergence becomes marginal when the network is dense enough. Indeed, when the average degree is 16 – in this case the network has 100 deterministic edges and 700 random edges – the speed of convergence is close to that of a complete graph, whose average degree is 99.

Scalability. The experiments above demonstrate a strong dependence of the convergence rate of DLM with the network topology. Here we show that the convergence rate is less dependent of the network size using small world networks as a test case. For that matter we consider connected small world networks composed of: 1) $n = 100$ agents with 100 deterministic edges and 100 random edges; 2) $n = 100$ agents with 100 deterministic edges and 700 random edges;

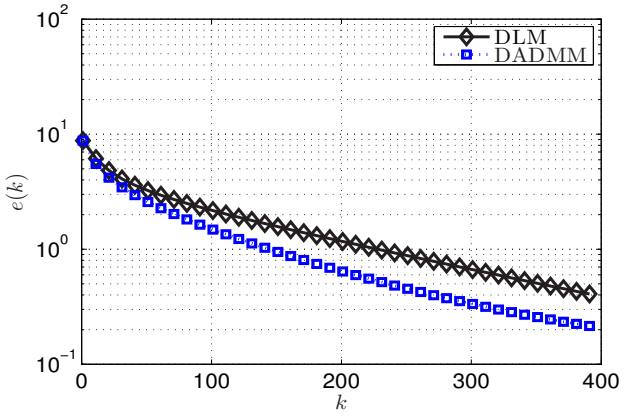

Fig. 5. Comparison of DLM and DADMM on the logistic regression problem. In a random small world network composed of $n = 100$ agents with 100 deterministic edges and 100 random edges, each agent i has $q_i = 50$ samples and each sample is of $p = 100$ dimension. DLM requires between 20% to 30% more iterations to achieve a target accuracy than DADMM, but on average, each DADMM iteration requires 80 gradient descent iterations. It follows that in terms of computation cost DADMM is about 60 times more expensive than DLM.

3) $n = 500$ agents with 500 deterministic edges and 500 random edges; 4) $n = 500$ agents with 500 deterministic edges and 3500 random edges. The corresponding optimal DLM parameters are 1) $c = 2.8$ and $\rho = 2$; 2) $c = 0.4$ and $\rho = 3$; 3) $c = 3.2$ and $\rho = 2$; 4) $c = 0.6$ and $\rho = 3$. Fig. 4 shows that convergence rates of DLM are similar for the networks with the same average degree, which means similar network connectedness, rather than the same network size. Also observe that the performance of DLM does not degrade much when the network size n increases.

B. Logistic Regression

We consider application of DLM to a logistic regression problem. Agent i has access to q_i sample vectors and corresponding classes. Denote the sample vectors as $u_{il} \in \mathbb{R}^p$ and the corresponding classes as $y_{il} \in \{-1, 1\}$ with $l = 1, \ldots, q_i$. We are interested in observing samples $u \in \mathbb{R}^p$ and estimating the probability $\mathrm{P}\left[y = 1 \,\middle|\, u\right]$ of observing class $y = 1$. We postulate that this probability is given by the logistic function $\mathrm{P}\left[y = 1 \,\middle|\, u\right] = 1/(1 + \exp(-u^T \tilde{x}))$ for some vector $\tilde{x}$ to be determined using the sample and class observation pairs $\{u_{il}, y_{il}\}_{i,l}$ available at *all* agents. Given this model it follows that the maximum likelihood estimate of the vector $\tilde{x}$ is given by – see, e.g., [36] –

$$\tilde{x}^* := \operatorname*{argmin}_{\tilde{x}} \sum_{i=1}^{n} \sum_{l=1}^{q_i} \log\Big(1 + \exp\big(-y_{il} u_{il}^T \tilde{x}\big)\Big). \tag{28}$$

This problem has the form in (1) with the local functions defined as

$$f_i(\tilde{x}) = \sum_{l=1}^{q_i} \log\Big(1 + \exp\big(-y_{il} u_{il}^T \tilde{x}\big)\Big). \tag{29}$$

For this specific choice of functions $f_i(\tilde{x})$ the iteration in (13) and Step 2 of Algorithm 1 becomes

$$x_i(k+1) = x_i(k) - \frac{1}{\tilde{d}_i}\Big[c \sum_{j \in \mathcal{N}_i} \big[x_i(k) - x_j(k)\big] + \phi_i(k)\Big] + \frac{1}{\tilde{d}_i} \sum_{l=1}^{q_i} \frac{y_{il} u_{il} \exp\big(-y_{il} u_{il}^T x_i(k)\big)}{1 + \exp\big(-y_{il} u_{il}^T x_i(k)\big)}. \tag{30}$$

The iteration in (14) and Step 4 of Algorithm 1 is independent of the specific form of $f_i(\tilde{x})$. Observe that the local cost functions $f_i(\tilde{x})$ are convex but not strongly convex. Thus, the linear convergence guarantees of Section III do not hold. The numerical results show that, nonetheless, DLM succeeds in finding $\tilde{x}^*$ and does so with a performance very close to the performance of DADMM – and a much smaller computational cost comparable to that of DGM.

As a particular numerical example consider a random small world network composed of $n = 100$ agents with 100 deterministic edges and 100 random edges. Each agent i has $q_i = 50$ samples and each sample is of dimension $p = 100$. Different from the least squares regression in Section IV-A, DADMM minimizations required at each step cannot be computed in closed form. We solve these minimizations through a local gradient descent algorithm with stepsize 0.01. We terminate the local gradient descent when the Euclidean norm between two successive solutions is less than 10^{-4}. For implementation of DADMM and DLM we use the parameters that minimize the absolute error $e(k)$ in (27) after running $k = 400$ iterations. These parameters are $c = 1.4$ for DADMM and $c = 3$ and $\rho = 4$ for DLM.

