2015

Multi-Agent Distributed Optimization via Inexact Consensus ADMM

Tsung-Hui Chang
National Taiwan University of Science and Technology

Mingyi Hong
Iowa State University, mingyi@iastate.edu

Xiangfeng Wang
East China Normal University

Multi-Agent Distributed Optimization via Inexact Consensus ADMM

Abstract
Multi-agent distributed consensus optimization problems arise in many signal processing applications. Recently, the alternating direction method of multipliers (ADMM) has been used for solving this family of problems. ADMM based distributed optimization method is shown to have faster convergence rate compared with classic methods based on consensus subgradient, but can be computationally expensive, especially for problems with complicated structures or large dimensions. In this paper, we propose lowcomplexity algorithms that can reduce the overall computational cost of consensus ADMM by an order of magnitude for certain large-scale problems. Central to the proposed algorithms is the use of an inexact step for each ADMM update, which enables the agents to perform cheap computation at each iteration. Our convergence analyses show that the proposed methods converge well under some convexity assumptions. Numerical results show that the proposed algorithms offer considerably lower computational complexity than the standard ADMM based distributed optimization methods.

Keywords
distributed optimization, ADMM, consensus

Disciplines
Industrial Engineering | Systems Engineering

Comments
This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482, doi: 10.1109/TSP.2014.2367458. Posted with permission.

Multi-Agent Distributed Optimization via Inexact Consensus ADMM

Tsung-Hui Chang⋆, Mingyi Hong† and Xiangfeng Wang‡

Abstract

Multi-agent distributed consensus optimization problems arise in many signal processing applications. Recently, the alternating direction method of multipliers (ADMM) has been used for solving this family of problems. ADMM based distributed optimization method is shown to have faster convergence rate compared with classic methods based on consensus subgradient, but can be computationally expensive, especially for problems with complicated structures or large dimensions. In this paper, we propose low-complexity algorithms that can reduce the overall computational cost of consensus ADMM by an order of magnitude for certain large-scale problems. Central to the proposed algorithms is the use of an inexact step for each ADMM update, which enables the agents to perform cheap computation at each iteration. Our convergence analyses show that the proposed methods converge well under some convexity assumptions. Numerical results show that the proposed algorithms offer considerably lower computational complexity than the standard ADMM based distributed optimization methods.

Keywords– Distributed optimization, ADMM, Consensus

EDICS: OPT-DOPT, MLR-DIST, NET-DISP, SPC-APPL.

The work of Tsung-Hui Chang is supported by National Science Council, Taiwan (R.O.C.), under Grant NSC 102-2221-E-011-005-MY3. Part of this work is presented in IEEE ICASSP 2014.

⋆Tsung-Hui Chang is the corresponding author. Address: Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan, (R.O.C.). E-mail: tsunghui.chang@ieee.org.

†Mingyi Hong is with Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, E-mail: mhong@umn.edu

‡Xiangfeng Wang is with Shanghai Key Lab for Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, 200062, China, E-mail: xfwang@sei.ecnu.edu.cn

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482, doi: 10.1109/TSP.2014.2367458. Posted with permission.

I. INTRODUCTION

We consider a network with multiple agents, for example a sensor network, a data cloud network or a communication network. The agents seek to collaborate to accomplish certain task. For example, distributed database servers may cooperate for data mining or for parameter learning in order to fully exploit the data collected from individual servers [1]. Another example arises from large-scale machine learning applications [2], where a computation task may be executed by collaborative microprocessors with individual memories and storage spaces [2]–[4]. Distributed optimization becomes favorable as it is not always efficient to pool all the local information for centralized computation, due to large size of problem dimension, a large amount of local data, energy constraints and/or privacy issues [5]–[8]. Many of the distributed optimization tasks, such as those described above, can be cast as an optimization problem of the following form

$$(\mathsf{P1}) \quad \min_{\boldsymbol{y}\in\mathbb{R}^K} \ \sum_{i=1}^{N} \phi_i(\boldsymbol{y}) \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^K$ is the decision variable and $\phi_i : \mathbb{R}^K \to \mathbb{R} \cup \{\infty\}$ is the cost function associated with agent i. Here the function ϕ_i is composed of a smooth component $f_i : \mathbb{R}^M \to \mathbb{R} \cup \{\infty\}$ (possibly with extended values) and a non-smooth component $g_i : \mathbb{R}^K \to \mathbb{R} \cup \{\infty\}$, i.e.,

$$\phi_i(\boldsymbol{y}) = f_i(\boldsymbol{A}_i\boldsymbol{y}) + g_i(\boldsymbol{y}), \tag{2}$$

where $\boldsymbol{A}_i \in \mathbb{R}^{M\times K}$ is some data matrix not necessarily of full rank. Such model is common in practice: the smooth component usually represents the cost function to be minimized, while the non-smooth component is often used as a regularization function [9] or an indicator function representing that $\boldsymbol{y}$ is subject to a constraint set[1].

In the setting of distributed optimization, it is commonly assumed that each agent i only has knowledge about the local information f_i, g_i and $\boldsymbol{A}_i$. The challenge is to obtain, for each agent in the system, the optimal $\boldsymbol{x}$ of $(\mathsf{P1})$ using only local information and messages exchanged with neighbors [5]–[8].

[1]For example, if $\boldsymbol{y} \in \mathcal{X} \subseteq \mathbb{R}^K$ for some set $\mathcal{X}$, then this can be implicitly included in the nonsmooth component g_i by letting [10, Section 5]

$$g_i(\boldsymbol{y}) = \begin{cases} 0 & \text{if } \boldsymbol{y} \in \mathcal{X} \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

In addition to (P1), another common problem formulation has the following form

$$\text{(P2)} \quad \min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N \in \mathbb{R}^K} \ \sum_{i=1}^{N} \phi_i(\boldsymbol{x}_i) \ \text{ s.t. } \ \sum_{i=1}^{N} \mathbf{E}_i \boldsymbol{x}_i = \boldsymbol{q}, \tag{4}$$

where $\mathbf{E}_i \in \mathbb{R}^{M\times K}$, $\boldsymbol{q} \in \mathbb{R}^M$ and ϕ_i is given as in (2). Unlike (P1), in (P2), each agent i owns a local control variable[2] $\boldsymbol{x}_i \in \mathbb{R}^K$, and these variables are coupled together through the linear constraint. Examples of (P2) include the basis pursuit (BP) problem [11], [12], the network flow control problem [13] and interference management problem in communication networks [14]. To relate (P2) with (P1), let $\boldsymbol{\nu} \in \mathbb{R}^M$ be the Lagrange dual variable associated with the linear constraint $\sum_{i=1}^N \mathbf{E}_i \boldsymbol{x}_i = \boldsymbol{q}$. The Lagrange dual problem of (P2) can be equivalently written as

$$\min_{\boldsymbol{\nu} \in \mathbb{R}^M} \ \sum_{i=1}^{N} \left(\varphi_i(\boldsymbol{\nu}) + \frac{1}{N} \boldsymbol{\nu}^T \boldsymbol{q} \right) \tag{5}$$

where

$$\varphi_i(\boldsymbol{\nu}) = \max_{\boldsymbol{x}_i} \left\{ -\phi_i(\boldsymbol{x}_i) - \boldsymbol{\nu}^T \mathbf{E}_i \boldsymbol{x}_i \right\}, \ i = 1, \ldots, N. \tag{6}$$

Problem (5) thus has the same form as (P1). Given the optimal $\boldsymbol{\nu}$ of (5) and assuming that (P2) has a zero duality gap [15], each agent i can obtain the associated optimal variable $\boldsymbol{x}_i$ by solving (6). Therefore, a distributed optimization method that can solve (P1) may also be used for (P2) through solving (5).

There is an extensive literature on distributed consensus optimization methods, such as the consensus subgradient methods; see [5], [6] and the recent developments in [7], [8], [16], [17]. The consensus subgradient methods are appealing owing to their simplicity and the ability to handle a wide range of problems. However, the convergence of the consensus subgradient methods are usually slow.

Recently, the alternating direction method of multipliers (ADMM) [10], [18] has become popular for solving problems with forms of (P1) and (P2) in a distributed fashion. In [14], distributed transmission designs for multi-cellular wireless communications were developed based on ADMM. In [19], several ADMM based distributed optimization algorithms were developed for solving the sparse LASSO problem [20]. In [12], using a different consensus formulation from [19] and assuming the availability of a certain coloring scheme for the graph, ADMM is applied to solving the BP problem [11] for both row partitioned and column partitioned data models [16]. In [21], the methodologies proposed in [12] are extended to handling a more general class of problems with forms of (P1) and (P2). In [22], a distributed ADMM with a sequential update rule is proposed; while in [23], the method is extended and can be implemented

[2]Here we let all $\boldsymbol{x}_i$'s have the same dimension without loss of generality.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

asynchronously. The fast practical performance of ADMM is corroborated by its nice theoretical property. In particular, ADMM was found to converge linearly for a large class of problems [24], [25], meaning a certain optimality measure can decrease by a constant fraction in each iteration of the algorithm. In [26], [27], such fast convergence rate has also been built for distributed optimization.

It is important to note that existing ADMM based algorithms can be readily used to solve problems (P1) and (P2). For example, by applying the consensus formulation proposed in [19] and ADMM to (P1), a fully parallelized distributed optimization algorithm can be obtained (where the agents update their variables in a fully parallel manner), which we refer to as the consensus ADMM (C-ADMM). To solve (P2), the same consensus formulation and ADMM can be used on its Lagrange dual problem in (5), referred to as the dual consensus ADMM (DC-ADMM). The main drawback of these algorithms lies in the fact that each agent needs to repeatedly solve certain subproblems to *global optimality*. This can be computationally demanding, especially when the cost functions f_i's have complicated structures or when the problem size is large [2]. If a low-accuracy suboptimal solution is used for these subproblems instead, the convergence is no longer guaranteed.

The main objective of this paper is to study algorithms that can significantly reduce the computational burden for the agents. In particular, we propose two algorithms, named the inexact consensus ADMM (IC-ADMM) and the inexact dual consensus ADMM (IDC-ADMM'), both of which allow the agents to perform a single proximal gradient (PG) step [28] at each iteration. The benefit of the proposed approach lies in the fact that the PG step is usually simple, especially when g_i's are structured functions [9], [28]. Notably, the cheap iterations of the proposed algorithms is made possible by *inexactly* solving the subproblems arising in C-ADMM and DC-ADMM, in a way that is not known in the ADMM or consensus literature. For example, the proposed IC-ADMM approximates the smooth functions f_i's in C-ADMM, which is very different from the known inexact ADMM methods [29], [30], where only the quadratic penalty is approximated (thus does not always result in cheap PG steps). We summarize our main contributions below.

- For (P1), we propose an IC-ADMM method for reducing the computational complexity of C-ADMM. Conditions for global convergence of IC-ADMM are analyzed. Moreover, we show that IC-ADMM converges linearly, under similar conditions as in [26].
- For (P2), we first propose a DC-ADMM method which can globally solve (P2) for any connected graph and convex ϕ_i's. We further propose an IDC-ADMM method for reducing the computational burden of DC-ADMM. Conditions for global (linear) convergence are presented.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Numerical examples for solving distributed sparse logistic regression problems [31] will show that the proposed IC-ADMM and IDC-ADMM methods converge much faster than the consensus subgradient method [5]. Further, compared with the original C-ADMM and DC-ADMM, the proposed method can reduce the overall computational cost by an order of magnitude.

The paper is organized as follows. Section II presents the applications and assumptions. The C-ADMM and IC-ADMM are presented in Section III; while DC-ADMM and IDC-ADMM are presented in Section IV. Numerical results are given in Section V and conclusions are drawn in Section VI.

Notations: $\boldsymbol{A} \succeq \mathbf{0}$ ($\succ \mathbf{0}$) means that matrix $\boldsymbol{A}$ is positive semidefinite (positive definite). $\boldsymbol{I}_K$ is the $K \times K$ identity matrix; $\mathbf{1}_K$ is the K-dimensional all-one vector. $\|\boldsymbol{a}\|_2$ denotes the Euclidean norm of vector $\boldsymbol{a}$, and $\|\boldsymbol{z}\|_{\boldsymbol{A}}^2 \triangleq \boldsymbol{z}^T \boldsymbol{A} \boldsymbol{z}$ for some $\boldsymbol{A} \succeq \mathbf{0}$. Notation $\otimes$ denotes the Kronecker product. $\mathrm{diag}\{a_1, \ldots, a_N\}$ is a diagonal matrix with the ith diagonal element being a_i; while $\mathrm{blkdiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\}$ is a block diagonal matrix with the ith diagonal block matrix being $\boldsymbol{A}_i$. $\lambda_{\max}(\boldsymbol{A})$ and $\lambda_{\min}(\boldsymbol{A})$ denote the maximum and minimum eigenvalues of matrix $\boldsymbol{A}$, respectively.

