DECENTRALIZED LINEARIZED ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Qing Ling[*] *Alejandro Ribeiro*[†]

[*] Department of Automation, University of Science and Technology of China, Hefei, Anhui, China
[†] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, USA

ABSTRACT

This paper develops a decentralized linearized alternating direction method of multipliers (LADMM) that minimizes the sum of local cost functions in a multi-agent network. Through linearizing the local cost functions agents can obtain their local solutions with simple algebraic operations and gradient descent steps. We prove that the algorithm linearly converges to the optimal solution given that the local cost functions are strongly convex and have Lipschitz gradients. The decentralized LADMM has similar computations as the distributed (sub)gradient method but outperforms the latter, which is unable to achieve linear rate of convergence and convergence to the exact optimal solution simultaneously. Compared to its non-linearized counterpart that suffers from high computation burden, the decentralized LADMM has a comparable rate of convergence according to both theoretical analysis and numerical experiments.

Index Terms— Multi-agent network, decentralized optimization, linearized alternating direction method of multipliers

1. INTRODUCTION

We consider a multi-agent 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 that aim to solve $\tilde{x}^* := \operatorname{argmin} \ \sum_{i=1}^{n} f_i(\tilde{x})$. The problem is decentralized because the objective is separated into local cost functions $f_i : \mathbb{R}^p \to \mathbb{R}$ that are known to different agents i. The decentralized optimization problem (1) arise in various applications, such as event detection in wireless sensor networks [1], state estimation in smart grids [2], spectrum sensing in cognitive radio networks [3], etc.

Algorithms that solve the decentralized optimization problem (1) fall into either the primal domain or the dual domain. In primal domain methods, each agent averages its local solution with those of neighbors and descends along its local negative (sub)gradient direction. Typical primal domain methods include the distributed (sub)gradient method [4, 5, 6] and the dual averaging method [7]. Dual domain methods rewrite (1) to a constrained form where the constraints force local solutions to reach a 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 [8]. The decentralized alternating direction method of multipliers (ADMM) modifies dual ascent by introducing a quadratic regularization term and improves its numerical stability and rate of convergence [9, 10, 11].

The main advantage of the primal domain methods is their low computation burden. The average and (sub)gradient descent operations are simple and hence affordable to those agents having limited computation abilities. However, the existing primal domain methods suffer from either slow convergence or low accuracy. With time-varying stepsizes, the distributed (sub)gradient method and the dual averaging method converge to the optimal solution at sublinear rates [6, 7]. 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 are strongly convex and have Lipschitz gradients; however, the algorithm converges to a neighborhood of the optimal solution [5]. Contrarily, the dual domain methods often bring high computation burden. At each iteration, each agent needs to solve an optimization problem whose objective is the local cost function plus a linear term in the dual ascent method [8] or plus a quadratic term in the decentralized ADMM [9, 10]. At the cost of high computation burden, the decentralized ADMM is able to achieve fast convergence to the exact optimal solution; its rate of convergence is linear when the local cost functions are strongly convex and have Lipschitz gradients [11].

This paper develops the decentralized linearized alternating direction method of multipliers (LADMM) that enjoys the advantages of both the primal and dual methods, i.e., low computation burden and fast convergence to the exact optimal solution; specifically, we establish its linear rate of convergence under the same assumptions on the local cost functions as in [5] and [11].

The main idea of the decentralized LADMM is to linearize the local cost functions such that agents can obtain their local solutions through simple algebraic operations and gradient descent steps. Nevertheless, the decentralized LADMM is not a trivial application of the centralized LADMM. The existing centralized LADMM often linearizes the augmented quadratic terms such that the subproblems have explicit solutions, which is unsuitable in decentralized optimization [12, 13]. Linearizing the original cost functions makes algorithm analysis intractable; in view of this difficulty, [14] introduces an extra gradient step after the linearization step. Due to the special structure of the decentralized optimization problem, this paper is able to analyze the decentralized LADMM that directly linearizes the original local cost functions.