The results are shown in Fig. 5. Both algorithms converge towards $\tilde{x}^*$, but none of them converges linearly. The number of iterations required by DLM to achieve a target accuracy is larger than those required by DADMM. The difference is minimal, however. This small increase in the number of iterations results in a large reduction in the computation cost of each iteration. Each DLM step requires computation of the update in (27). Each DADMM iteration requires computation of gradient descent steps that are numerically analogous to the DLM step in (27). On average, each DADMM iteration in Fig. 5 requires 80 gradient descent iterations. As DLM requires between 20% to 30% more iterations to achieve a target accuracy than DADMM, it follows that in terms of computation cost DADMM is about 60 times more expensive than DLM.

V. Conclusion

We introduced DLM, a decentralized version of the linearized alternating direction method of multipliers to solve optimization problems with separable objectives. The method is a variation of the decentralized alternating direction method of multipliers (DADMM). The main difference is that instead of performing a minimization step in the primal domain, an objective cost linearization is used to yield a step whose computational cost is akin to that of a gradient descent step. This modification results in DLM having a computational cost per iteration that is one to two orders of magnitude smaller than the cost of DADMM. The algorithm was proven to converge

to optimal arguments at a linear rate when the local objective functions have Lipschitz continuous gradients and are strongly convex. Numerical experiments were conducted for a least squares problems as well as for a logistic regression problem. In both cases the number of iterations required by DLM to achieve a target accuracy are of the same order of those required by DADMM. Besides having a much smaller total computational cost than DADMM, DLM also outperforms the distributed gradient method (DGM) and its accelerated variant.

APPENDIX I
PROOF OF PROPOSITION 1

Proof: The proof is analogous to similar results in [23], [31] and given here for completeness. Substituting the multiplier update $\lambda(k) = \lambda(k+1) - c\big[Ax(k+1) + Bz(k+1)\big]$ in (7) into the update for the primal variables $x(k+1)$ in (9) leads to

$$\begin{aligned}&\nabla f(x(k)) + \rho\big[x(k+1) - x(k)\big] \\ &+ A^T\lambda(k+1) - cA^TB\big[z(k+1) - z(k)\big] = 0.\end{aligned} \tag{31}$$

Similarly, substituting the multiplier update $\lambda(k) = \lambda(k+1) - c\big[Ax(k+1) + Bz(k+1)\big]$ in (7) into the expression for the auxiliary variable $z(k+1)$ in (10) leads to

$$B^T\lambda(k+1) = 0. \tag{32}$$

Recalling the definitions of $B = [-I_{mp}; -I_{mp}]$ and $\lambda(k+1) = [\alpha(k+1); \beta(k+1)]$ it follows from (32) that $\alpha(k+1) = -\beta(k+1)$ for all $k \geq 0$. Since we have $\alpha(0) = -\beta(0)$ by hypothesis, it follows that $\alpha(k) = -\beta(k)$ for all $k \geq 0$. Using this fact, the definition of $A = [A_s; A_d]$, and the definition of the oriented incidence matrix $E_o = A_s - A_d$, we conclude that for all $k \geq 0$

$$A^T\lambda(k) = A_s^T\alpha(k) - A_d^T\alpha(k) = E_o^T\alpha(k). \tag{33}$$

Further observe that from the definitions of $A = [A_s; A_d]$, $B = [-I_{mp}; -I_{mp}]$ and the unoriented incidence matrix $E_u = A_s + A_d$, it follows that $A^TB = [A_s^T, A_d^T][-I_{mp}; -I_{mp}] = -A_s^T - A_d^T = -E_u^T$. Substituting this expression and (33) into (31) yields

$$\begin{aligned}&\nabla f(x(k)) + \rho\big[x(k+1) - x(k)\big] \\ &+ E_o^T\alpha(k+1) + cE_u^T\big[z(k+1) - z(k)\big] = 0.\end{aligned} \tag{34}$$

Now consider (7) and recall that $\lambda(k) = [\alpha(k); \beta(k)]$ to separate the equality along the $\alpha(k)$ and $\beta(k)$ directions

$$\begin{aligned}\alpha(k+1) &= \alpha(k) + c\big[A_sx(k+1) - z(k+1)\big], \\ \beta(k+1) &= \beta(k) + c\big[A_dx(k+1) - z(k+1)\big].\end{aligned} \tag{35}$$

Since we know from (32) and the initialization hypothesis that $\alpha(k) = -\beta(k)$ for all $k \geq 0$ we can sum up the two equalities in (35) to obtain $c\big[A_sx(k) - z(k)\big] + c\big[A_dx(k) - z(k)\big] = 0$ for all $k > 0$. Reorder terms to write

$$\frac{1}{2}E_ux(k) = \frac{1}{2}(A_s + A_d)x(k) = z(k), \tag{36}$$

where we also use the definition of the unoriented edge incidence matrix $E_u = A_s + A_d$ to write the first equality. Since by initialization hypothesis $(1/2)E_ux(0) = z(0)$, (36) is true for all times $k \geq 0$.