II. Applications and Network Model

A. Application to Data Regression

As discussed in Section I, (P1) and (P2) arise in many problems in sensor networks, data networks and machine learning tasks. Here let us focus on the classical regression problems. We consider a general formulation that incorporates the LASSO [19] and logistic regression (LR) [31] as special instances. Let $\boldsymbol{A} = [\boldsymbol{A}_1^T, \ldots, \boldsymbol{A}_N^T]^T \in \mathbb{R}^{NM \times K}$ denote a regression data matrix, where $\boldsymbol{A}_i \in \mathbb{R}^{M \times K}$ for all $i = 1, \ldots, N$. For a row partitioned data (RPD) model [12, Fig. 1], [16], the distributed regression problem is given by

$$\min_{\boldsymbol{y} \in \mathbb{R}^K} \sum_{i=1}^{N} \Psi_i(\boldsymbol{y}; \boldsymbol{A}_i, \boldsymbol{b}_i), \tag{7}$$

where $\Psi_i(\boldsymbol{y}; \boldsymbol{A}_i, \boldsymbol{b}_i)$ is the cost function defined on the local regression data $\boldsymbol{A}_i$ and a local response signal $\boldsymbol{b}_i \in \mathbb{R}^M$. For example, the LASSO problem has $\Psi_i(\boldsymbol{y}; \boldsymbol{A}_i, \boldsymbol{b}_i) = \|\boldsymbol{b}_i - \boldsymbol{A}_i \boldsymbol{y}\|_2^2 + g_i(\boldsymbol{y})$. Similarly, for the LR problem, one has

$$\Psi_i(\boldsymbol{y}; \boldsymbol{A}_i, \boldsymbol{b}_i) = \sum_{m=1}^{M} \log\left(1 + \exp(-b_{im} \boldsymbol{a}_{im}^T \boldsymbol{y})\right) + g_i(\boldsymbol{y}), \tag{8}$$

where $\boldsymbol{A}_i = [\boldsymbol{a}_{i1}, \ldots, \boldsymbol{a}_{iM}]^T$ contains M training data vectors and $b_{im} \in \{\pm 1\}$ are binary labels for the training data. It is clear that (7) has the same form as (P1). Here, the non-smooth function g_i can be

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

1-norm for sparse regression, as well as mixture with an indicator functions specifying that $\boldsymbol{y}$ is confined in certain constraint set.

On the other hand, let $\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_N] \in \mathbb{R}^{M \times NK}$ denote a regression data matrix, where $\mathbf{E}_i \in \mathbb{R}^{M \times K}$ for all $i = 1, \ldots, N$. Then, for the column partitioned data (CPD) model [12, Fig. 1], [16], the distributed regression problem is formulated as

$$\min_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^K} \sum_{i=1}^{N} \Psi_i(\boldsymbol{x}_i; \mathbf{E}_i, \boldsymbol{b}), \tag{9}$$

where the response signal $\boldsymbol{b}$ is known to all agents while each agent i has a local regression variable $\boldsymbol{x}_i \in \mathbb{R}^K$ and local regression data matrix $\mathbf{E}_i = [\boldsymbol{e}_{i1}, \ldots, \boldsymbol{e}_{iM}]^T \in \mathbb{R}^{M \times K}$. For example, the LR problem has

$$\Psi_i(\boldsymbol{x}_i; \mathbf{E}_i, \boldsymbol{b}) = \sum_{m=1}^{M} \log\big(1 + \exp(-b_m \sum_{i=1}^{N} \boldsymbol{e}_{im}^T \boldsymbol{x}_i)\big) + g_i(\boldsymbol{x}_i). \tag{10}$$

By introducing a slack variable $\boldsymbol{z} = [z_1, \ldots, z_M]^T \triangleq \sum_{i=1}^{N} \mathbf{E}_i \boldsymbol{x}_i$, the CPD LR problem can be reformulated as

$$\begin{aligned} \min_{\substack{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^K, \\ \boldsymbol{z} \in \mathbb{R}^M}} & \left\{ \sum_{m=1}^{M} \log\big(1 + \exp(-b_m z_m)\big) + \sum_{i=1}^{N} g_i(\boldsymbol{x}_i) \right\} \\ \text{s.t. } & \textstyle\sum_{i=1}^{N} \mathbf{E}_i \boldsymbol{x}_i - \boldsymbol{z} = \mathbf{0}, \end{aligned} \tag{11}$$

which is an instance of (P2). In Section V, we will primarily test our algorithms on the RPD and CPD regression problems.

B. Network Model and Assumptions

Let an undirected graph $\mathcal{G}$ denote a multi-agent network, which contains a node set $V = \{1, \ldots, N\}$ and an edge set $\mathcal{E}$. An edge $(i, j) \in \mathcal{E}$ if and only if agent i and agent j can communicate with each other (i.e., neighbors). The edge set $\mathcal{E}$ defines an adjacency matrix $\boldsymbol{W} \in \{0, 1\}^{N \times N}$, where $[\boldsymbol{W}]_{i,j} = 1$ if $(i, j) \in \mathcal{E}$ and $[\boldsymbol{W}]_{i,j} = 0$ otherwise. In addition, one can define an index subset $\mathcal{N}_i = \{j \in V \mid (i, j) \in \mathcal{E}\}$ for the neighbors of each agent i, and a degree matrix $\boldsymbol{D} = \mathrm{diag}\{|\mathcal{N}_1|, \ldots, |\mathcal{N}_N|\}$ (a diagonal matrix). With $\boldsymbol{W}$ and $\boldsymbol{D}$, the Laplacian matrix of $\mathcal{G}$ is given by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ which is a positive semidefinite matrix (i.e., $\boldsymbol{L} \succeq \mathbf{0}$) and satisfies $\boldsymbol{L}\mathbf{1}_N = \mathbf{0}$ [32].

We make the following assumptions on $\mathcal{G}$ and problems (P1) and (P2).

Assumption 1 *The undirected graph $\mathcal{G}$ is connected.*

Assumption 1 implies that any two agents in the network can always influence each other in the long run. We also have the following assumptions on problems (P1) and (P2).

Assumption 2 *(a) In* (P1)*, the functions* $\phi_i : \mathbb{R}^K \to \mathbb{R} \cup \{\infty\}$ *are proper closed convex functions; at every* $\boldsymbol{y}$ *for which both* $f_i(\boldsymbol{A}_i\boldsymbol{y})$ *and* $g_i(\boldsymbol{y})$ *are well defined and* $\phi_i(\boldsymbol{y}) < \infty$*, there exists at least one bounded subgradient* $\partial\phi_i(\boldsymbol{y}) \in \mathbb{R}^K$ *such that* $\phi_i(\boldsymbol{x}) \geq \phi_i(\boldsymbol{y}) + (\partial\phi_i(\boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y})\ \forall \boldsymbol{x} \in \mathbb{R}^K$*. Moreover, the minimum of* (P1) *can be attained.*

(b) In (P2)*, the functions* $\phi_i : \mathbb{R}^K \to \mathbb{R} \cup \{\infty\}$ *are proper closed convex functions;* ϕ_i *has at least one bounded subgradient at every* $\boldsymbol{x}_i$ *for which both* $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)$ *and* $g_i(\boldsymbol{x}_i)$ *are well defined and* $\phi_i(\boldsymbol{x}_i) < \infty$*; the minimum of* (P2) *is attained and so is its optimal dual value; moreover, strong duality holds for* (P2)*.*

Assumption 3 *For all* $i \in V$*, the smooth function* f_i *in* (2) *is strongly convex, i.e., there exists some* $\sigma^2_{f,i} > 0$ *such that*

$$(\nabla f_i(\boldsymbol{y}) - \nabla f_i(\boldsymbol{x}))^T(\boldsymbol{y} - \boldsymbol{x}) \geq \sigma^2_{f,i}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2\ \forall \boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^M.$$

Moreover, f_i *has Lipschitz continuous gradients, i.e., there exists some* $L_{f,i} > 0$ *such that*

$$\|\nabla f_i(\boldsymbol{y}) - \nabla f_i(\boldsymbol{x})\|_2 \leq L_{f,i}\|\boldsymbol{y} - \boldsymbol{x}\|_2\ \ \forall \boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^M. \tag{12}$$

Note that, even under Assumption 3, $\phi_i(\boldsymbol{x}) = f_i(\boldsymbol{A}_i\boldsymbol{x}) + g_i(\boldsymbol{x})$ is not necessarily strongly convex in $\boldsymbol{x}$ since the matrix $\boldsymbol{A}_i$ can be fat and rank deficient. Both the LASSO problem [19] and the LR function in (8) satisfy Assumption 3 [3].

III. Distributed Consensus ADMM

In Section III-A, we briefly review the original C-ADMM [19] for solving (P1). In Section III-B, we propose a computationally efficient inexact C-ADMM method.

A. Review of C-ADMM

Under Assumption 1, (P1) can be equivalently written as

$$\min_{\substack{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N,\\ \{\boldsymbol{t}_{ij}\}}} \ \sum_{i=1}^{N} \phi_i(\boldsymbol{y}_i) \tag{13a}$$

$$\text{s.t. } \boldsymbol{y}_i = \boldsymbol{t}_{ij}\ \forall\ j \in \mathcal{N}_i,\ i \in V, \tag{13b}$$

$$\boldsymbol{y}_j = \boldsymbol{t}_{ij}\ \forall\ j \in \mathcal{N}_i,\ i \in V, \tag{13c}$$

[3]The logistic regression function $\log(1 + \exp(-x))$ is strongly convex given that x lies in a compact set.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

where $\{\boldsymbol{t}_{ij}\}$ are slack variables. According to (13), each agent i can optimize its local function $f_i(\boldsymbol{A}_i\boldsymbol{y}_i)+g_i(\boldsymbol{y}_i)$ with respect to a local copy of $\boldsymbol{y}$, i.e, $\boldsymbol{y}_i$, under the consensus constraints in (13b) and (13c). In [19], ADMM is employed to solve (13) in a distributed manner. Let $\{\boldsymbol{u}_{ij}\}$ and $\{\boldsymbol{v}_{ij}\}$ denote the Lagrange dual variables associated with constraints (13b) and (13c), respectively. According to [19], ADMM leads to the following iterative updates at each iteration k:

$$\boldsymbol{u}_{ij}^{(k)}=\boldsymbol{u}_{ij}^{(k-1)}+\frac{c}{2}(\boldsymbol{y}_i^{(k-1)}-\boldsymbol{y}_j^{(k-1)})\ \forall j\in\mathcal{N}_i, i\in V, \tag{14a}$$

$$\boldsymbol{v}_{ij}^{(k)}=\boldsymbol{v}_{ij}^{(k-1)}+\frac{c}{2}(\boldsymbol{y}_j^{(k-1)}-\boldsymbol{y}_i^{(k-1)})\ \forall j\in\mathcal{N}_i, i\in V, \tag{14b}$$

$$\begin{aligned}\boldsymbol{y}_i^{(k)}=\arg\min_{\boldsymbol{y}_i}\ &\Big\{\phi_i(\boldsymbol{y}_i)+\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)}+\boldsymbol{v}_{ji}^{(k)})^T\boldsymbol{y}_i\\ &+c\textstyle\sum_{j\in\mathcal{N}_i}\left\|\boldsymbol{y}_i-\frac{\boldsymbol{y}_i^{(k-1)}+\boldsymbol{y}_j^{(k-1)}}{2}\right\|_2^2\Big\}\ \forall i\in V,\end{aligned} \tag{14c}$$

where $c>0$ is a penalty parameter and $\boldsymbol{u}_{ij}^{(0)}+\boldsymbol{v}_{ij}^{(0)}=\boldsymbol{0}\ \forall i,j$. Note that variables $\{\boldsymbol{t}_{ij}^{(k)}\}$ are not shown in (14) as they can be expressed by variables $\{\boldsymbol{y}_i^{(k-1)}\}$; see [19] for the details.

The updates in (14) are useful for convergence analysis. For practical implementation, we define $\boldsymbol{p}_i^{(k)}\triangleq\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)}+\boldsymbol{v}_{ji}^{(k)})$, $i\in V$. Then, (14) boils down to Algorithm 1.

Algorithm 1 C-ADMM for solving (P1)

1: **Given** initial variables $\boldsymbol{y}_i^{(0)}\in\mathbb{R}^K$ and $\boldsymbol{p}_i^{(0)}=\boldsymbol{0}$ for each agent i, $i\in V$. Set $k=1$.