2. 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

Qing Ling is supported by NSFC 61004137. Alejandro Ribeiro is supported by NSF CAREER CCF-0952867, NSF CCF-1017454, and AFOSR MURI FA9550-10-1-0567.

can communicate to j. Assume the communication is bidirectional so that if $e \sim (i,j)$ there exists another arc $e' \sim (j,i)$. Agents adjacent to i is termed its neighbors and denoted as a set $\mathcal{N}_i$. The cardinality of this set is the degree d_i of agent i. Define the block arc source matrix $A_s \in \mathbb{R}^{mp\times np}$ where the block $(A_s)_{e,i} = I_p \in \mathbb{R}^{p\times p}$ is an identity matrix if the arc $e \sim (i,j)$ originates at node i and is null otherwise. Likewise, define the block arc destination matrix $A_d \in \mathbb{R}^{mp\times np}$ where 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 can be 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 then 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 is $D = (1/2)(L_o+L_u)$. Denote Γ_u and γ_u as the largest and smallest eigenvalues of L_u, respectively, and Γ_u as the smallest nonzero eigenvalue of L_o; Γ_u, γ_u, and γ_o are measures of network connectedness.

To develop the decentralized LADMM that solves (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

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

The constraints $x_i = z_{ij}$ and $x_j = z_{ij}$ force neighboring agents i and j to reach a consensus on their local copies x_i and x_j. For a connected network, (2) is equivalent to (1) in the sense that for all i and j the optimal solutions $x_i = \tilde{x}^*$ and $z_{ij} = \tilde{x}^*$ where $\tilde{x}^*$ is the optimal solution of (1).

Define $x = [x_1;\dots;x_n] \in \mathbb{R}^{np}$ concatenating all local copies x_i and $z = [z_1;\dots;z_m] \in \mathbb{R}^{mp}$ concatenating all auxiliary variables $z_e = z_{ij}$, and let the aggregate function $f : \mathbb{R}^{np} \to \mathbb{R}$ as $f(x) := \sum_{i=1}^n f_i(x_i)$. Defining $A = [A_s; A_d] \in \mathbb{R}^{2mp\times np}$ and $B = [-I_{mp}; -I_{mp}]$, we can rewrite (2) in a matrix form as

$$\min \ f(x), \quad \text{s.t. } Ax + Bz = 0, \tag{3}$$

where $Ax+Bz = 0$ include two parts $A_s x - z = 0$ and $A_d x - z = 0$. Consider 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. Define $\lambda = [\alpha;\beta] \in \mathbb{R}^{2mp}$ associated with the constraint $Ax + Bz = 0$, the augmented Lagrangian function of (3) is

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

where $c > 0$ is a positive constant.

At each iteration, the decentralized LADMM minimizes the augmented Lagrangian function with respect to x and z in an alternating direction manner, where the minimization of x is inexact and the minimization of z is exact. Then the Lagrange multiplier λ is updated through dual ascent. At time k and given past iterates $x(k-1)$, $z(k-1)$ and $\lambda(k-1)$, the primal iterate $x(k)$ is defined as

$$\begin{aligned} x(k) := \arg\min_x \ &\nabla f(x(k-1))^T x + \frac{\rho}{2}\|x - x(k-1)\|^2 \\ &+ \lambda(k-1)^T(Ax + Bz(k-1)) + \frac{c}{2}\|Ax + Bz(k-1)\|^2, \end{aligned} \tag{4}$$

where the linearization parameter ρ is a constant. In (4), $\nabla f(x(k-1))^T(x-x(k-1)) + \frac{\rho}{2}\|x-x(k-1)\|^2$ is a quadratic approximation of $f(x)$ at point $x(k-1)$. Note that in the decentralized ADMM, to update $x(k)$ the quadratic approximation term is replaced by the original cost function $f(x)$ [11]. Represent $x(k)$ as the solution of the first order optimality condition

$$\begin{aligned} &\nabla f(x(k-1)) + \rho(x(k) - x(k-1)) \\ &+ A^T\lambda(k-1) + cA^T\big[Ax(k) + Bz(k-1)\big] = 0. \end{aligned} \tag{5}$$