Using (36) to eliminate $z(k)$ from the update for $\alpha(k)$ in (35) yields

$$\begin{aligned}\alpha(k+1) &= \alpha(k) + c\big[A_sx(k+1) - \frac{1}{2}(A_s + A_d)x(k+1)\big] \\ &= \alpha(k) + \frac{c}{2}E_ox(k+1).\end{aligned} \tag{37}$$

Here we use the definition of the oriented edge incidence matrix $E_o = A_s - A_d$. Multiplying both sides of (37) by E_o^T and using the definitions of the oriented Laplacian matrix $L_o = (1/2)E_o^TE_o$ and the vector $\phi(k) = E_o^T\alpha(k)$, we obtain the update for $\phi(k+1)$ in (12). Likewise, use (36) to eliminate $z(k+1)$ and $z(k)$ from (34) so as to write

$$\begin{aligned}&\nabla f(x(k)) + \Big(\frac{c}{2}E_u^TE_u + \rho I_{np}\Big)\big[x(k+1) - x(k)\big] \\ &+ E_o^T\alpha(k+1) = 0.\end{aligned} \tag{38}$$

From the definition $E_o^T\alpha(k+1) = \phi(k+1)$ and the equality $\phi(k+1) = \phi(k) + L_ox(k+1)$ in (12), we know that $E_o^T\alpha(k+1) = \phi(k) + L_ox(k+1)$. Using this equality and the definition of the unoriented Laplacian $L_u = (1/2)E_u^TE_u$, rewrite (38) to

$$\begin{aligned}&\nabla f(x(k)) + (cL_u + \rho I_{np})\big[x(k+1) - x(k)\big] \\ &+ \phi(k) + cL_ox(k+1) = 0.\end{aligned} \tag{39}$$

Regrouping terms in (39) and observing that the degree matrix is $D = (1/2)(L_o + L_u)$ yield

$$\begin{aligned}&(2cD + \rho I_{np})x(k+1) \\ &= (cL_u + \rho I_{np})x(k) - \nabla f(x(k)) - \phi(k).\end{aligned} \tag{40}$$

The update for $x(k+1)$ in (11) follows from (40) by using the notations $\tilde{D} = 2cD + \rho I_{np}$ and $\tilde{L}_u = cL_u + \rho I_{np}$. ■

APPENDIX II
PROOF OF LEMMA 1

Proof: Write down the KKT conditions for the decentralized optimization problem in (4) so as to obtain the equalities

$$\nabla f(x^*) + A^T\lambda^* = 0, \tag{41}$$

$$B^T\lambda^* = 0, \tag{42}$$

$$Ax^* + Bz^* = 0. \tag{43}$$

The definition of the matrix $B = [-I_{mp}; -I_{mp}]$ and the equality $B^T\lambda^* = 0$ in (42) imply that the optimal multiplier $\lambda^* = [\alpha^*; \beta^*]$ must satisfy $\alpha^* = -\beta^*$.

Using the fact of $\alpha^* = -\beta^*$ and the definitions of $A = [A_s; A_d]$ and $E_o = A_s - A_d$, we can rewrite (41) as

$$\nabla f(x^*) + E_o^T\alpha^* = 0. \tag{44}$$

For any optimal primal solution x^*, (44) suggests that there are multiple optimal dual solutions α^*. To see so, observe that the oriented Laplacian $L_o = (1/2)E_o^TE_o \in \mathbb{R}^{np\times np}$ is rank deficient. Therefore, the rank of $E_o^T \in \mathbb{R}^{np\times mp}$ is less than np, which is no more than mp for any connected network with $n > 1$ agents. Consequently, there are multiple vectors $\alpha^* \in \mathbb{R}^{mp}$ satisfying (44).

Given an optimal dual solution $\hat{\alpha}$ that satisfies $\nabla f(x^*) + E_o^T\hat{\alpha} = 0$, its projection onto the column space of E_o, denoted by α^*, is also an optimal dual solution. This is true because according to the property of projection, $E_o^T\left[\hat{\alpha} - \alpha^*\right] = 0$ and hence $E_o^T\hat{\alpha} = E_o^T\alpha^*$. Therefore, α^* satisfies $\nabla f(x^*) + E_o^T\alpha^* = 0$ and is an optimal dual solution, showing that there exists an optimal dual variable α^* lying in the column space of E_o.

We prove uniqueness of such an α^* by contradiction. Consider two vectors $E_o r_a$ and $E_o r_b$ both lying in the column space of E_o where $r_a, r_b \in \mathbb{R}^{np}$ and $E_o r_a \neq E_o r_b$. If they are both optimal dual solutions, then from (44)

$$\begin{aligned} \nabla f(x^*(k)) + E_o^T E_o r_a = 0, \\ \nabla f(x^*(k)) + E_o^T E_o r_a = 0. \end{aligned} \tag{45}$$

Subtracting the two equalities in (45) yields

$$E_o^T E_o[r_a - r_b] = 0. \tag{46}$$

Observing that $\|E_o^T E_o[r_a - r_b]\| \geq \sqrt{2\gamma_o}\|E_o[r_a - r_b]\|$ where γ_o is the smallest nonzero eigenvalue value of $L_o = E_o^T E_o/2$ and hence $\sqrt{2\gamma_o}$ is the smallest nonzero singular value of E_o, (46) implies that $\|E_o[r_a - r_b]\| = 0$, which contradicts with $E_o r_a \neq E_o r_b$. Since this is absurd we must have $E_o r_a = E_o r_b$ implying that there is a unique optimal dual solution α^* lying in the column space of E_o. ■