2: **repeat**

3: For all $i\in V$ (in parallel), $\boldsymbol{p}_i^{(k)}=\boldsymbol{p}_i^{(k-1)}+c\sum_{j\in\mathcal{N}_i}(\boldsymbol{y}_i^{(k-1)}-\boldsymbol{y}_j^{(k-1)})$,

$$\boldsymbol{y}_i^{(k)}=\arg\min_{\boldsymbol{y}_i}\ \Big\{f_i(\boldsymbol{A}_i\boldsymbol{y}_i)+g_i(\boldsymbol{y}_i)+\boldsymbol{y}_i^T\boldsymbol{p}_i^{(k)}+c\textstyle\sum_{j\in\mathcal{N}_i}\left\|\boldsymbol{y}_i-\frac{\boldsymbol{y}_i^{(k-1)}+\boldsymbol{y}_j^{(k-1)}}{2}\right\|_2^2\Big\}. \tag{15}$$

4: **Set** $k=k+1$.

5: **until** a predefined stopping criterion (e.g., a maximum iteration number) is satisfied.

It is important to note from Step 4 and Step 5 of Algorithm 1 that, except for the parameter c which has to be universally known, each agent i updates the variables $(\boldsymbol{y}_i^{(k)},\boldsymbol{p}_i^{(k)})$ in a fully parallel manner, by only using the local function ϕ_i and messages $\{\boldsymbol{y}_j^{(k-1)}\}_{j\in\mathcal{N}_i}$, which come from its direct neighbors. It has been shown in [19] that, under Assumptions 1 and 2, C-ADMM is guaranteed to converge for any

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

$c > 0$[4]:

$$\lim_{k\to\infty} \boldsymbol{y}_i^{(k)} = \boldsymbol{y}^\star, \ \ \lim_{k\to\infty} (\boldsymbol{u}_{ij}^{(k)}, \boldsymbol{v}_{ij}^{(k)}) = (\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star), \ \forall j, i, \tag{16}$$

where $\boldsymbol{y}^\star \triangleq \boldsymbol{y}_1^\star = \cdots = \boldsymbol{y}_N^\star$ and $\{\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star\}$ denote a pair of optimal primal and dual solutions to problem (13), and $\boldsymbol{y}^\star$ is optimal to (P1). It is also shown that C-ADMM can converge linearly when ϕ_i's are purely smooth (i.e., $g_i(\boldsymbol{y}_i) = 0 \ \forall i$) and strongly convex with respect to $\boldsymbol{y}_i$ [26].

One key issue about C-ADMM is that the subproblem in (15) is not always easy to solve. For instance, for the LR function in (8), the associated subproblem (15) is given by

$$\begin{aligned} \boldsymbol{y}_i^{(k)} = \arg \ \min_{\boldsymbol{y}_i} \ \Bigg\{ & \sum_{m=1}^{M} \log\left(1 + \exp(-b_{im}\boldsymbol{a}_{im}^T \boldsymbol{y}_i)\right) + g_i(\boldsymbol{y}_i) \\ & + \boldsymbol{y}_i^T \boldsymbol{p}_i^{(k)} + c \sum_{j\in\mathcal{N}_i} \Big\| \boldsymbol{y}_i - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2} \Big\|_2^2 \Bigg\}. \end{aligned} \tag{17}$$

As seen, due to the complicated LR cost, problem (17) cannot yield simple solutions, and a numerical solver has to be employed. Clearly, obtaining a high-accuracy solution of (17) can be computationally expensive, especially when the problem dimension or the number of training data is large. While a low-accuracy solution to (17) can be adopted for complexity reduction, it may destroy the convergence behavior of C-ADMM, as will be shown in Section V.

B. Proposed Inexact C-ADMM

To reduce the complexity of C-ADMM, instead of solving subproblem (15) directly, we consider the following update:

$$\begin{aligned} \boldsymbol{y}_i^{(k)} = \arg \ \min_{\boldsymbol{y}_i} \ \Big\{ & \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)})^T \boldsymbol{A}_i (\boldsymbol{y}_i - \boldsymbol{y}_i^{(k-1)}) \\ & + \frac{\beta_i}{2} \|\boldsymbol{y}_i - \boldsymbol{y}_i^{(k-1)}\|_2^2 + g_i(\boldsymbol{y}_i) + \boldsymbol{y}_i^T \boldsymbol{p}_i^{(k)} + c \textstyle\sum_{j\in\mathcal{N}_i} \Big\| \boldsymbol{y}_i - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2} \Big\|_2^2 \Big\}. \end{aligned} \tag{18}$$

In (18) we have replaced the smooth cost function $f_i(\boldsymbol{A}_i \boldsymbol{y}_i)$ in (15) with a proximal first-order approximation around $\boldsymbol{y}_i^{(k-1)}$:

$$\nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)})^T \boldsymbol{A}_i (\boldsymbol{y}_i - \boldsymbol{y}_i^{(k-1)}) + \frac{\beta_i}{2} \|\boldsymbol{y}_i - \boldsymbol{y}_i^{(k-1)}\|_2^2,$$

where $\beta_i > 0$ is a penalty parameter of the proximal quadratic term. To obtain a concise representation of $\boldsymbol{y}_i^{(k)}$, let us define the *proximity operator* for the non-smooth function g_i at a given point $\boldsymbol{s} \in \mathbb{R}^K$ as

[4]In general, the parameter c is chosen empirically. Only for some special instance, optimal c may be analytically found; e.g., see [33].

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

[28]

$$\text{prox}_{g_i}^{\gamma_i}[\boldsymbol{s}] \triangleq \arg \min_{\boldsymbol{y}} \left\{ g_i(\boldsymbol{y}) + \frac{\gamma_i}{2}\|\boldsymbol{y} - \boldsymbol{s}\|_2^2 \right\}, \tag{19}$$

where $\gamma_i = \beta_i + 2c|\mathcal{N}_i|$. Clearly, using this definition, (18) can be expressed more compactly as

$$\begin{aligned} \boldsymbol{y}_i^{(k)} &= \arg\min_{\boldsymbol{y}_i} \left\{ g_i(\boldsymbol{y}) + \frac{\gamma_i}{2} \left\| \boldsymbol{y}_i - \frac{1}{\gamma_i}\big(\beta_i \boldsymbol{y}_i^{(k-1)} - \boldsymbol{p}_i^{(k)} \right. \right. \\ &\quad \left. \left. - \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) + c \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)})\big) \right\|_2^2 \right\} \\ &= \text{prox}_{g_i}^{\gamma_i}\left[\frac{1}{\gamma_i}\left(\beta_i \boldsymbol{y}_i^{(k-1)} - \boldsymbol{p}_i^{(k)} - \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) \right.\right. \\ &\quad \left.\left. + c \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}) \right) \right], \end{aligned} \tag{20}$$

which is a proximal gradient (PG) update.

The PG updates like (20) often admit closed-form expression, especially when g_i's are functions including the ℓ_1 norm, Euclidean norm, infinity norm and matrix nuclear norm [34]. For example, when $g_i(\boldsymbol{y}) = \|\boldsymbol{y}\|_1$, (19) has a closed-form solution known as the soft thresholding operator [28], [34]:

$$\mathcal{S}\left[\boldsymbol{s}, \tfrac{1}{\gamma_i}\right] = \left(\boldsymbol{s} - \tfrac{1}{\gamma_i}\mathbf{1}_K\right)^+ + \left(-\boldsymbol{s} - \tfrac{1}{\gamma_i}\mathbf{1}_K\right)^+, \tag{21}$$

where $(x)^+ \triangleq \max\{x, 0\}$. The IC-ADMM is presented in Algorithm 2.

Algorithm 2 Proposed IC-ADMM for solving (P1)

1: **Given** initial variables $\boldsymbol{y}_i^{(0)} \in \mathbb{R}^K$ and $\boldsymbol{p}_i^{(0)} = \mathbf{0}$ for each agent i, $i \in V$. Set $k = 1$.

2: **repeat**

3: For all $i \in V$ (in parallel),

$$\boldsymbol{p}_i^{(k)} = \boldsymbol{p}_i^{(k-1)} + c \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{y}_i^{(k-1)} - \boldsymbol{y}_j^{(k-1)}),$$

$$\begin{aligned} \boldsymbol{y}_i^{(k)} &= \text{prox}_{g_i}^{\gamma_i}\left[\frac{1}{\gamma_i}\left(\beta_i \boldsymbol{y}_i^{(k-1)} - \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) \right.\right. \\ &\quad \left.\left. - \boldsymbol{p}_i^{(k)} + c \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}) \right) \right]. \end{aligned} \tag{22}$$

4: **Set** $k = k + 1$.

5: **until** a predefined stopping criterion (e.g., a maximum iteration number) is satisfied.

Although the idea of "inexact ADMM" is not new, our approach is significantly different from the existing methods [29], [30], where the inexact update is obtained by approximating the quadratic penalization term only. It can be seen that problem (17) is still difficult to solve even the inexact update

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

in [29], [30] is applied. Two notable exceptions are the algorithms proposed in [35] and [36] where the cost function is also linearized. However, an additional back substitution step and two extragradient steps are required in [35] and [36], respectively, which is not suited for distributed optimization.

The convergence properties of IC-ADMM is characterized by the following theorem.

Theorem 1 *Suppose that Assumptions 1, 2(a) and 3 hold. Let*

$$\beta_i > \frac{L_{f,i}^2}{\sigma_{f,i}^2}\lambda_{\max}(\boldsymbol{A}_i^T\boldsymbol{A}_i) - c\lambda_{\min}(\boldsymbol{D}+\boldsymbol{W}) > 0 \ \forall i \in V, \tag{23}$$

and let $\boldsymbol{y}^\star \triangleq \boldsymbol{y}_1^\star = \cdots = \boldsymbol{y}_N^\star$ *and* $\{\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star\}$ *denote a pair of optimal primal and dual solutions to problem* (13) *(i.e.,* (P1)*).*

(a) For Algorithm 2, $\boldsymbol{y}_1^{(k)}, \ldots, \boldsymbol{y}_N^{(k)}$ *converge to a common point* $\boldsymbol{y}^\star$*.*

(b) If $\phi_i(\boldsymbol{y}) = f_i(\boldsymbol{A}_i\boldsymbol{y})$*, where* $\boldsymbol{A}_i$ *has full column rank, for all* $i \in V$*, then we have*

$$\lim_{k\to\infty} \|\boldsymbol{y}^{(k)} - \mathbf{1}_N \otimes \boldsymbol{y}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}} + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 = 0 \text{ linearly},$$

where $\boldsymbol{y}^{(k)} = [(\boldsymbol{y}_1^{(k)})^T, \ldots, (\boldsymbol{y}_N^{(k)})^T]^T$*;* $\boldsymbol{u}_i^{(k)} \in \mathbb{R}^{K|\mathcal{N}_i|}$ *(*$\boldsymbol{u}_i^\star$*) is a vector that stacks* $\boldsymbol{u}_{ij}^{(k)}$ *(*$\boldsymbol{u}_{ij}^\star$*)* $\forall j \in \mathcal{N}_i$*;* $\boldsymbol{u}^{(k)} \in \mathbb{R}^{K\sum_{i=1}^N |\mathcal{N}_i|}$ *(*$\boldsymbol{u}^\star$*) stacks* $\boldsymbol{u}_i^{(k)}$ *(*$\boldsymbol{u}_i^\star$*)* $\forall i = 1, \ldots, N$*. and*

$$\boldsymbol{G} \triangleq \boldsymbol{D}_\beta + c((\boldsymbol{D}+\boldsymbol{W}) \otimes \boldsymbol{I}_K) \succ \mathbf{0}, \tag{24}$$

$$\mathbf{M} \triangleq \tilde{\boldsymbol{A}}^T(\boldsymbol{D}_{\sigma_f} - \frac{1}{2}\boldsymbol{D}_\rho)\tilde{\boldsymbol{A}} \succ \mathbf{0}, \tag{25}$$

for some $0 < \alpha < 1$ *and* $\rho > 0$*. Here,* $\tilde{\boldsymbol{A}} = \mathrm{blkdiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\}$*;* $\boldsymbol{D}_\beta = \mathrm{diag}\{\beta_1, \ldots, \beta_N\} \otimes \boldsymbol{I}_K$*;* $\boldsymbol{D}_{\sigma_f} = \mathrm{diag}\{\sigma_{f,1}^2, \ldots, \sigma_{f,N}^2\} \otimes \boldsymbol{I}_K$*; and* $\boldsymbol{D}_\rho = \mathrm{diag}\{\rho_1, \ldots, \rho_N\} \otimes \boldsymbol{I}_K$*.*

The proof is presented in Appendix A. Theorem 1 implies that, given sufficiently large β_i's, IC-ADMM not only achieves consensus and optimality, but also converges linearly provided that ϕ_i is purely smooth and strongly convex. Note that, to ensure (23), the global knowledge of $\lambda_{\min}(\boldsymbol{D}+\boldsymbol{W})$ is required by all agents. As a parallel work, we should mention that a concurrent result similar as Theorem 1(b) is presented in [37].