Using the value of $x(k)$ from (5) along with the previous dual iterate $\lambda(k-1)$ the primal iterate $z(k)$ is defined as $z(k) := \operatorname{argmin}_z L_k(x(k), z, \lambda(k-1))$ and explicitly given by the solution of the first order optimality condition

$$B^T\lambda(k-1) + cB^T\big[Ax(k) + Bz(k)\big] = 0. \tag{6}$$

The dual iterate $\lambda(k)$ is then updated by the constraint violation $Ax(k) + Bz(k)$ corresponding to primal iterates $x(k)$ and $z(k)$ in order to compute

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

Similar to the decentralized ADMM, with proper initialization the decentralized LADMM iterates (5)-(7) can be simplified such that the auxiliary variable $z(k)$ is eliminated and the Lagrange multipliers $\alpha \in \mathbb{R}^{mp}$ and $\beta \in \mathbb{R}^{mp}$ is replaced by a smaller dimension vector $\phi = [\phi_1;\dots;\phi_n] \in \mathbb{R}^{np}$. The simplification technique is akin to those used in decentralized implementations of the ADMM for static optimization problems; see e.g., [1], [9, Ch. 3], and [11]. We omit the detailed derivation and give the following proposition.

Proposition 1 *Consider iterations* (5)-(7)*. Require the initial multipliers $\lambda(0) = [\alpha(0);\beta(0)]$ to satisfy $\alpha(0) = -\beta(0)$, the initial auxiliary variables $z(0)$ to be such that $E_o x(0) = 2z(0)$ and further define variables $\phi(k) := E_o^T\alpha(k) \in \mathbb{R}^{np}$. Then, for all times $k > 0$, $x(k)$ can be alternatively generated by the recursion*

$$\begin{aligned} &(2cD + \rho I_{np})x(k) - (cL_u + \rho I_{np})x(k-1) \\ &\qquad + \nabla f(x(k-1)) + \phi(k-1) = 0, \end{aligned} \tag{8}$$

$$\phi(k) = \phi(k-1) + cL_o x(k). \tag{9}$$

The iterations in (8) and (9) can be implemented in a decentralized manner. Consider the component of the update for $x(k)$ corresponding to the variable $x_i(k)$. Using the definitions of the degree matrix D and the unoriented Laplacian L_u we can write this component in (8) as

$$\begin{aligned} &(2cd_i + \rho)x_i(k) = (cd_i + \rho)x_i(k-1) \\ &+ c\textstyle\sum_{j\in\mathcal{N}_i} x_j(k-1) - \nabla f_i(x_i(k-1)) - \phi_i(k-1). \end{aligned} \tag{10}$$

Likewise, using the definitions of the oriented Laplacian L_o the update in (9) can be written as

$$\phi_i(k) = \phi_i(k-1) + c\textstyle\sum_{j\in\mathcal{N}_i}\big[x_i(k) - x_j(k)\big]. \tag{11}$$

At the initialization stage, $\phi(0)$ is chosen in the column space of L_o (e.g., $\phi(0) = 0$). 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. Such an initialization is necessary for the analysis in Section 3. The decentralized LADMM run by agent i is summarized in Algorithm 1.

The decentralized LADMM is advantageous over the decentralized ADMM due to its low computation burden. The iterations (10)

Algorithm 1 Decentralized LADMM at agent i

Require: Initialize local variables to $x_i(0) = 0$, $\phi_i(0) = 0$.
Require: Initialize neighboring variables $x_j(0) = 0$ for all $j \in \mathcal{N}_i$.
1: **for** times k = 1, 2, ... **do**
2: Compute local solution $x_i(k)$ from [cf. (10)]

$$(2cd_i + \rho)x_i(k) = (cd_i + \rho)x_i(k-1) + c\sum\nolimits_{j\in\mathcal{N}_i} x_j(k-1) - \nabla f_i(x_i(k-1)) - \phi_i(k-1).$$

3: Transmit $x_i(k)$ to and receive $x_j(k)$ from neighbors $j \in \mathcal{N}_i$.
4: Update local dual variable $\phi_i(k)$ as [cf. (11)]

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

5: **end for**

and (11) only contain simple algebraic operations and gradient descent steps. As a comparison, the counterpart of (10) in the decentralized ADMM is the minimization of $f_i(x_i)$ augmented by a quadratic term of x_i that is often nontrivial. In this sense the decentralized LADMM is more close to a dual domain method. However, the existing dual domain methods are unable to achieve fast convergence and high accuracy simultaneously, while the decentralized LADMM converges at a linear rate to the optimal solution under certain conditions, as we will analyze in Section 3.