APPENDIX III
PROOF OF LEMMA 2

Proof : In this proof we reuse some intermediate results from the proofs of Proposition 1 and Lemma 2. Begin by considering (37) and reorder terms to conclude that under the initial conditions $\alpha(0) = -\beta(0)$ and $E_o x(0) = 2z(0)$, we have that

$$\frac{c}{2}E_o x(k+1) = \alpha(k+1) - \alpha(k). \tag{47}$$

Further consider (38) and use the definition $\tilde{L}_u = cL_u + \rho I_{np} = (c/2)E_u^T E_u + \rho I_{np}$ to write

$$\nabla f(x(k)) + \tilde{L}_u\big[x(k+1) - x(k)\big] + E_o^T\alpha(k+1) = 0. \tag{48}$$

For a pair of optimal primal and dual solutions x^* and α^*, combining the KKT conditions $\nabla f(x^*) + A^T\lambda^* = 0$ [cf. (41)] and $B^T\lambda^* = 0$ [cf. (42)] yields

$$\nabla f(x^*) + E_o^T\alpha^* = 0, \tag{49}$$

as we have shown in (44). Now we consider the other KKT condition $Ax^* + Bz^* = 0$ [cf. (43)]. From the definitions of $A = [A_s; A_d]$ and $B = [-I_{mp}; -I_{mp}]$, we separate the condition into $A_s x^* - z^* = 0$ and $A_d x^* - z^* = 0$. Subtracting the two equalities and using the definition of the oriented incidence matrix $E_o = A_s - A_d$, it follows that $E_o x^* = 0$ and consequently

$$\frac{c}{2}E_o x^* = 0. \tag{50}$$

Subtracting (49) from (48) yields (18) and subtracting (50) from (47) yields (19). ■

APPENDIX IV
PROOF OF LEMMA 3

Proof: From Assumptions 1 and 2 the aggregate cost function $f(x)$ is convex and has Lipschitz continuous gradients with constant M_f, therefore it holds, see e.g., [37],

$$\begin{aligned} &\frac{1}{M_f}\|\nabla f(x(k)) - \nabla f(x^*)\|^2 \\ &\quad\leq \left[x(k) - x^*\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\quad= \left[x(k+1) - x^*\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\qquad+ \left[x(k) - x(k+1)\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right]. \end{aligned} \tag{51}$$

We consider the two terms on the right-hand side of (51) separately. For the first summand in (51) substitute the result in (18) of Lemma 2 for the factor $\nabla f(x(k)) - \nabla f(x^*)$ so as to write

$$\begin{aligned} &\left[x(k+1) - x^*\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\quad= \left[x(k+1) - x^*\right]^T\tilde{L}_u\left[x(k) - x(k+1)\right] \\ &\qquad- \left[x(k+1) - x^*\right]^T E_o^T\left[\alpha(k+1) - \alpha^*\right]. \end{aligned} \tag{52}$$

According to (19) we know that $(c/2)E_o[x(k+1) - x^*] = \alpha(k+1) - \alpha(k)$, hence (52) can be rewritten as

$$\begin{aligned} &\left[x(k+1) - x^*\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\quad= \left[x(k+1) - x^*\right]^T\tilde{L}_u\left[x(k) - x(k+1)\right] \\ &\qquad- \frac{2}{c}\left[\alpha(k+1) - \alpha(k)\right]^T\left[\alpha(k+1) - \alpha^*\right]. \end{aligned} \tag{53}$$

Use the the definition of the Euclidean norm with respect to the matrix $\tilde{L}_u$ to conclude that

$$\begin{aligned} &2\left[x(k+1) - x^*\right]^T\tilde{L}_u\left[x(k) - x(k+1)\right] \\ &= \|x(k) - x^*\|^2_{\tilde{L}_u} - \|x(k+1) - x^*\|^2_{\tilde{L}_u} - \|x(k) - x(k-1)\|^2_{\tilde{L}_u}, \end{aligned} \tag{54}$$

which can be easily verified by expanding the squares and canceling terms in the right hand side. Further observe that

$$\begin{aligned} &2\left[\alpha(k+1) - \alpha(k)\right]^T\left[\alpha(k+1) - \alpha^*\right] \\ &\quad= \|\alpha(k+1) - \alpha(k)\|^2 - \|\alpha(k) - \alpha^*\|^2 + \|\alpha(k+1) - \alpha^*\|^2, \end{aligned} \tag{55}$$

which can be easily verified as well by expanding the squares and canceling terms in the right-hand side. Substituting (54) and (55) into (53) and using the definition of the energy function $V_{x^*,\alpha^*}(k)$ in (20), yields

$$\begin{aligned} &\left[x(k+1) - x^*\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\quad= V_{x^*,\alpha^*}(k) - V_{x^*,\alpha^*}(k+1) \\ &\qquad- \frac{1}{2}\|x(k+1) - x(k)\|^2_{\tilde{L}_u} - \frac{1}{c}\|\alpha(k+1) - \alpha(k)\|^2. \end{aligned} \tag{56}$$

The second summand in the right-hand side of (51) can be upper bounded using the basic inequality $\langle r_a, r_b\rangle \leq (M_f/4)\|r_a\|^2 + (1/M_f)\|r_b\|^2$, where $r_a, r_b \in \mathbb{R}^{np}$ and $M_f > 0$, to write

$$\begin{aligned} &\left[x(k) - x(k+1)\right]^T\left[\nabla f(x(k)) - \nabla f(x^*)\right] \\ &\leq \frac{M_f}{4}\|x(k) - x(k+1)\|^2 + \frac{1}{M_f}\|\nabla f(x(k)) - \nabla f(x^*)\|^2. \end{aligned} \tag{57}$$