Remark 1 We remark that the convergence condition in (23) depends on the network topology. Let $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ denote the Laplacian matrix of $\mathcal{G}$. Then $\boldsymbol{D}+\boldsymbol{W} = 2\boldsymbol{D} - \boldsymbol{L}$. By the graph theory [32], the normalized Laplacian matrix, i.e., $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$, must have $\lambda_{\max}(\tilde{\boldsymbol{L}}) \leq 2$. Further, $\lambda_{\max}(\tilde{\boldsymbol{L}}) < 2$ if and only if the connected graph $\mathcal{G}$ is not bipartite. Thus, we have $\lambda_{\min}(\boldsymbol{D}+\boldsymbol{W}) = \lambda_{\min}(\boldsymbol{D}^{\frac{1}{2}}(2\boldsymbol{I}_N - \tilde{\boldsymbol{L}})\boldsymbol{D}^{\frac{1}{2}}) \geq 0$, and $\lambda_{\min}(\boldsymbol{D}+\boldsymbol{W}) > 0$ whenever $\mathcal{G}$ is non-bipartite.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

IV. Distributed Dual Consensus ADMM

In this section, we turn the focus to (P2). In Section IV-A, we present a DC-ADMM method for solving (P2). In Section IV-B, an inexact DC-ADMM method is proposed.

A. Proposed DC-ADMM

The DC-ADMM is obtained by applying the C-ADMM (Algorithm 1) to problem (5) which is equivalent to the Lagrange dual of (P2). Firstly, similar to (13), we write problem (5) as

$$\min_{\substack{\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_N \\ \{\boldsymbol{t}_{ij}\}}} \sum_{i=1}^{N} \left(\varphi_i(\boldsymbol{\nu}_i) + \frac{1}{N}\boldsymbol{\nu}_i^T \boldsymbol{q} \right) \tag{26a}$$

$$\text{s.t. } \boldsymbol{\nu}_i = \boldsymbol{t}_{ij},\ \boldsymbol{\nu}_j = \boldsymbol{t}_{ij}\ \forall\, j \in \mathcal{N}_i,\ i \in V, \tag{26b}$$

where $\boldsymbol{\nu}_i \in \mathbb{R}^M$ is the ith agent's local copy of the dual variable $\boldsymbol{\nu}$ and φ_i is given in (6). Following a similar argument as in deriving Algorithm 1, we obtain the following update steps at each iteration k

$$\boldsymbol{p}_i^{(k)} = \boldsymbol{p}_i^{(k-1)} + c \textstyle\sum_{j\in\mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}), \tag{27a}$$

$$\begin{aligned} \boldsymbol{\nu}_i^{(k)} = \arg\min_{\boldsymbol{\nu}_i\in\mathbb{R}^M} \bigg\{ &\varphi_i(\boldsymbol{\nu}_i) + \frac{1}{N}\boldsymbol{\nu}_i^T\boldsymbol{q} + \boldsymbol{\nu}_i^T\boldsymbol{p}_i^{(k)} \\ &+ c \textstyle\sum_{j\in\mathcal{N}_i} \left\| \boldsymbol{\nu}_i - \frac{\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)}}{2} \right\|_2^2 \bigg\}\ \forall\, i \in V, \end{aligned} \tag{27b}$$

where, with a slight abuse of notation,

$$\boldsymbol{p}_i^{(k)} = \textstyle\sum_{j\in\mathcal{N}_i} (\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)}), \tag{28}$$

in which $\{\boldsymbol{u}_{ij}\}$ and $\{\boldsymbol{v}_{ij}\}$ are dual variables associated with the two constraints in (26b) and are updated in a similar fashion as in (14a) and (14b), i.e.,

$$\boldsymbol{u}_{ij}^{(k)} = \boldsymbol{u}_{ij}^{(k-1)} + \frac{c}{2}(\boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)})\ \forall j \in \mathcal{N}_i, i \in V, \tag{29a}$$

$$\boldsymbol{v}_{ij}^{(k)} = \boldsymbol{v}_{ij}^{(k-1)} + \frac{c}{2}(\boldsymbol{\nu}_j^{(k-1)} - \boldsymbol{\nu}_i^{(k-1)})\ \forall j \in \mathcal{N}_i, i \in V. \tag{29b}$$

In general, subproblem (27b) is not easy to handle because φ_i is implicit and (27b) is in fact a min-max optimization problem given by

$$\begin{aligned} \boldsymbol{\nu}_i^{(k)} = \arg\min_{\boldsymbol{\nu}_i}\max_{\boldsymbol{x}_i} \bigg\{ &-\phi_i(\boldsymbol{x}_i) - \boldsymbol{\nu}_i^T\mathbf{E}_i\boldsymbol{x}_i + \frac{1}{N}\boldsymbol{\nu}_i^T\boldsymbol{q} \\ &+ \boldsymbol{\nu}_i^T\boldsymbol{p}_i^{(k)} + c\textstyle\sum_{j\in\mathcal{N}_i} \left\| \boldsymbol{\nu}_i - \frac{\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)}}{2} \right\|_2^2 \bigg\}. \end{aligned} \tag{30}$$

Fortunately, since the objective function in (30) is convex in $\boldsymbol{\nu}_i$ for any $\boldsymbol{x}_i$ and is concave in $\boldsymbol{x}_i$ for any $\boldsymbol{\nu}_i$, the minimax theorem [38, Proposition 2.6.2] can be applied so that the min-max problem (30) can

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

be equivalently solved by considering its max-min counterpart and saddle point exists. Specifically, the max-min counterpart of (30) is given by

$$\max_{\boldsymbol{x}_i}\min_{\boldsymbol{\nu}_i}\Big\{-\phi_i(\boldsymbol{x}_i)-\boldsymbol{\nu}_i^T\mathbf{E}_i\boldsymbol{x}_i+\frac{1}{N}\boldsymbol{\nu}_i^T\boldsymbol{q}+\boldsymbol{\nu}_i^T\boldsymbol{p}_i^{(k)} + c\textstyle\sum_{j\in\mathcal{N}_i}\big\|\boldsymbol{\nu}_i-\frac{\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)}}{2}\big\|_2^2\Big\} \tag{31}$$

$$=\max_{\boldsymbol{x}_i}\min_{\boldsymbol{\nu}_i}\Big\{-\phi_i(\boldsymbol{x}_i)+(c|\mathcal{N}_i|)\Big\|\boldsymbol{\nu}_i-\tfrac{1}{2|\mathcal{N}_i|}\big[\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)})-\frac{1}{c}\boldsymbol{p}_i^{(k)}+\frac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i-\frac{1}{N}\boldsymbol{q})\big]\Big\|_2^2 - \tfrac{c}{4|\mathcal{N}_i|}\Big\|\tfrac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i-\tfrac{1}{N}\boldsymbol{q}) - \tfrac{1}{c}\boldsymbol{p}_i^{(k)}+\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)})\Big\|_2^2\Big\} \tag{32}$$

where the equality is obtained by completing the quadratic term of $\boldsymbol{\nu}_i$. Let $\boldsymbol{x}_i^{(k)}$ be an inner maximizer of (30) so that $(\boldsymbol{\nu}_i^{(k)},\boldsymbol{x}_i^{(k)})$ is a saddle point of (30). Then, $(\boldsymbol{x}_i^{(k)},\boldsymbol{\nu}_i^{(k)})$ is a pair of outer-inner solution to (31) and (32) [38, Proposition 2.6.1]. From (32), the inner minimizer $\boldsymbol{\nu}_i^{(k)}$ can be uniquely determined by

$$\boldsymbol{\nu}_i^{(k)}=\tfrac{1}{2|\mathcal{N}_i|}\big[\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)})-\frac{1}{c}\boldsymbol{p}_i^{(k)} + \frac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i^{(k)}-\frac{1}{N}\boldsymbol{q})\big], \tag{33}$$

and that the outer maximizer is given by

$$\boldsymbol{x}_i^{(k)}=\arg\ \min_{\boldsymbol{x}_i}\ \Big\{\phi_i(\boldsymbol{x}_i)+\frac{c}{4|\mathcal{N}_i|}\big\|\tfrac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i-\tfrac{1}{N}\boldsymbol{q}) - \tfrac{1}{c}\boldsymbol{p}_i^{(k)}+\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)}+\boldsymbol{\nu}_j^{(k-1)})\big\|_2^2\Big\}. \tag{34}$$

As a result, the min-max subproblem (27b) can actually be obtained by first solving the subproblem (34) with respect to the primal variable $\boldsymbol{x}_i$ followed by evaluating $\boldsymbol{\nu}_i^{(k)}$ using the close-form in (33). The proposed DC-ADMM is summarized in Algorithm 3.

Interestingly, while DC-ADMM handles the equivalent dual problem in (5), it directly yields primal optimal solution of (P2), as we state in the following theorem.

Theorem 2 *Suppose that Assumptions 1 and 2(b) hold. Then* $(\boldsymbol{\nu}_1^{(k)},\ldots,\boldsymbol{\nu}_N^{(k)})$ *converges to a common point* $\boldsymbol{\nu}^\star$*, which is optimal to the dual problem* (5)*. Moreover, any limit point of* $(\boldsymbol{x}_1^{(k)},\ldots,\boldsymbol{x}_N^{(k)})$ *is primal optimal to* (P2).

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Algorithm 3 Proposed DC-ADMM for solving (P2)

1: **Given** initial variables $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^K$, $\boldsymbol{\nu}_i^{(0)} \in \mathbb{R}^M$ and $\boldsymbol{p}_i^{(0)} = \mathbf{0}$ for each agent i, $i \in V$. Set $k = 1$.

2: **repeat**

3: For all $i \in V$ (in parallel),

$$\boldsymbol{p}_i^{(k)} = \boldsymbol{p}_i^{(k-1)} + c \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}),$$

$$\begin{aligned} \boldsymbol{x}_i^{(k)} = \arg \min_{\boldsymbol{x}_i} \bigg\{ &\phi_i(\boldsymbol{x}_i) + \frac{c}{4|\mathcal{N}_i|} \big\| \tfrac{1}{c}(\mathbf{E}_i \boldsymbol{x}_i - \tfrac{1}{N}\boldsymbol{q}) \\ &- \tfrac{1}{c}\boldsymbol{p}_i^{(k)} + \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) \big\|_2^2 \bigg\}, \end{aligned} \tag{35}$$

$$\begin{aligned} \boldsymbol{\nu}_i^{(k)} = &\tfrac{1}{2|\mathcal{N}_i|} \big(\textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) - \tfrac{1}{c}\boldsymbol{p}_i^{(k)} \\ &+ \tfrac{1}{c}(\mathbf{E}_i \boldsymbol{x}_i^{(k)} - \tfrac{1}{N}\boldsymbol{q}) \big). \end{aligned} \tag{36}$$

4: **Set** $k = k + 1$.

5: **until** a predefined stopping criterion is satisfied.