3. LINEAR RATE OF CONVERGENCE

This section establishes linear rate of convergence of the decentralized LADMM. Throughout this section, we make the following assumption on the local cost functions f_i.

Assumption 1 Local cost functions are differentiable, strongly convex, and have Lipschitz gradients. For agent i there exists positive constants $m_f > 0$ and $M_f > 0$ such that for any pair of points $\tilde{x}_a$ and $\tilde{x}_b$ it holds $[\tilde{x}_a - \tilde{x}_b]^T[\nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b)] \geq m_f\|\tilde{x}_a - \tilde{x}_b\|^2$ and $\|\nabla f_i(\tilde{x}_a) - \nabla f_i(\tilde{x}_b)\| \leq M_f\|\tilde{x}_a - \tilde{x}_b\|$.

Assumption 1 implies that $f(x) := \sum_{i=1}^n f_i(x_i)$ is also strongly convex and has Lipschitz gradients. For any pair of points x_a and x_b it holds

$$\big[x_a - x_b\big]^T\big[\nabla f(x_a) - \nabla f(x_b)\big] \geq m_f\|x_a - x_b\|^2, \tag{12}$$

and

$$\|\nabla f(x_a) - \nabla f^k(x_b)\| \leq M_f\|x_a - x_b\|. \tag{13}$$

Next we investigate the convergence of the primal iterate $x(k)$ and the dual iterate $\alpha(k)$ to their optima. Under the assumption of strong convexity, the optimal primal solution $x^* = [\tilde{x}^*; \cdots; \tilde{x}^*]$ is unique since $\tilde{x}^*$ is the unique optimal solution of (1). There exist multiple optimal dual solutions; however, we consider a unique optimal dual solution $\lambda^* = [\alpha^*; \beta^*]$ where $\alpha^* = -\beta^*$ lies in the column space of E_o. Existence and uniqueness of such an α^* are proved in [11].

Lemma 1 *Consider the iterates* (5)-(7). *Let the Lagrange multiplier $\lambda(k) = [\alpha(k); \beta(k)]$ be initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables be initialized by $E_o x(0) = 2z(0)$. Let the linearization parameter ρ be chosen such that $c\gamma_u + \rho > 0$ and $m_f(c\gamma_u + \rho)^2 > M_f^2/2$. Under Assumption 1, there exists a contraction parameter $\delta > 0$ such that*

$$\frac{1+\delta}{2}\|x(k) - x^*\|_H^2 + \frac{1+\delta}{c}\|\alpha(k) - \alpha^*\|^2 \leq \frac{1}{2}\|x(k-1) - x^*\|_H^2 + \frac{1}{c}\|\alpha(k-1) - \alpha^*\|^2. \tag{14}$$

Proof: With the initialization $E_o x(0) = 2z(0)$, reorganizing the iterations (5)-(7) we have

$$\nabla f(x(k-1)) + H(x(k) - x(k-1)) + E_o^T\alpha(k) = 0, \tag{15}$$

$$\frac{c}{2}E_o x(k) = \alpha(k) - \alpha(k-1), \tag{16}$$

where $H := cL_u + \rho I_{np}$. In (15) and (16) we have cancelled z and β. Indeed (15) and (16) are equivalent to (8) and (9) noticing that $2D = L_o + L_u$, $2L_o = E_o^T E_o$, and the definition $\phi(k) = E_o^T\alpha(k)$. On the other hand, the KKT conditions of (3) are

$$\nabla f(x^*) + E_o^T\alpha^* = 0, \tag{17}$$

$$E_o x^* = 0. \tag{18}$$

Subtracting (17) from (15) and (18) from (16) yield

$$\nabla f(x(k-1)) - \nabla f(x^*) = H(x(k-1) - x(k)) - E_o^T(\alpha(k) - \alpha^*), \tag{19}$$

$$\frac{c}{2}E_o(x(k) - x^*) = \alpha(k) - \alpha(k-1). \tag{20}$$

First, the strong convexity of f implies that

$$\begin{aligned} m_f\|x(k) - x^*\|^2 &\leq [x(k) - x^*]^T[\nabla f(x(k)) - \nabla f(x^*)] \\ &= [x(k) - x^*]^T[\nabla f(x(k)) - \nabla f(x(k-1))] \\ &\quad + [x(k) - x^*]^T[\nabla f(x(k-1)) - \nabla f(x^*)]. \end{aligned} \tag{21}$$