Substituting the equality in (56) and the upper bound in (57) for the corresponding terms of (51) yields after regrouping terms

$$V_{x^*,\alpha^*}(k) - V_{x^*,\alpha^*}(k+1) \tag{58}$$
$$\geq \frac{1}{2}\|x(k+1)-x(k)\|^2_{\tilde{L}_u - M_f I_{np}/2} + \frac{1}{c}\|\alpha(k+1)-\alpha(k)\|^2.$$

Observe the fact that the smallest eigenvalue of $\tilde{L}_u$ is $c\gamma_u + \rho$ such that the smallest eigenvalue of $\tilde{L}_u - M_f I_{np}/2$ is $c\gamma_u + \rho - M_f/2$. Hence $c\gamma_u + \rho - M_f/2 > 0$ guarantees that $\|x(k+1)-x(k)\|^2_{\tilde{L}_u - M_f I_{np}/2} \geq \xi \|x(k+1)-x(k)\|^2_{\tilde{L}_u}$ where $\xi = (c\gamma_u + \rho - M_f/2)/(c\gamma_u + \rho)$. Also, we know that $\|\alpha(k+1) - \alpha(k)\|^2 \geq \xi \|\alpha(k+1)-\alpha(k)\|^2$ as $\xi < 1$. Substituting these inequalities into (58) yields (22) and completes the proof. ■

Appendix V
Proof of Theorem 1

Proof: Lemma 3 implies that $V_{x^*,\alpha^*}(k)$ is monotonically non-increasing. Since $c\gamma_u + \rho - 2M_f > 0$ by hypothesis, we know that $\tilde{L}_u \succ 0$ as its smallest eigenvalue $c\gamma_u + \rho > 0$, and thus $V_{x^*,\alpha^*}(k)$ is nonnegative. These two facts guarantee convergence of $V_{x^*,\alpha^*}(k)$, which further guarantees convergence of $(1/2)\|x(k+1)-x(k)\|^2_{\tilde{L}_u} + (1/c)\|\alpha(k+1)-\alpha(k)\|^2$ to 0. Again, due to $\tilde{L}_u \succ 0$ we conclude that both $x(k+1)-x(k)$ and $\alpha(k+1)-\alpha(k)$ converge to 0. From the convergence of $x(k+1)-x(k)$ and (38), we conclude that $\nabla f(x(k)) + E_o^T \alpha(k+1)$ converges to 0, which further implies that $\nabla f(x(k+1)) + E_o^T \alpha(k+1)$ converges to 0 by the Lipschitz continuity of $\nabla f(x)$. From the convergence of $\alpha(k+1) - \alpha(k)$ and (37), we conclude that $(c/2)E_o x(k)$ converges to 0.

Let x^* and α^* be a pair of optimal primal and dual solutions of (4) whose values are finite. Monotonicity and nonnegativity of $V_{x^*,\alpha^*}(k)$ imply that the sequence $(x(k),\alpha(k))$ lies in a compact region. Therefore, $(x(k),\alpha(k))$ has at least a subsequence that converges to a limit point. From the discussion above, for any limit point $(\hat{x},\hat{\alpha})$ we know that $\nabla f(\hat{x}) + E_o^T\hat{\alpha}$ and $(c/2)E_o\hat{x} = 0$. Hence, we conclude that any limit point $(\hat{x},\hat{\alpha})$ satisfies the KKT conditions (cf. (49) and (50)) and is an optimal solution to (4).

To complete the proof, it remains to show that the sequence $(x(k),\alpha(k))$ only has a unique limit point. Let $(\hat{x}_a,\hat{\alpha}_a)$ and $(\hat{x}_b,\hat{\alpha}_b)$ be any two limit points of $(x(k),\alpha(k))$. As we have proved above, both $(\hat{x}_a,\hat{\alpha}_a)$ and $(\hat{x}_b,\hat{\alpha}_b)$ are optimal solutions to (4). Similar to (20), we define the energy functions

$$V_{\hat{x}_a,\hat{\alpha}_a}(x,\alpha) := \frac{1}{2}\|x-\hat{x}_a\|^2_{\tilde{L}_u} + \frac{1}{c}\|\alpha-\hat{\alpha}_a\|^2, \tag{59}$$
$$V_{\hat{x}_b,\hat{\alpha}_b}(x,\alpha) := \frac{1}{2}\|x-\hat{x}_b\|^2_{\tilde{L}_u} + \frac{1}{c}\|\alpha-\hat{\alpha}_b\|^2.$$

Also, we let $V_{\hat{x}_a,\hat{\alpha}_a}(k) = V_{\hat{x}_a,\hat{\alpha}_a}(x(k),\alpha(k))$ and $V_{\hat{x}_b,\hat{\alpha}_b}(k) = V_{\hat{x}_b,\hat{\alpha}_b}(x(k),\alpha(k))$. From (22) we have

$$V_{\hat{x}_a,\hat{\alpha}_a}(k) \geq V_{\hat{x}_a,\hat{\alpha}_a}(k+1), \tag{60}$$
$$V_{\hat{x}_b,\hat{\alpha}_b}(k) \geq V_{\hat{x}_b,\hat{\alpha}_b}(k+1).$$