Proof: Since DC-ADMM is a direct application of C-ADMM to the dual problem (5), it follows from [19] that as $k \to \infty$,

$$\boldsymbol{\nu}_i^{(k)} \to \boldsymbol{\nu}^\star,\ \boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_j^{(k)} \to \mathbf{0}\ \forall j \in \mathcal{N}_i,\ i \in V. \tag{37}$$

What remains is to show that any limit point of $(\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_N^{(k)})$ is asymptotically optimal to (P2), i.e., as $k \to \infty$,

$$\partial \phi_i(\boldsymbol{x}_i^{(k)}) + \mathbf{E}_i^T \boldsymbol{\nu}^\star \to \mathbf{0}\ \forall i \in V, \tag{38}$$

$$\textstyle\sum_{i=1}^N \mathbf{E}_i \boldsymbol{x}_i^{(k)} - \boldsymbol{q} \to \mathbf{0}. \tag{39}$$

To show (38), consider the optimality condition of (34), i.e.,

$$\begin{aligned} \mathbf{0} &= \partial \phi_i(\boldsymbol{x}_i^{(k)}) + \frac{1}{2|\mathcal{N}_i|} \mathbf{E}_i^T \bigg(\frac{1}{c}(\mathbf{E}_i \boldsymbol{x}_i^{(k)} - \frac{1}{N}\boldsymbol{q}) \\ &\qquad - \tfrac{1}{c}\boldsymbol{p}_i^{(k)} + \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) \bigg) \\ &= \partial \phi_i(\boldsymbol{x}_i^{(k)}) + \mathbf{E}_i^T \boldsymbol{\nu}_i^{(k)}, \end{aligned} \tag{40}$$

where the second equality is obtained by (33). Since (40) holds for all k and $\boldsymbol{\nu}_i^{(k)} \to \boldsymbol{\nu}^\star$ by (37), (38) is true when $k \to \infty$.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

To show (39), rewrite (33) as follows

$$\begin{aligned}\mathbf{0} = &-(\mathbf{E}_i\boldsymbol{x}_i^{(k)} - \frac{1}{N}\boldsymbol{q}) + 2c\textstyle\sum_{j\in\mathcal{N}_i}\left(\boldsymbol{\nu}_i^{(k)} - \frac{\boldsymbol{\nu}_i^{(k)}+\boldsymbol{\nu}_j^{(k)}}{2}\right)\\ &+\boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)})\\ = &-(\mathbf{E}_i\boldsymbol{x}_i^{(k)} - \frac{1}{N}\boldsymbol{q}) + \boldsymbol{p}_i^{(k+1)}\\ &+ c\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}),\end{aligned} \tag{41}$$

where the last equality is obtained by (28) and (29). Upon summing (41) for $i = 1, \ldots, N$, and by the fact that

$$\sum_{i=1}^N \boldsymbol{p}_i^{(k)} = \sum_{i=1}^N\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)}) = \mathbf{0}$$

(by applying (A.13) and (A.14) in Appendix A), we can obtain

$$\sum_{i=1}^N \mathbf{E}_i\boldsymbol{x}_i^{(k)} - \boldsymbol{q} = c\sum_{i=1}^N\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}). \tag{42}$$

Note that $\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)} \to \mathbf{0}\ \forall i \in V$ as inferred from $\boldsymbol{\nu}_i^{(k)} \to \boldsymbol{\nu}^\star\ \forall i \in V$ in (37). By applying this fact to (42), we obtain that (39) is true as $k \to \infty$. ■

Interestingly, from (42), one observes that the primal feasibility of $(\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_N^{(k)})$ to (P2) depends on the agents' consensus on the dual variable $\boldsymbol{\nu}$.

We remark that Algorithm 3 is different from the D-ADMM algorithm in [12, Algorithm 3]. Firstly, Algorithm 3 can be implemented in a fully parallel manner; secondly, Algorithm 3 does not involve solving a min-max subproblem at each iteration; thirdly, convergence of Algorithm 3 can be achieved without the assumption that the graph $\mathcal{G}$ is bipartite.

B. *Proposed Inexact DC-ADMM*

In this subsection, we propose an inexact version of DC-ADMM, referred to as the IDC-ADMM. In view of the fact that solving the subproblem in (35) can be expensive, we consider an inexact update of $\boldsymbol{x}_i^{(k)}$. Specifically, since a non-trivial $\mathbf{E}_i$ can also complicate the solution[5], we propose to approximate both $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)$ and the quadratic term $\frac{c}{4|\mathcal{N}_i|}\|\frac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i - \frac{1}{N}\boldsymbol{q}) - \frac{1}{c}\boldsymbol{p}_i^{(k)} + \sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)})\|_2^2$ in (35)

[5]When $\mathbf{E}_i$ has orthogonal columns (e.g., $\mathbf{E}_i^T\mathbf{E}_i = \alpha\boldsymbol{I}_K$ for some $\alpha \in \mathbb{R}$), then it may not be necessary to approximate the quadratic term.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

by a proximal first-order approximation around $\boldsymbol{x}_i^{(k-1)}$; this leads to the following update

$$
\begin{aligned}
\boldsymbol{x}_i^{(k)} = \arg\min_{\boldsymbol{x}_i} \bigg\{ \bigg[& \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{x}_i^{(k-1)}) + \tfrac{1}{2|\mathcal{N}_i|} \mathbf{E}_i^T \Big(\tfrac{1}{c} (\mathbf{E}_i \boldsymbol{x}_i^{(k-1)} \\
& - \tfrac{1}{N} \boldsymbol{q}) - \tfrac{1}{c} \boldsymbol{p}_i^{(k)} + \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) \Big) \bigg]^T (\boldsymbol{x}_i - \boldsymbol{x}_i^{(k-1)}) \\
& + \frac{\beta_i}{2} \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k-1)}\|_2^2 + g_i(\boldsymbol{x}_i) \bigg\},
\end{aligned} \tag{43}
$$

where, with a slight abuse of notation, $\beta_i > 0$ is a penalty parameter. By (19), equation (43) can be further written as the following PG update

$$
\begin{aligned}
\boldsymbol{x}_i^{(k)} &= \arg\min_{\boldsymbol{x}_i} \bigg\{ \tfrac{\beta_i}{2} \bigg\| \boldsymbol{x}_i - \bigg[\boldsymbol{x}_i^{(k-1)} - \tfrac{1}{\beta_i} \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{x}_i^{(k-1)}) \\
&\qquad - \tfrac{1}{2\beta_i |\mathcal{N}_i|} \mathbf{E}_i^T \big(\tfrac{1}{c} (\mathbf{E}_i \boldsymbol{x}_i^{(k-1)} - \tfrac{1}{N} \boldsymbol{q}) - \tfrac{1}{c} \boldsymbol{p}_i^{(k)} \\
&\qquad + \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) \big) \bigg] \bigg\|_2^2 + g_i(\boldsymbol{x}_i) \bigg\} \\
&= \mathrm{prox}_{g_i}^{\beta_i} \bigg[\boldsymbol{x}_i^{(k-1)} - \tfrac{1}{\beta_i} \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{x}_i^{(k-1)}) \\
&\qquad - \tfrac{1}{2\beta_i |\mathcal{N}_i|} \mathbf{E}_i^T \big(\tfrac{1}{c} (\mathbf{E}_i \boldsymbol{x}_i^{(k-1)} - \tfrac{1}{N} \boldsymbol{q}) - \tfrac{1}{c} \boldsymbol{p}_i^{(k)} \\
&\qquad + \textstyle\sum_{j \in \mathcal{N}_i} (\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) \big) \bigg].
\end{aligned} \tag{44}
$$

We summarize the proposed IDC-ADMM in Algorithm 4.

The convergence property of IDC-ADMM is stated below.

Theorem 3 *Suppose that Assumptions 1, 2(b) and 3 hold and*

$$
\beta_i > \lambda_{\max} \bigg(\tfrac{L_{f,i}^2}{\sigma_{f,i}^2} \boldsymbol{A}_i^T \boldsymbol{A}_i + \tfrac{1}{2|\mathcal{N}_i| c} \mathbf{E}_i^T \mathbf{E}_i \bigg) \ \forall i \in V. \tag{47}
$$

Let $\boldsymbol{x}^\star = [(\boldsymbol{x}_1^\star)^T, \ldots, (\boldsymbol{x}_N^\star)^T]^T$ *denote an optimal solution to* (P2)*, and let* $\boldsymbol{\nu}^\star \triangleq \boldsymbol{\nu}_1^\star = \cdots = \boldsymbol{\nu}_N^\star$ *and* $\{\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star\}$ *denote a pair of optimal primal and dual solutions to problem* (26) *(i.e.,* (5)*).*

(a) The sequence $\boldsymbol{x}^{(k)} = [(\boldsymbol{x}_1^{(k)})^T, \ldots, (\boldsymbol{x}_N^{(k)})^T]^T$ *generated from Algorithm 4 converges to* $\boldsymbol{x}^\star$ *of* (P2) *while* $\boldsymbol{\nu}_1^{(k)}, \ldots, \boldsymbol{\nu}_N^{(k)}$ *converge to a common point* $\boldsymbol{\nu}^\star$ *of problem* (5)*.*

(b) If $\phi_i(\boldsymbol{x}) = f_i(\boldsymbol{A}_i \boldsymbol{x})$*, where* $\boldsymbol{A}_i$ *has full column rank, and* $\mathbf{E}_i$ *has full row rank, for all* $i \in V$*, then for some* $0 < \alpha < 1$ *and* $\rho > 0$*, we have*

$$
\begin{aligned}
& \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{\alpha \mathbf{M} + \frac{1}{2} \boldsymbol{P}}^2 + \frac{1}{c} \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \\
& + \frac{c}{2} \|\boldsymbol{\nu}^{(k)} - \mathbf{1}_N \otimes \boldsymbol{\nu}^\star\|_{(\boldsymbol{D} + \boldsymbol{W}) \otimes \boldsymbol{I}_M}^2 \to 0 \text{ linearly},
\end{aligned} \tag{48}
$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Algorithm 4 Proposed IDC-ADMM for solving (P2)

1: **Given** initial variables $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^K$ and $\boldsymbol{p}_i^{(0)} = \mathbf{0}$ for each agent i, $i \in V$. Set $k = 1$.

2: **repeat**

3: For all $i \in V$ (in parallel),

$$\boldsymbol{p}_i^{(k)} = \boldsymbol{p}_i^{(k-1)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}),$$

$$\begin{aligned}\boldsymbol{x}_i^{(k)} = \mathrm{prox}_{g_i}^{\beta_i}\bigg[&\boldsymbol{x}_i^{(k-1)} - \tfrac{1}{\beta_i}\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) \\ &- \tfrac{1}{2\beta_i|\mathcal{N}_i|}\mathbf{E}_i^T\big(\tfrac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i^{(k-1)} - \tfrac{1}{N}\boldsymbol{q}) - \tfrac{1}{c}\boldsymbol{p}_i^{(k)} \\ &+ \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)})\big)\bigg],\end{aligned} \tag{45}$$

$$\begin{aligned}\boldsymbol{\nu}_i^{(k)} = \tfrac{1}{2|\mathcal{N}_i|}\big(&\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)}) - \tfrac{1}{c}\boldsymbol{p}_i^{(k)} \\ &+ \tfrac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i^{(k)} - \tfrac{1}{N}\boldsymbol{q})\big).\end{aligned} \tag{46}$$

4: **Set** $k = k + 1$.

5: **until** a predefined stopping criterion (e.g., a maximum iteration number) is satisfied.

where $\boldsymbol{u}^{(k)}$ *and* $\boldsymbol{u}^\star$ *are defined similarly as in Theorem 1,* $\mathbf{M}$ *is defined in* (25), *and* $\boldsymbol{P} \triangleq \boldsymbol{D}_\beta - \frac{1}{2c}\mathrm{blkdiag}\{\frac{1}{|\mathcal{N}_1|}\mathbf{E}_1^T\mathbf{E}_1, \ldots, \frac{1}{|\mathcal{N}_N|}\mathbf{E}_N^T\mathbf{E}_N\} \succ \mathbf{0}$.

The proof is presented in Appendix B. Note that, in addition to the smooth and strongly convex objective function, IDC-ADMM also requires matrices $\mathbf{E}_i$'s to have full row rank in order to have a linear convergence rate.

V. Numerical Results

In this section, we examine the numerical performance of Algorithm 1 to 4 presented so far.

A. Performance of C-ADMM and IC-ADMM

To test C-ADMM (Algorithm 1) and IC-ADMM (Algorithm 2), we considered the distributed RPD LR problem in (7) with $\Psi_i(\boldsymbol{y};\boldsymbol{A}_i,\boldsymbol{b}_i)$ in (8) and $g_i(\boldsymbol{y}) = \frac{\lambda}{N}\|\boldsymbol{y}\|_1 + \eta(\boldsymbol{y})$, where $\lambda > 0$ is a penalty parameter, and $\eta(\boldsymbol{y})$ is an indicator function specifying that the regression variables lie in a set $\mathcal{X} = \{\boldsymbol{y} \in \mathbb{R}^K \mid |x_i| \leq a\ \forall\, i\}$ for some $a > 0$ (see Eqn. (3)). We considered a simple two image classification task. Specifically, we used the images D24 and D68 from the Brodatz data set (http://www.ux.uis.no/~tranden/brodatz.html) to generate the regression data matrix $\boldsymbol{A}$. We randomly extracted $(NM)/2$ overlapping patches with dimension $\sqrt{K} \times \sqrt{K}$ from the two images, respectively, followed by vectorizing the M patches into

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

vectors and stacking all of them into an $M \times K$ matrix. The rows of the matrix were randomly shuffled and the resultant matrix was used as the data matrix $\boldsymbol{A}$. For the RPD LR problem (7), we horizontally partitioned the matrix $\boldsymbol{A}$ into N submatrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N$, each with dimension $M \times K$. These matrices were used as the training data. Note that each $\boldsymbol{A}_i$ contains patches from both images. The binary labels $\boldsymbol{b}_i$'s then were generated accordingly with 1 for one image and -1 for the other. The connected graph $\mathcal{G}$ was randomly generated following the same method as in [39].