Substituting (19) into (21) and using (20), (21) can be rewritten as

$$\begin{aligned} &m_f\|x(k) - x^*\|^2 \\ &\leq [x(k) - x^*]^T[\nabla f(x(k)) - \nabla f(x(k-1))] \\ &+ \frac{1}{2}\|x(k-1) - x^*\|_H^2 - \frac{1}{2}\|x(k) - x^*\|_H^2 - \frac{1}{2}\|x(k) - x(k-1)\|_H^2 \\ &+ \frac{1}{c}\|\alpha(k-1) - \alpha^*\|^2 - \frac{1}{c}\|\alpha(k) - \alpha^*\|^2 - \frac{1}{c}\|\alpha(k) - \alpha(k-1)\|^2. \end{aligned} \tag{22}$$

Next, we prove that there exists a contraction parameter

$$\delta = \min\left\{\frac{m_f - \frac{\theta}{2}}{\frac{(c\Gamma_u+\rho)^2}{2} + \frac{\mu M_f^2}{c\gamma_o}}, \frac{\frac{(c\gamma_u+\rho)^2}{2} - \frac{M_f^2}{2\theta}}{\frac{\mu}{\mu-1}\frac{(c\Gamma_u+\rho)^2}{2c\gamma_o} + \frac{\mu M_f^2}{c\gamma_o}}\right\} > 0, \tag{23}$$

such that it holds

$$\begin{aligned} &m_f\|x(k) - x^*\|^2 + \frac{1}{2}\|x(k) - x(k-1)\|_H^2 \\ &- [x(k) - x^*]^T[\nabla f(x(k)) - \nabla f(x(k-1))] \\ \geq &\frac{\delta}{2}\|x(k) - x^*\|_H^2 + \frac{\delta}{c}\|\alpha(k) - \alpha^*\|^2. \end{aligned} \tag{24}$$

In (23), μ is an arbitrary constant satisfying $\mu > 1$ and θ is an arbitrary constant satisfying $2m_f > \theta > M_f^2/(c\gamma_u + \rho)^2$; such a θ exists since by hypothesis $m_f(c\gamma_u + \rho)^2 > M_f^2/2$.

To prove (22), we find lower bounds for left-hand side terms and upper bounds for right-hand side terms. Since ∇f is Lipschitz continuous, for any $\mu > 1$ we have

$$M_f^2\|x(k-1)-x^*\|^2 \geq \|\nabla f(x(k-1))-\nabla f(x^*)\|^2 \quad (25)$$
$$= \|H(x(k-1)-x(k))-E_o^T(\alpha(k)-\alpha^*)\|^2$$
$$\geq \frac{1}{\mu}\|E_o^T(\alpha(k)-\alpha^*)\|^2 - \frac{1}{\mu-1}\|H(x(k-1)-x(k))\|^2.$$

Since the largest eigenvalue of H is $c\Gamma_u+\rho$ so we have $\|H(x(k-1)-x(k))\|^2 \leq (c\Gamma_u+\rho)^2\|(x(k-1)-x(k))\|^2$; also, since both $\alpha(k)$ and α^* lie in the column space of E_o and the smallest nonzero eigenvalue of $L_o=(E_o^TE^o)/2$ is γ_o it holds $\|E_o^T(\alpha(k)-\alpha^*)\|^2 \geq 2\gamma_o\|\alpha(k)-\alpha^*\|^2$. Using these inequalities (25) leads to

$$\frac{\delta}{2c\gamma_o}\mu M_f^2\|x(k-1)-x^*\|^2 \quad (26)$$
$$+\frac{\delta}{2c\gamma_o}\frac{\mu(c\Gamma_u+\rho)^2}{\mu-1}\|x(k-1)-x(k)\|^2 \geq \frac{\delta}{c}\|\alpha(k)-\alpha^*\|^2.$$