Hence we know the limits

$$\lim_{k\to\infty} V_{\hat{x}_a,\hat{\alpha}_a}(k) = \eta_a < \infty, \tag{61}$$
$$\lim_{k\to\infty} V_{\hat{x}_b,\hat{\alpha}_b}(k) = \eta_b < \infty.$$

Consider the equality

$$\begin{aligned} V_{\hat{x}_a,\hat{\alpha}_a}(k) - V_{\hat{x}_b,\hat{\alpha}_b}(k) &= -\langle x(k), \hat{x}_a - \hat{x}_b\rangle_{\tilde{L}_u} - \frac{2}{c}\langle \alpha(k), \hat{\alpha}_a - \hat{\alpha}_b\rangle \\ &\quad + \frac{1}{2}\|\hat{x}_a\|^2_{\tilde{L}_u} + \frac{1}{c}\|\hat{\alpha}_a\|^2 - \frac{1}{2}\|\hat{x}_b\|^2_{\tilde{L}_u} - \frac{1}{c}\|\hat{\alpha}_b\|^2. \end{aligned} \tag{62}$$

Since $(\hat{x}_a,\hat{\alpha}_a)$ is a limit point of $(x(k),\alpha(k))$, using (61) and taking the limit of (62) leads to

$$\begin{aligned} \eta_a - \eta_b &= -\langle \hat{x}_a, \hat{x}_a - \hat{x}_b\rangle_{\tilde{L}_u} - \frac{2}{c}\langle \hat{\alpha}_a, \hat{\alpha}_a - \hat{\alpha}_b\rangle \\ &\quad + \frac{1}{2}\|\hat{x}_a\|^2_{\tilde{L}_u} + \frac{1}{c}\|\hat{\alpha}_a\|^2 - \frac{1}{2}\|\hat{x}_b\|^2_{\tilde{L}_u} - \frac{1}{c}\|\hat{\alpha}_b\|^2 \\ &= -\frac{1}{2}\|\hat{x}_a - \hat{x}_b\|^2_{\tilde{L}_u} - \frac{1}{c}\|\hat{\alpha}_a - \hat{\alpha}_b\|^2. \end{aligned} \tag{63}$$

Similarly, since $(\hat{x}_b,\hat{\alpha}_b)$ is a limit point of $(x(k),\alpha(k))$, using (61) and taking the limit of (62) leads to

$$\begin{aligned} \eta_a - \eta_b &= -\langle \hat{x}_b, \hat{x}_a - \hat{x}_b\rangle_{\tilde{L}_u} - \frac{2}{c}\langle \hat{\alpha}_b, \hat{\alpha}_a - \hat{\alpha}_b\rangle \\ &\quad + \frac{1}{2}\|\hat{x}_a\|^2_{\tilde{L}_u} + \frac{1}{c}\|\hat{\alpha}_a\|^2 - \frac{1}{2}\|\hat{x}_b\|^2_{\tilde{L}_u} - \frac{1}{c}\|\hat{\alpha}_b\|^2 \\ &= \frac{1}{2}\|\hat{x}_a - \hat{x}_b\|^2_{\tilde{L}_u} + \frac{1}{c}\|\hat{\alpha}_a - \hat{\alpha}_b\|^2. \end{aligned} \tag{64}$$

Thus we must have $(1/2)\|\hat{x}_a - \hat{x}_b\|^2_{\tilde{L}_u} + (1/c)\|\hat{\alpha}_a - \hat{\alpha}_b\|^2 = 0$, which proves that the limit point of $(x(k),\alpha(k))$ is unique. ■

Appendix VI
Proof of Theorem 2

Proof: We begin the proof in a way similar to what we have done in the proof of Lemma 3. Instead of using the fact that the aggregate cost function $f(x)$ is convex (cf. Assumption 1) and has Lipschitz gradients with constant M_f (cf. Assumption 2) in Lemma 3, here we observe that the aggregate cost function $f(x)$ is strongly convex with constant m_f (cf. Assumption 3). Further, we consider the relation between $x(k+1) - x^*$ and $\nabla f(x(k+1)) - \nabla f(x^*)$ instead of that between $x(k) - x^*$ and $\nabla f(x(k)) - \nabla f(x^*)$. Under Assumption 3 it holds

$$\begin{aligned} m_f\|x(k+1)-x^*\|^2 &\leq [x(k+1)-x^*]^T[\nabla f(x(k+1)) - \nabla f(x^*)] \\ &= [x(k+1)-x^*]^T[\nabla f(x(k)) - \nabla f(x^*)] \\ &\quad + [x(k+1)-x^*]^T[\nabla f(x(k+1)) - \nabla f(x(k))]. \end{aligned} \tag{65}$$

Manipulating the first term at the right-hand side of (65) as we have done in the proof of Lemma 3 (cf. (52)-(55)), we obtain the equality (cf. (56))

$$\begin{aligned} [x(k+1)-x^*]^T[\nabla f(x(k)) - \nabla f(x^*)] &= V_{x^*,\alpha^*}(k) - V_{x^*,\alpha^*}(k+1) \\ &\quad - \frac{1}{2}\|x(k+1)-x(k)\|^2_{\tilde{L}_u} - \frac{1}{c}\|\alpha(k+1)-\alpha(k)\|^2. \end{aligned} \tag{66}$$