To implement C-ADMM (Algorithm 1), we employed the fast iterative shrinkage thresholding algorithm (FISTA) [40], [41] to solve subproblem (15) for each agent i. For (15), the associated FISTA steps can be shown as

$$\tilde{\boldsymbol{y}}_i^{(\ell)} = \max\bigg\{-a, \min\bigg\{a, \mathcal{S}\bigg[\boldsymbol{z}_i^{(\ell-1)} - \rho_i^{(\ell)}\bigg[\boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{z}_i^{(\ell-1)}) + \boldsymbol{p}_i^{(k)} + 2c\sum_{j\in\mathcal{N}_i}(\boldsymbol{z}_i^{(\ell-1)} - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2})\bigg], \frac{\lambda \rho_i^{(\ell)}}{N}\bigg]\bigg\}\bigg\}, \tag{49a}$$

$$\boldsymbol{z}_i^{(\ell)} = \tilde{\boldsymbol{y}}_i^{(\ell)} + \frac{\ell-1}{\ell+2}(\tilde{\boldsymbol{y}}_i^{(\ell)} - \tilde{\boldsymbol{y}}_i^{(\ell-1)}), \tag{49b}$$

where ℓ denotes the inner iteration index of FISTA, $\rho_i^{(\ell)} > 0$ is a step size and $\mathcal{S}$ is defined in (21). The stopping criterion of (49) was based on the PG residue (pgr) $\mathsf{pgr} = \|\boldsymbol{z}_i^{(\ell-1)} - \tilde{\boldsymbol{y}}_i^{(\ell)}\|/(\rho_i^{(\ell)}\sqrt{K})$ [40], [41]. For obtaining a high-accuracy solution of (15), one may set the stopping criterion as, e.g., $\mathsf{pgr} < 10^{-5}$. Suppose that FISTA stops at iteration $\ell_i(k)$. We then set $\boldsymbol{y}_i^{(k)} = \tilde{\boldsymbol{y}}_i^{(\ell_i(k))}$ as a solution to subproblem (15).

For IC-ADMM (Algorithm 2), the corresponding step in (20) is given by

$$\boldsymbol{y}_i^{(k)} = \max\bigg\{-a, \min\bigg\{a, \frac{1}{\gamma_i}\mathcal{S}\bigg[\beta \boldsymbol{y}_i^{(k-1)} - \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) - \boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}), \frac{\lambda}{N}\bigg]\bigg\}\bigg\}. \tag{50}$$

From (**??**) and (49), the complexity of agent i at iteration k of C-ADMM is given by the order of $K + \ell_i(k)(2MK + 2K)$ if one counts only the multiplication operations; while from (**??**) and (50), the per-iteration complexity of each agent in IC-ADMM is given by the order of $K + (2MK + 2K)$. One can see that, for each agent i, the computational complexity of C-ADMM per iteration k (we refer this as the "ADMM iteration (ADMM Ite.)") is roughly $\ell_i(k)$ times that of IC-ADMM.

The stopping criterion of Algorithms 1 and 2 was based on measuring the solution accuracy $\mathsf{acc} = (\mathsf{obj}(\hat{\boldsymbol{y}}^{(k)}) - \mathsf{obj}^\star)/\mathsf{obj}^\star$ and variable consensus error $\mathsf{cserr} = \sum_{i=1}^N \|\hat{\boldsymbol{y}}^{(k)} - \boldsymbol{y}_i^{(k)}\|_2^2/N$, where $\hat{\boldsymbol{y}}^{(k)} = (\sum_{i=1}^N \boldsymbol{y}_i^{(k)})/N$, $\mathsf{obj}(\hat{\boldsymbol{y}}^{(k)})$ denotes the objective value of (7) given $\boldsymbol{y} = \hat{\boldsymbol{y}}^{(k)}$, and $\mathsf{obj}^\star$ is the optimal

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

value of (7) which was obtained by FISTA [40], [41] with a high solution accuracy of $\textsf{pgr} < 10^{-6}$. The two algorithms were set to stop whenever $\textsf{acc}$ and $\textsf{cserr}$ are both smaller than preset target values.

In Table I(a), we considered a simulation example of $N = 10$, $K = 10,000$, $M = 10$, $\lambda = 0.1$ and $a = 1$, and display the comparison results. We not only present the required ADMM iterations but also the computation time per agent[6] (in second) of the two methods. The convergence curves of C-ADMM and IC-ADMM with respect to the ADMM iteration are also shown in Figs. 1(a) and 1(b). The stopping conditions are $\textsf{acc} < 10^{-4}$ and $\textsf{cserr} < 10^{-5}$. For C-ADMM, we considered two cases, one with the stopping condition of FISTA for solving subproblem (15) set to $\textsf{pgr} < 10^{-5}$ and the other with that set to $\textsf{pgr} < 10^{-4}$. The penalty parameter c for C-ADMM was set to $c = 0.03$ and the step size $\rho_i^{(\ell)}$ of FISTA (see (49)) was set to a constant $\rho_i^{(\ell)} = 0.1$. The penalty parameters c and β of IC-ADMM were set to $c = 0.01$ and $\beta = 1.2$. We observe from Table I(a) that IC-ADMM in general requires more ADMM iterations than C-ADMM; however, the computation time is significantly smaller, as also illustrated in Figure 1(c). Specifically, the computation time of IC-ADMM is around $44.56/2.14 \approx 20.8$ times smaller than that of C-ADMM ($\textsf{pgr} < 10^{-5}$). We also observe that C-ADMM ($\textsf{pgr} < 10^{-4}$) consumes a smaller computation time for achieving $\textsf{acc} < 10^{-4}$. However, the associated $\textsf{cserr} = 3.425 \times 10^{-4}$ does not achieve the target value 10^{-5}. In fact, C-ADMM ($\textsf{pgr} < 10^{-4}$) cannot reduce $\textsf{cserr}$ properly. As one can see from Fig. 1(b), the $\textsf{cserr}$ curve of C-ADMM ($\textsf{pgr} < 10^{-4}$) keeps relatively high and does not decrease along the iterations. In Fig. 1(a) and Fig. 1(b), we also plot the convergence curves of the consensus subgradient method in [5], where the diminishing step size $10/k$ was used. As one can see, the consensus subgradient method converges much slower than IC-ADMM.

In Table I(b), we considered another example with the network size increased to $N = 50$. We set $c = 0.004$ for C-ADMM and $\rho_i^{(\ell)} = 0.1$ for FISTA; while for IC-ADMM, we set $c = 0.008$ and $\beta = 1.2$. The computation times of C-ADMM and IC-ADMM under this setting are also shown in Fig. 1(c). We can observe similar comparison results from Table I(b) and Fig. 1(c). Specifically, the computation time of IC-ADMM is around 8.75 times smaller than C-ADMM ($\textsf{pgr} < 10^{-5}$). When considering a lower accuracy of $\textsf{pgr} < 10^{-4}$, it is found that C-ADMM cannot properly converge.

To corroborating the linear convergence behavior of C-ADMM and IC-ADMM as claimed in Theorem 1(b)), we consider a problem instance of (7) with $\lambda = 0$, $N = 10$, $K = 25$, $M = 1,000$ and $a = 10$. We set $c = 0.2$ for C-ADMM and $\rho_i^{(\ell)} = 0.01$ and $\textsf{pgr} < 10^{-5}$ for FISTA; while for IC-ADMM, we set

[6]The simulation was performed on a desktop computer with 8-core Intel 1.3GHz CPU and 8 GB RAM. All the algorithms were implemented by MATLAB codes.

TABLE I: Comparison of C-ADMM and IC-ADMM

(a) $N = 10,\ K = 10,000,\ M = 10,\ \lambda = 0.1,\ a = 1.$

	C-ADMM (pgr $< 10^{-5}$)	**C-ADMM** (pgr $< 10^{-4}$)	**IC-ADMM**
ADMM Ite.	810	675	2973
Compt. Time (sec)	**44.56**	**17.86**	**2.14**
acc$< 10^{-4}$	9.982×10^{-5}	9.91×10^{-5}	9.99×10^{-5}
cserr$< 10^{-5}$	1.53×10^{-6}	$\mathbf{3.425 \times 10^{-4}}$	$\mathbf{3.859 \times 10^{-9}}$

(b) $N = 50,\ K = 10,000,\ M = 10,\ \lambda = 0.15,\ a = 1.$

	C-ADMM (pgr $< 10^{-5}$)	**C-ADMM** (pgr $< 10^{-4}$)	**IC-ADMM**
ADMM Ite.	952	N/A	7,251
Compt. Time (sec)	**81.72**	N/A	**9.33**
acc$< 10^{-4}$	9.99×10^{-5}	N/A	9.999×10^{-5}
cserr$< 10^{-5}$	1.305×10^{-7}	N/A	$\mathbf{1.169 \times 10^{-10}}$

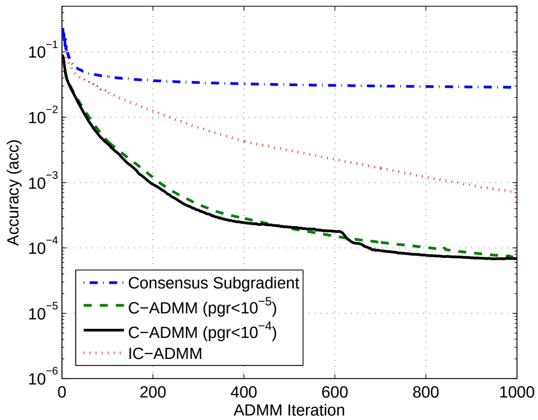
$c = 1.2$ and $\beta = 10$. The convergence curves are shown in Figure 2. One can see from this figure that both algorithms converge linearly under this setting.

B. Performance of DC-ADMM and IDC-ADMM

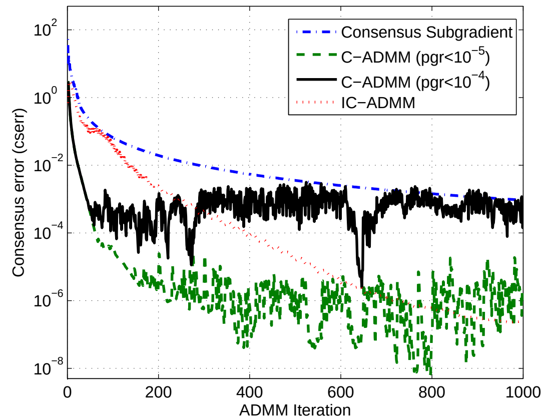
We examine the performance of DC-ADMM (Algorithm 3) and IDC-ADMM (Algorithm 4) by considering the distributed CPD LR problem in (9), with $\Psi_i(\boldsymbol{x}_i; \mathbf{E}_i, \boldsymbol{b})$ in (10) and $g_i(\boldsymbol{x}_i) = \lambda \|\boldsymbol{x}_i\|_1$. Each variable $\boldsymbol{x}_i$ is subject to the constraint set $\mathcal{X}_i = \{\boldsymbol{x}_i \in \mathbb{R}^{K/N} \mid |[\boldsymbol{x}_i]_j| \leq a\ \forall j\}$ for some $a > 0$. DC-ADMM and IDC-ADMM were applied to handle the associated problem (11). The regression data matrix $\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_N]$ was generated following the same way as generating $\boldsymbol{A}$ in Section V-A. To implement DC-ADMM, we employed FISTA [40], [41] to solve subproblem (35) and the solution accuracy was measured by the PG residue of FISTA.

In Table II(a), we show the comparison results for an example of $N = 50$, $K = 200$, $M = 100$, $\lambda = 0.05$ and $a = 10$. The convergence curves are also shown in Figs. 3(a) to 3(c). It was set $c = 0.05$ for DC-ADMM and the step size of FISTA $\rho_i^{(\ell)}$ was determined based on a line search rule [41]. We see from Table II(a) that, for achieving acc $< 10^{-4}$, DC-ADMM (pgr $< 10^{-5}$) took 329 ADMM iterations whereas IDC-ADMM took 10,814 iterations. However, the computation time of DC-ADMM (pgr $< 10^{-5}$)

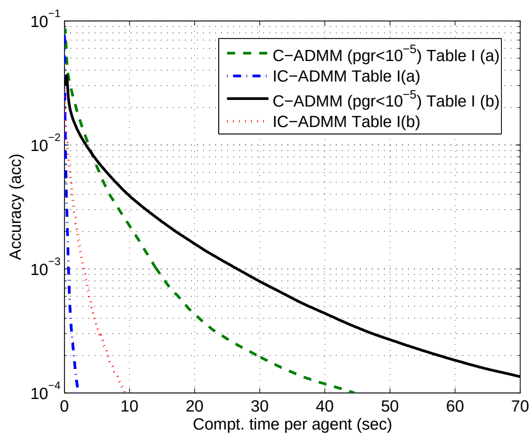
This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.