Again from the Lipschitz continuity of ∇f for any $\theta > 0$

$$-[x(k)-x^*]^T[\nabla f(x(k))-\nabla f(x(k-1))] \quad (27)$$
$$\geq -\frac{\theta}{2}\|x(k)-x^*\|^2 - \frac{1}{2\theta}\|\nabla f(x(k))-\nabla f(x(k-1))\|^2$$
$$\geq -\frac{\theta}{2}\|x(k)-x^*\|^2 - \frac{1}{2\theta}M_f^2\|x(k)-x(k-1)\|^2.$$

Since the largest and smallest eigenvalues of H are $c\Gamma_u+\rho$ and $c\gamma_u+\rho$ (which is positive by hypothesis), respectively, $\|x(k)-x(k-1)\|_H^2 \geq (c\gamma_u+\rho)^2\|x(k)-x(k-1)\|^2$ and $\|x(k)-x^*\|_H^2 \leq (c\Gamma_u+\rho)^2\|x(k)-x^*\|^2$. Combining these two inequalities as well as (26) and (27), the sufficient condition of (24) is

$$\left(m_f-\frac{\theta}{2}-\frac{\delta(c\Gamma_u+\rho)^2}{2}\right)\|x(k)-x^*\|^2+ \quad (28)$$
$$\left(\frac{(c\gamma_u+\rho)^2}{2}-\frac{1}{2\theta}M_f^2-\frac{\delta}{2c\gamma_o}\frac{\mu(c\Gamma_u+\rho)^2}{\mu-1}\right)\|x(k)-x(k-1)\|^2$$
$$\geq \frac{\delta}{2c\gamma_o}\mu M_f^2\|x(k-1)-x^*\|^2,$$

which is true for the contraction parameter $\delta > 0$ in (23) since $\|x(k-1)-x^*\|^2 \leq 2\|x(k)-x^*\|^2+2\|x(k)-x(k-1)\|^2$.

Combining (22) and (24) yields (14) and completes the proof. □

Lemma 1 demonstrates that $\frac{1}{2}\|x(k)-x^*\|_H^2+\frac{1}{c}\|\alpha(k)-\alpha^*\|^2$ decreases by a factor of at least $1/(1+\delta)$ from its previous value. When $\delta > 0$, Lemma 1 indicates Q-linear convergence of $\frac{1}{2}\|x(k)-x^*\|_H^2+\frac{1}{c}\|\alpha(k)-\alpha^*\|^2$ to 0. Noticing that H is positive definite as $c\gamma_u+\rho > 0$, Lemma 1 implies R-linear convergence of $x(k)$ to x^* since $\frac{1}{2}\|x(k)-x^*\|_H^2 \leq \frac{1}{2}\|x(k)-x^*\|_H^2+\frac{1}{c}\|\alpha(k)-\alpha^*\|^2$. The result is given in the following theorem.

Theorem 1 *Under the assumptions in Lemma 1, the primal solution $x(k)$ generated by the iterates* (5)-(7) *linearly converges to the unique optimal solution x^*.*

The contraction parameter δ is determined by Γ_u (the largest eigenvalue of the unoriented Laplacian L_u), γ_u (the smallest eigenvalue of the unoriented Laplacian L_u), γ_o (the smallest nonzero eigenvalue of the oriented Laplacian L_o), M_f (the Lipschitz constant of ∇f), m_f (the strong convexity constant of f), c and ρ (the decentralized LADMM parameters), as well as θ and μ (the two arbitrary constants satisfying $2m_f > \theta > M_f^2/(c\gamma_u+\rho)^2$ and $\mu > 1$). Larger contraction parameter means faster convergence.