Substituting (66) into (65) yields

$$\begin{aligned} m_f\|x(k+1)-x^*\|^2 &\leq [x(k+1)-x^*]^T[\nabla f(x(k+1)) - \nabla f(x(k))] \\ &\quad + V_{x^*,\alpha^*}(k) - V_{x^*,\alpha^*}(k+1) \\ &\quad - \frac{1}{2}\|x(k+1)-x(k)\|^2_{\tilde{L}_u} - \frac{1}{c}\|\alpha(k+1)-\alpha(k)\|^2. \end{aligned} \tag{67}$$

Next, we prove that there exists a contraction parameter

$$\delta = \min\left\{ \frac{m_f - \theta/2}{(c\Gamma_u+\rho)/2 + (\mu M_f^2)/(c\gamma_o)}, \right. \tag{68}$$
$$\left. \frac{(c\gamma_u+\rho)/2 - M_f^2/(2\theta)}{\mu(c\Gamma_u+\rho)^2/[(\mu-1)(2c\gamma_o)] + \mu M_f^2/(c\gamma_o)} \right\} > 0,$$

such that it holds

$$\begin{aligned} &\delta V_{x^*,\alpha^*}(k+1) \qquad (69)\\ &\leq\ m_f\|x(k+1)-x^*\|^2 + \frac{1}{2}\|x(k+1)-x(k)\|^2_{\tilde{L}_u} \\ &\quad - [x(k+1)-x^*]^T\,[\nabla f(x(k+1)) - \nabla f(x(k))]. \end{aligned}$$

In (68), μ is an arbitrary constant satisfying $\mu > 1$ and θ is an arbitrary constant satisfying $2m_f > \theta > M_f^2/(c\gamma_u+\rho)^2$. Observe that such a θ exists since by hypothesis $m_f(c\gamma_u+\rho)^2 > M_f^2/2$ and guarantees δ to be positive.

To prove (69), we develop lower bounds for its left-hand side terms and upper bounds for its right-hand side terms. Observing that $\nabla f(x)$ is Lipschitz continuous with constant M_f and substituting the expression of $\nabla f(x(k)) - \nabla f(x^*)$ in (18), we have

$$\begin{aligned} &M_f^2\|x(k)-x^*\|^2 \qquad (70)\\ &\quad\geq \|\nabla f(x(k)) - \nabla f(x^*)\|^2 \\ &\quad= \|\tilde{L}_u[x(k)-x(k+1)] - E_o^T[\alpha(k+1)-\alpha^*]\|^2. \end{aligned}$$

Using the basic inequality $\|v_b - v_a\|^2 \geq (1/\mu)\|v_a\|^2 - (1/(\mu-1))\|v_b\|^2$ that holds for any $\mu > 1$, we separate the right-hand side of (70) and obtain

$$\begin{aligned} &M_f^2\|x(k)-x^*\|^2 \qquad (71)\\ &\geq \frac{1}{\mu}\|E_o^T[\alpha(k+1)-\alpha^*]\|^2 - \frac{1}{\mu-1}\|\tilde{L}_u[x(k)-x(k+1)]\|^2. \end{aligned}$$

Since the largest eigenvalue of $\tilde{L}_u$ is $c\Gamma_u+\rho$, we have $\|\tilde{L}_u[x(k)-x(k+1)]\|^2 \leq (c\Gamma_u+\rho)^2\|x(k)-x(k+1)\|^2$; also, since both $\alpha(k+1)$ and α^* lie in the column space of E_o and the smallest nonzero eigenvalue of $L_o = (E_o^T E_o)/2$ is γ_o it holds $\|E_o^T[\alpha(k+1)-\alpha^*]\|^2 \geq 2\gamma_o\|\alpha(k+1)-\alpha^*\|^2$. Using these inequalities (71) leads to

$$\begin{aligned} \frac{\delta}{c}\|\alpha(k+1)-\alpha^*\|^2 \leq\ & \frac{\delta}{2c\gamma_o}\mu M_f^2\|x(k)-x^*\|^2 \qquad (72)\\ &+ \frac{\delta\mu(c\Gamma_u+\rho)^2}{2c\gamma_o(\mu-1)}\|x(k)-x(k+1)\|^2. \end{aligned}$$

Again from the basic inequality and Lipschitz continuity of $\nabla f(x)$, for any $\theta > 0$ it holds

$$\begin{aligned} &-[x(k+1)-x^*]^T\,[\nabla f(x(k+1)) - \nabla f(x(k))] \qquad (73)\\ &\quad\geq -\frac{\theta}{2}\|x(k+1)-x^*\|^2 - \frac{1}{2\theta}\|\nabla f(x(k+1)) - \nabla f(x(k))\|^2 \\ &\quad\geq -\frac{\theta}{2}\|x(k+1)-x^*\|^2 - \frac{1}{2\theta}M_f^2\|x(k+1)-x(k)\|^2. \end{aligned}$$

Since the largest and smallest eigenvalues of $\tilde{L}_u$ are $c\Gamma_u+\rho$ and $c\gamma_u+\rho$ (which is positive by hypothesis), respectively, $\|x(k+1)-x(k)\|^2_{\tilde{L}_u} \geq (c\gamma_u+\rho)\|x(k+1)-x(k)\|^2$ and $\|x(k+1)-x^*\|^2_{\tilde{L}_u} \leq (c\Gamma_u+\rho)\|x(k+1)-x^*\|^2$. Combining these two inequalities as well as (72) and (73), the sufficient condition of (69) is

$$\begin{aligned} &\left(m_f - \frac{\theta}{2} - \frac{\delta(c\Gamma_u+\rho)}{2}\right)\|x(k+1)-x^*\|^2 \qquad (74)\\ &+ \left(\frac{c\gamma_u+\rho}{2} - \frac{M_f^2}{2\theta} - \frac{\delta\mu(c\Gamma_u+\rho)^2}{2c\gamma_o(\mu-1)}\right)\|x(k+1)-x(k)\|^2 \\ &\geq \frac{\delta}{2c\gamma_o}\mu M_f^2\|x(k)-x^*\|^2, \end{aligned}$$

which is true for the contraction parameter $\delta > 0$ in (68) since $2\|x(k+1)-x^*\|^2 + 2\|x(k+1)-x(k)\|^2 \geq \|x(k)-x^*\|^2$. Combining (67) and (69) yields the claim in (24). ■