(a)



(b)



(c)

Fig. 1: Convergence curves of C-ADMM and IC-ADMM.

is around $42.78/1.92 \approx 22.28$ times higher than IDC-ADMM. When one reduce the solution accuracy of FISTA for solving subproblem (35) to $\textsf{pgr} < 10^{-4}$, DC-ADMM cannot reach the high accuracy of $\textsf{acc} < 10^{-4}$, as observed in Fig. 3(a). From Fig. 3(b), one can see that DC-ADMM converges much faster than IDC-ADMM with respect to the ADMM iterations. However, as shown from Fig. 3(c), the comparison result is reversed when one counts the computation times.

In Table II(b), we considered another example with K increased to 800. We set $c = 0.05$ for DC-ADMM, and set $c = 0.08$ and $\beta = 5$ for IDC-ADMM. From Table II(b) and Figs. 3(b) and 3(c), one can observe similar results.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

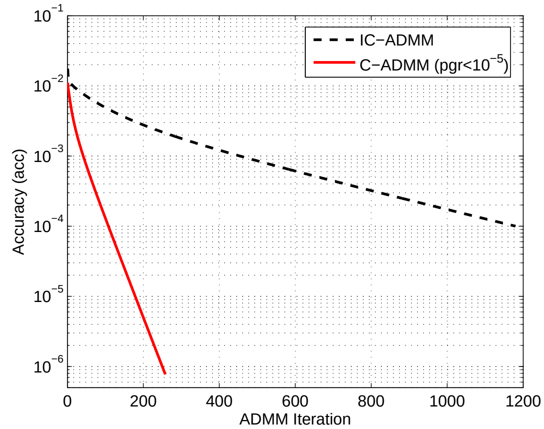


Fig. 2: Convergence curves of C-ADMM and IC-ADMM.

TABLE II: Comparison of DC-ADMM and IDC-ADMM

(a) $N = 50$, $K = 200$, $M = 100$, $\lambda = 0.05$, $a = 10$.

	DC-ADMM (pgr $< 10^{-5}$)	**DC-ADMM** (pgr $< 10^{-4}$)	**IDC-ADMM**
ADMM Ite.	329	N/A	10814
Compt. Time (sec)	**42.78**	N/A	**1.92**
acc$< 10^{-4}$	9.928×10^{-5}	N/A	9.997×10^{-5}

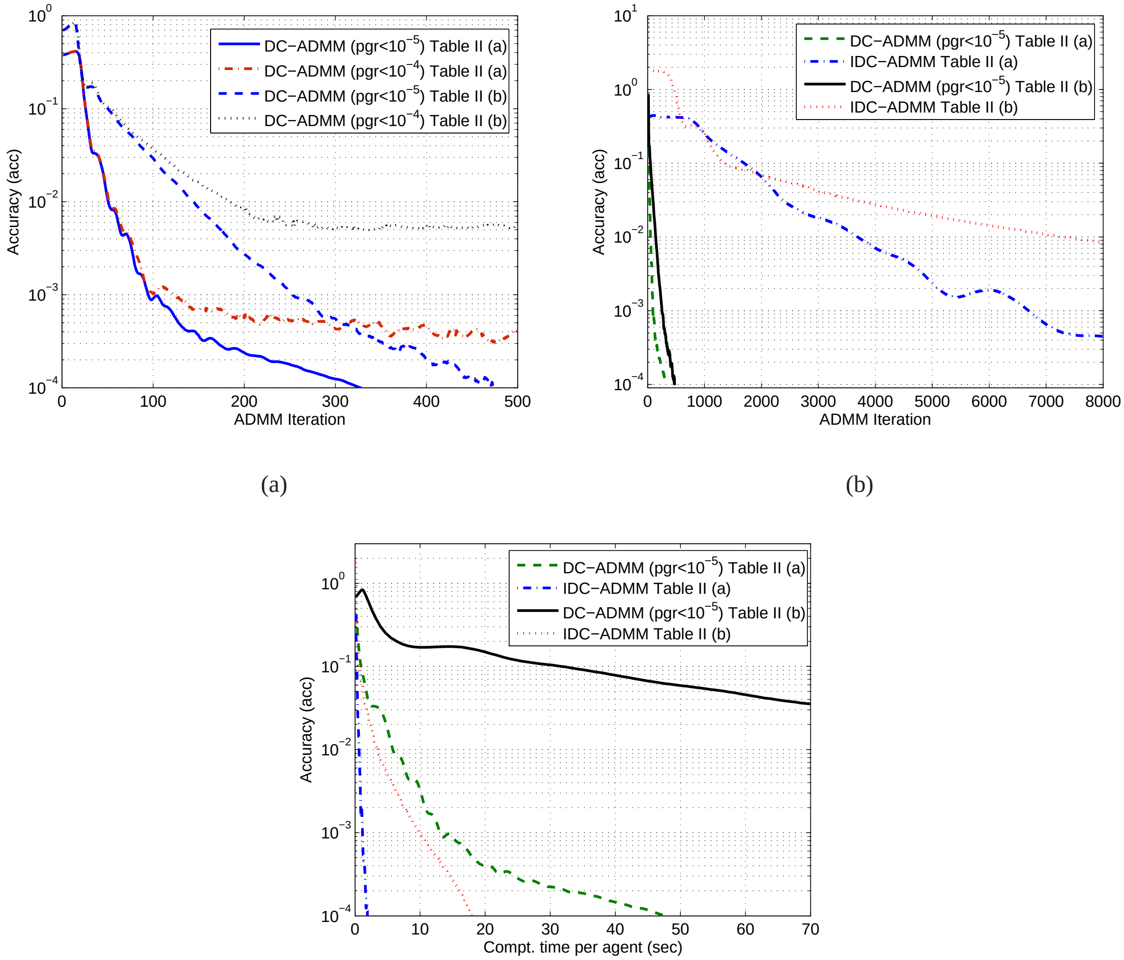
(b) $N = 50$, $K = 800$, $M = 100$, $\lambda = 0.01$, $a = 20$.

	DC-ADMM (pgr $< 10^{-5}$)	**DC-ADMM** (pgr $< 10^{-4}$)	**IDC-ADMM**
ADMM Ite.	475	N/A	38728
Compt. Time (sec)	**427.73**	N/A	**18.07**
acc$< 10^{-4}$	9.777×10^{-5}	N/A	9.999×10^{-5}

VI. Conclusions

In this paper, we have presented ADMM based distributed optimization methods for solving problems (P1) and (P2) in multi-agent networks. In particular, aiming at reducing the computational complexity of C-ADMM for solving large-scale instances of (P1) with complicated objective functions, we have proposed the IC-ADMM method (Algorithm 2) where agents perform one PG update only at each iteration. For (P2), we have proposed the DC-ADMM method (Algorithm 3) and its complexity reduced

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.



(a) (b)

(c)

Fig. 3: Convergence curves of DC-ADMM and IDC-ADMM.

counterpart IDC-ADMM (Algorithm 4). Preliminary numerical results based on the distributed LR problems (7) and (11) have shown that the proposed methods converge faster than the consensus subgradient method. Moreover, both IC-ADMM and IDC-ADMM require more ADMM iterations than C-ADMM and DC-ADMM, but the traded computational complexity reduction is significant.

Appendix A
Proof of Theorem 1

Proof of Theorem 1(a): Let $\tilde{\boldsymbol{y}}^\star \triangleq [(\boldsymbol{y}_1^\star)^T, \ldots, (\boldsymbol{y}_N^\star)^T]^T$ and $\{\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star, j \in \mathcal{N}_i\}_{i=1}^N$ be a pair of optimal primal and dual solutions to problem (13). Then they satisfy the following Karush-Kuhn-Tucker (KKT)

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

conditions: $\forall i \in V$,

$$\boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^\star) + \partial g_i(\boldsymbol{y}_i^\star) + \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ji}^\star) = \boldsymbol{0}, \tag{A.1}$$

$$\boldsymbol{y}_i^\star = \boldsymbol{y}_j^\star \ \forall j \in \mathcal{N}_i, \tag{A.2}$$

$$\boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ij}^\star = \boldsymbol{0}, \ \forall j \in \mathcal{N}_i, \tag{A.3}$$

where $\partial g_i(\boldsymbol{y}_i^\star)$ denotes the subgradient of g_i at $\boldsymbol{y}_i^\star$. Under Assumption 1, (A.2) implies that $\boldsymbol{y}^\star \triangleq \boldsymbol{y}_1^\star = \cdots = \boldsymbol{y}_N^\star$ and $\tilde{\boldsymbol{y}}^\star = \boldsymbol{1}_N \otimes \boldsymbol{y}^\star$, i.e., consensus among agents is reached, and thus $\boldsymbol{y}^\star$ is optimal to the original problem (P1).

By recalling that $\boldsymbol{p}_i^{(k)} = \sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)}) \ \forall i \in V$, and by the optimality condition of (18) [15], we have that

$$\begin{aligned} \boldsymbol{0} = &\boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) + \beta_i(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_i^{(k-1)}) + \partial g_i(\boldsymbol{y}_i^{(k)}) \\ &+ \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)}) \\ &+ 2c \textstyle\sum_{j\in\mathcal{N}_i}\big(\boldsymbol{y}_i^{(k)} - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2}\big). \end{aligned} \tag{A.4}$$

By combining (A.4) with (A.1), one obtains

$$\begin{aligned} &\boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) - \boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}^\star) + \beta_i(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_i^{(k-1)}) \\ &+ \partial g_i(\boldsymbol{y}_i^{(k)}) - \partial g_i(\boldsymbol{y}^\star) + \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)} - \boldsymbol{u}_{ij}^\star - \boldsymbol{v}_{ji}^\star) \\ &+ 2c \textstyle\sum_{j\in\mathcal{N}_i}\big(\boldsymbol{y}_i^{(k)} - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2}\big) = \boldsymbol{0}. \end{aligned} \tag{A.5}$$

Adding and subtracting $\boldsymbol{A}_i^T \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k)})$ in the left hand side (LHS) of (A.5) followed by multiplying $(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star)$ on both sides yields

$$\begin{aligned} &(\nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k-1)}) - \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k)}))^T \boldsymbol{A}_i(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) + \beta_i(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_i^{(k-1)})^T(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &+ (\nabla f_i(\boldsymbol{A}_i \boldsymbol{y}_i^{(k)}) - \nabla f_i(\boldsymbol{A}_i \boldsymbol{y}^\star))^T \boldsymbol{A}_i(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &+ (\partial g_i(\boldsymbol{y}_i^{(k)}) - \partial g_i(\boldsymbol{y}^\star))^T(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) + \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} - \boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ji}^{(k)} - \boldsymbol{v}_{ji}^\star)^T(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &+ 2c \textstyle\sum_{j\in\mathcal{N}_i}\left(\boldsymbol{y}_i^{(k)} - \frac{\boldsymbol{y}_i^{(k-1)} + \boldsymbol{y}_j^{(k-1)}}{2}\right)^T(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) = \boldsymbol{0}. \end{aligned} \tag{A.6}$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Note that the first term on the LHS of (A.6) can be lower bounded as

$$\begin{aligned}&(\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k-1)})-\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k)}))^T\boldsymbol{A}_i(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &\geq \tfrac{-1}{2\rho_i}\|\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k-1)})-\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k)})\|_2^2\\ &\qquad\qquad -\tfrac{\rho_i}{2}\|\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star\|^2_{\boldsymbol{A}_i^T\boldsymbol{A}_i}\\ &\geq \tfrac{-L_{f,i}^2}{2\rho_i}\|\boldsymbol{y}_i^{(k-1)}-\boldsymbol{y}_i^{(k)}\|^2_{\boldsymbol{A}_i^T\boldsymbol{A}_i}-\tfrac{\rho_i}{2}\|\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star\|^2_{\boldsymbol{A}_i^T\boldsymbol{A}_i}\end{aligned}\tag{A.7}$$

for any $\rho_i>0$, where the second inequality is due to (12) in Assumption 3. By the strong convexity of f_i and convexity of g_i, the third and fourth terms of (A.6) can respectively be lower bounded as

$$\begin{aligned}&(\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k)})-\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}^\star))^T\boldsymbol{A}_i(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &\qquad\qquad\geq \sigma_{f,i}^2\|\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star\|^2_{\boldsymbol{A}_i^T\boldsymbol{A}_i},\end{aligned}\tag{A.8}$$

$$(\partial g_i(\boldsymbol{y}_i^{(k)})-\partial g_i(\boldsymbol{y}^\star))^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\geq 0.\tag{A.9}$$