4. NUMERICAL EXPERIMENTS

Consider a bidirectionally connected network composed of $n = 100$ agents where $m = 752$ arcs out of 9900 possible arcs are randomly chosen to be connected. Agent i measures a true signal $\tilde{x}^o \in \mathbb{R}^5$ through $y_i = U_i\tilde{x}^o + e_i$; here elements of $e_i \in \mathbb{R}^5$ follow zero-mean Gaussian distribution with standard deviation 0.1, elements of $U_i \in \mathbb{R}^{5\times 5}$ follow zero-mean Gaussian distribution with standard deviation 1, and $U_i^TU_i$ is positive definite. To run least squares regression, the local cost function of agent i is $f_i(\tilde{x}) = \|U_i\tilde{x}^o - y_i\|^2/2$.

Fig. 1 compares three decentralized algorithms: the decentralized LADMM, the decentralized ADMM, and the distributed gradient method (DGM). The decentralized LADMM adopts the iterations (10) and (11) where (10) reduces to simple algebraic operations. The decentralized ADMM is similar to that of the decentralized ADMM except that the iteration (10) is to minimize a quadratic function. The DGM updates agent i's local solution through $x_i(k) = \sum_{j\in\mathcal{N}_i\cup i} w_{ij}x_j(k-1) - \epsilon(k)\nabla f_i(x_i(k-1))$ where $\epsilon(k)$ is the stepsize and $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ is a weight matrix chosen with the maximum-degree rule [15]. The performance metric is average squared error defined as $\sum_{i=1}^n \|x_i(k)-\tilde{x}^*\|^2/n$.

The decentralized ADMM uses $c = 1.2$ that achieves the fastest convergence. The decentralized LADMM keeps $c = 1.2$ and uses different linearization parameters ρ. When $\rho = 8$, the decentralized LADMM converges to the optimal solution within 300 iterations that are comparable with 200 iterations needed by the decentralized ADMM. At the cost of slightly slower convergence, the decentralized LADMM is much easier to implement than its non-linearized counterpart due to its simple computations. The algorithm diverges when ρ is too small and converges slowly when ρ is too large, as Lemma 1 shows. The DGM can converge at a linear rate if the stepsize is fixed, but the solution is different than the optimal solution. Using a time-varying stepsize such as $\epsilon(k) = \epsilon(0)/k$ the DGM guarantees convergence to the optimal solution; $\epsilon(0) = 0.2$ is set to achieve the fastest convergence. However, the convergence is slow according to both theoretical analysis and numerical experiments.

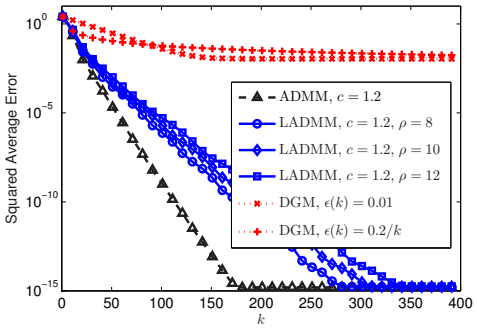

Fig. 1. Comparison of the three decentralized algorithms.

5. CONCLUSION

In this paper we develop the decentralized LADMM that modifies the decentralized ADMM through linearizing the local cost functions and reducing its computation burden. Its computation is similar to that of decentralized gradient descent but significantly improves the convergence properties. In summary, the decentralized LADMM bridges the gap between the primal domain and dual domain methods and takes advantages of the both.

6. REFERENCES

[1] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," IEEE Transactions on Signal Processing, vol. 58, pp. 5262–5276, 2010

[2] G. Giannakis, V. Kekatos, N. Gatsis, S. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," Manuscript available at http://arxiv.org/pdf/1302.0885v1.pdf

[3] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multi-hop wideband cognitive networks," IEEE Journal of Selected Topics in Signal Processing, vol. 5, pp. 37–48, 2011

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," IEEE Transactions on Automatic Control, vol. 54, pp. 48–61, 2009

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

[6] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," Manuscript available at http://arxiv.org/pdf/1112.2972v3.pdf

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

[8] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," In: Proceedings of SPAWC, 2005

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

[10] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links - Part I: distributed estimation of deterministic signals," IEEE Transactions on Signal Processing, vol. 56, pp. 350–364, 2008

[11] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," Manuscript available at http://arxiv.org/pdf/1307.5561v2.pdf

[12] 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

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

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

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