REFERENCES

[1] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2014

[2] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," IEEE Trans. Signal Process., vol. 58, pp. 3816–3827, 2010

[3] N. Nguyen, N. Nasrabadi, and T. Tran, "Robust multi-sensor classification via joint sparse representation," In: Proceedings of International Conference on Information Fusion, 2011

[4] C. Eksin and A. Ribeiro, "Distributed network optimization with heuristic rational agents," IEEE Trans. Signal Process., vol. 60, pp. 5396–5411, 2012

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

[6] V. Kekatos and G. Giannakis, "Distributed robust power system state estimation," IEEE Trans. Power Syst., vol. 28, pp. 1617–1626, 2013

[7] J. Bazerque, G. Mateos, and G. Giannakis, "Group-lasso on splines for spectrum cartograph," IEEE Trans. Signal Process., vol. 59, pp. 4648–4663, 2011

[8] 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, pp. 37–48, 2011

[9] 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, pp. 327–337, 2011

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine Learning, vol. 3, pp. 1–122, 2010

[11] K. Tsianos, S. Lawlor, and M. Rabbat, "Communication/computation tradeoffs in consensus-based distributed optimization," In: Proceedings of Advances in Neural Information Processing Systems, 2012

[12] K. Tsianos, S. Lawlor, and M. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," In: Proceedings of Allerton Conference on Communication, Control, and Computing, 2012

[13] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," IEEE Trans. Autom. Control, vol. 54, pp. 48–61, 2009

[14] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," Journal of Optimization Theory and Applications, vol. 147, pp. 516–545, 2010

[15] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," Manuscript available at `http://arxiv.org/pdf/1310.7063v1.pdf`

[16] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," IEEE Transactions on Automatic Control, vol. 59, pp. 1131–1146, 2014

[17] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," IEEE Trans. Autom. Control, vol. 57, pp. 592–606, 2012

[18] K. Tsianos and M. Rabbat, "Distributed dual averaging for convex optimization under communication delays," In: Procedings of American Control Conference, 2012

[19] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," In: Proceedings of IEEE International Workshop on Signal Processing Advances for Wireless Communications, 2005

[20] F. Jakubiec and A. Ribeiro, "D-MAP: Distributed maximum a posteriori probability estimation of dynamic systems," IEEE Trans. Signal Process., vol. 61, pp. 450–466, 2013

[21] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Second Edition, Athena Scientific, 1997

[22] 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, pp. 350–364, 2008

[23] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," IEEE Trans. Signal Process., vol. 58, pp. 5262–5276, 2010

[24] 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, pp. 1750–1761, 2014

[25] M. Ng, F. Wang, and X. Yuan, "Inexact alternating direction methods for image recovery," SIAM Journal on Scientific Computing, vol. 33, pp. 1643–1668, 2011

[26] S. Ma, "Alternating proximal gradient method for convex minimization," Manuscript available at `http://www.optimization-online.org/DB_FILE/2012/09/3608.pdf`

[27] S. Ma and S. Zhang, "An extragradient-based alternating direction method for convex minimization," Manuscript available at `http://arxiv.org/pdf/1301.6308.pdf`

[28] T. Chang, M. Hong, and X. Wang, "Multiagent distributed large-scale optimization by inexact consensus alternating direction method of multipliers," In: Proceedings of the 39th IEEE International Conference on Acoustics, Speech, and Signal Processing, 2014

[29] P. Bianchi, W. Hachem, and F. Iutzeler, "A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization," Manuscript available at `http://arxiv.org/pdf/1407.0898v1.pdf`

[30] D. Cvetkovic, P. Rowlinson, and S. Simic, "Signless Laplacians of finite graphs," Linear Algebra and Its Applications, vol. 423, pp. 155–171, 2007

[31] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," IEEE Trans. Signal Process., vol. 62, pp. 1185–1197, 2014

[32] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," SIAM Review, vol. 46, pp. 667–689, 2004

[33] W. Deng, M. Lai, and W. Yin, "On the $o(1/k)$ convergence and parallelization of the alternating direction method of multipliers," Manuscript available at `http://arxiv.org/pdf/1312.3040v1.pdf`

[34] M. Hong, Z. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," Manuscript available at `http://arxiv.org/pdf/1410.3390.pdf`

[35] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Manuscript available at `http://www.optimization-online.org/DB_FILE/2012/08/3578.pdf`

[36] K. Murphy, *Machine Learning: A Probabilistic Perspective*, The MIT Press, 2012

[37] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2004

[38] M. Hong and Z. Luo, "On the linear convergence of the alternating direction method of multipliers," Manuscript available at `http://arxiv.org/pdf/1208.3922v3.pdf`

[39] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," Manuscript available at `http://arxiv.org/pdf/1312.1085v1.pdf`

[40] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," Manuscript available at `http://arxiv.org/pdf/1306.2454v2.pdf`