Moreover, it follows from (14a) and (14b) that the fifth term of (A.6) can be expressed as

$$\begin{aligned}&\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)}-\boldsymbol{u}_{ij}^\star+\boldsymbol{v}_{ji}^{(k)}-\boldsymbol{v}_{ji}^\star)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &=\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)}-\boldsymbol{u}_{ij}^\star+\boldsymbol{v}_{ji}^{(k+1)}-\boldsymbol{v}_{ji}^\star)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &\qquad -2c\textstyle\sum_{j\in\mathcal{N}_i}\big(\boldsymbol{y}_i^{(k)}-\frac{\boldsymbol{y}_i^{(k)}+\boldsymbol{y}_j^{(k)}}{2}\big)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star).\end{aligned}\tag{A.10}$$

By substituting (A.7) to (A.10) into (A.6) and summing over $i=1,\ldots,N$, we obtain

$$\begin{aligned}&\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{\mathbf{M}}-\frac{1}{2}\|\boldsymbol{y}^{(k-1)}-\boldsymbol{y}^{(k)}\|^2_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}}\\ &\qquad\qquad +(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T\boldsymbol{D}_\beta(\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star)\\ &+\sum_{i=1}^N\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)}-\boldsymbol{u}_{ij}^\star)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &+\sum_{i=1}^N\sum_{j\in\mathcal{N}_i}(\boldsymbol{v}_{ji}^{(k+1)}-\boldsymbol{v}_{ji}^\star)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &+2c\sum_{i=1}^N\sum_{j\in\mathcal{N}_i}\bigg(\frac{\boldsymbol{y}_i^{(k)}+\boldsymbol{y}_j^{(k)}}{2}-\frac{\boldsymbol{y}_i^{(k-1)}+\boldsymbol{y}_j^{(k-1)}}{2}\bigg)^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\ &\leq 0,\end{aligned}\tag{A.11}$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

where $\boldsymbol{y}^{(k)} = [(\boldsymbol{y}_1^{(k)})^T, \ldots, (\boldsymbol{y}_N^{(k)})^T]^T$, $\tilde{\boldsymbol{A}} = \text{blkdiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\}$, $\boldsymbol{D}_{L_f} = \text{diag}\{L_{f,1}^2, \ldots, L_{f,N}^2\} \otimes \boldsymbol{I}_K$, $\boldsymbol{D}_\beta = \text{diag}\{\beta_1, \ldots, \beta_N\} \otimes \boldsymbol{I}_K$, $\boldsymbol{D}_\rho = \text{diag}\{\rho_1, \ldots, \rho_N\} \otimes \boldsymbol{I}_K$, and as defined in (25),

$$\mathbf{M} = \tilde{\boldsymbol{A}}^T(\boldsymbol{D}_{\sigma_f} - \frac{1}{2}\boldsymbol{D}_\rho)\tilde{\boldsymbol{A}}.$$

It can be observed from (A.3) and also (14a) and (14b) that

$$\boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ij}^\star = \boldsymbol{0} \ \forall j, i, \tag{A.12}$$

$$\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ij}^{(k)} = \boldsymbol{0} \ \forall j, i, k, \tag{A.13}$$

given the initial $\boldsymbol{u}_{ij}^{(0)} + \boldsymbol{v}_{ij}^{(0)} = \boldsymbol{0} \ \forall j, i, k$ which is equivalent to setting $\boldsymbol{p}_i^{(k)} = \boldsymbol{0} \ \forall i \in V$ (See Step 1 of Algorithm 2). Besides, due to the symmetric property of $\boldsymbol{W}$, for any $\{\alpha_{ij}\}$, we have

$$\begin{aligned}\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \alpha_{ij} &= \sum_{i=1}^N \sum_{j=1}^N [\boldsymbol{W}]_{i,j} \alpha_{ij} \\ &= \sum_{i=1}^N \sum_{j=1}^N [\boldsymbol{W}]_{i,j} \alpha_{ji} = \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \alpha_{ji}.\end{aligned} \tag{A.14}$$

By the above two properties, the fourth and fifth terms in the LHS of (A.11) can be written as

$$\begin{aligned}&\textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{u}_{ij}^{(k+1)} - \boldsymbol{u}_{ij}^\star)^T (\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &\qquad + \textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{v}_{ji}^{(k+1)} - \boldsymbol{v}_{ji}^\star)^T (\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &= \textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{u}_{ij}^{(k+1)} - \boldsymbol{u}_{ij}^\star)^T (\boldsymbol{y}_i^{(k)} - \boldsymbol{y}^\star) \\ &\qquad + \textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{v}_{ij}^{(k+1)} - \boldsymbol{v}_{ij}^\star)^T (\boldsymbol{y}_j^{(k)} - \boldsymbol{y}^\star) \\ &= \textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{u}_{ij}^{(k+1)} - \boldsymbol{u}_{ij}^\star)^T (\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_j^{(k)}) \\ &= \tfrac{2}{c} \textstyle\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\boldsymbol{u}_{ij}^{(k+1)} - \boldsymbol{u}_{ij}^\star)^T (\boldsymbol{u}_{ij}^{(k+1)} - \boldsymbol{u}_{ij}^{(k)}) \\ &\triangleq \tfrac{2}{c} (\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)^T (\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}),\end{aligned} \tag{A.15}$$

where the first equality is owing to (A.14), the second equality is by (A.12) and (A.13), and the third equality is due to (14a). In (A.15), $\boldsymbol{u}^{(k)}$ ($\boldsymbol{u}^\star$) is a vector that stacks $\boldsymbol{u}_{ij}^{(k)}$ ($\boldsymbol{u}_{ij}^\star$) for all $j \in \mathcal{N}_i$, $i = 1, \ldots, N$.

The sixth term in the LHS of (A.11) can be rearranged as follows

$$\begin{aligned}
&c\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}_i^{(k-1)})^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\
&\qquad+c\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}(\boldsymbol{y}_j^{(k)}-\boldsymbol{y}_j^{(k-1)})^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\
&=c\sum_{i=1}^{N}|\mathcal{N}_i|(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}_i^{(k-1)})^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\
&\qquad+c\sum_{i=1}^{N}\sum_{j=1}^{N}[\boldsymbol{W}]_{i,j}(\boldsymbol{y}_j^{(k)}-\boldsymbol{y}_j^{(k-1)})^T(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}^\star)\\
&=c(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T(\boldsymbol{D}\otimes\boldsymbol{I}_K)(\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star)\\
&\qquad+c(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T(\boldsymbol{W}\otimes\boldsymbol{I}_K)(\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star)\\
&=c(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T[(\boldsymbol{D}+\boldsymbol{W})\otimes\boldsymbol{I}_K](\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star).
\end{aligned}\tag{A.16}$$

Note that, by the graph theory [32], the normalized Laplacian matrix, i.e., $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$, have $\lambda_{\max}(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}})\leq 2$. Thus, in (A.16),

$$\boldsymbol{D}+\boldsymbol{W}=2\boldsymbol{D}-\boldsymbol{L}=\boldsymbol{D}^{\frac{1}{2}}(2\boldsymbol{I}_N-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{D}^{\frac{1}{2}}\succeq\boldsymbol{0}.$$

By substituting (A.15) and (A.16) into (A.11), we obtain

$$\begin{aligned}
&\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|_{\mathbf{M}}^2-\frac{1}{2}\|\boldsymbol{y}^{(k-1)}-\boldsymbol{y}^{(k)}\|^2_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}}\\
&\qquad+(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T\boldsymbol{G}(\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star)\\
&+\frac{2}{c}(\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star)^T(\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^{(k)})\leq 0,
\end{aligned}\tag{A.17}$$

where as defined in (24),

$$\boldsymbol{G}\triangleq\boldsymbol{D}_\beta+c((\boldsymbol{D}+\boldsymbol{W})\otimes\boldsymbol{I}_K)\succ\boldsymbol{0}.$$

Note that

$$\begin{aligned}
(\boldsymbol{a}^{(k)}-\boldsymbol{a}^{(k-1)})^T\boldsymbol{Q}(\boldsymbol{a}^{(k)}-\boldsymbol{a}^\star)&=\frac{1}{2}\|\boldsymbol{a}^{(k)}-\boldsymbol{a}^\star\|_{\boldsymbol{Q}}^2\\
&+\frac{1}{2}\|\boldsymbol{a}^{(k)}-\boldsymbol{a}^{(k-1)}\|_{\boldsymbol{Q}}^2-\frac{1}{2}\|\boldsymbol{a}^{(k-1)}-\boldsymbol{a}^\star\|_{\boldsymbol{Q}}^2
\end{aligned}\tag{A.18}$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

for any sequence $\boldsymbol{a}^{(k)}$ and matrix $\boldsymbol{Q} \succeq \boldsymbol{0}$. By applying (A.18) to each of the terms in (A.17), one obtains that

$$\begin{aligned}
&(\boldsymbol{y}^{(k)} - \tilde{\boldsymbol{y}}^\star)^T \left[\mathbf{M} + \frac{1}{2}\boldsymbol{G}\right](\boldsymbol{y}^{(k)} - \tilde{\boldsymbol{y}}^\star) + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \\
&\leq \frac{1}{2}(\boldsymbol{y}^{(k-1)} - \tilde{\boldsymbol{y}}^\star)^T\boldsymbol{G}(\boldsymbol{y}^{(k-1)} - \tilde{\boldsymbol{y}}^\star) \\
&\qquad + \frac{1}{c}\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^\star\|_2^2 - \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2^2 \\
&- (\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)})^T\left[\frac{1}{2}\boldsymbol{G} - \frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}\right](\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}).
\end{aligned} \tag{A.19}$$

Now, consider the condition on β_i in (23). It can be easily checked that (23) implies that

$$\sigma_{f,i}^2 - \frac{\rho_i}{2} > 0, \tag{A.20a}$$

$$\beta_i\boldsymbol{I}_K + c\lambda_{\min}(\boldsymbol{D} + \boldsymbol{W})\boldsymbol{I}_K - \frac{L_{f,i}^2}{\rho_i}\boldsymbol{A}_i^T\boldsymbol{A}_i \succ \boldsymbol{0}, \tag{A.20b}$$

for some $\sigma_{f,i}^2 \leq \rho_i < 2\sigma_{f,i}^2\ \forall i \in V$, and therefore

$$\mathbf{M} \succeq \boldsymbol{0},\ \boldsymbol{G} - \tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}} \succ \boldsymbol{0}. \tag{A.21}$$

With (A.21), (A.19) implies the following two results (R1) as $k \to \infty$, the sequence $\frac{1}{2}\|\boldsymbol{y}^{(k)} - \tilde{\boldsymbol{y}}^\star\|_{\boldsymbol{G}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2$ converges for any pair of optimal $\tilde{\boldsymbol{y}}^\star$ and $\boldsymbol{u}^\star$ to problem (13); and (R2)

$$\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)} \to \boldsymbol{0},\ \boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)} \to \boldsymbol{0}. \tag{A.22}$$

The result (R1) implies that the sequences of $\{\boldsymbol{y}_i^{(k)}\}$ and $\{\boldsymbol{u}_{ij}^{(k)}\}$ (so is $\{\boldsymbol{v}_{ij}^{(k)}\}$) are bounded. Let $\tilde{\hat{\boldsymbol{y}}} = [(\hat{\boldsymbol{y}}_1)^T, \ldots, (\hat{\boldsymbol{y}}_N)^T]^T$, $\hat{\boldsymbol{u}}_{ij}$ and $\hat{\boldsymbol{v}}_{ij}$ be a set of limit points of $\{\boldsymbol{y}^{(k)}\}$, $\{\boldsymbol{u}_{ij}^{(k)}\}$ and $\{\boldsymbol{v}_{ij}^{(k)}\}$, respectively. Firstly, by the result of $\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)} \to \boldsymbol{0}$ and (14a), we have

$$\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_j^{(k)} \to \boldsymbol{0} \Longrightarrow \hat{\boldsymbol{y}} \triangleq \hat{\boldsymbol{y}}_i = \hat{\boldsymbol{y}}_j,\ \forall j, i. \tag{A.23}$$

Secondly, by (A.13), we have

$$\hat{\boldsymbol{u}}_{ij} + \hat{\boldsymbol{v}}_{ij} = \boldsymbol{0}\ \forall j, i. \tag{A.24}$$

Thirdly, by applying the result of $\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)} \to \boldsymbol{0}$ and (A.23) to (A.4), we have

$$\boldsymbol{0} = \boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\hat{\boldsymbol{y}}_i) + \partial g_i(\hat{\boldsymbol{y}}_i) + \sum_{j\in\mathcal{N}_i}(\hat{\boldsymbol{u}}_{ij} + \hat{\boldsymbol{v}}_{ji}) \tag{A.25}$$

for all $i \in V$. So, $\tilde{\hat{\boldsymbol{y}}}$ and $\{\hat{\boldsymbol{u}}_{ij}, \hat{\boldsymbol{v}}_{ij}\}$ are in fact a pair of optimal primal and dual solutions to problem (13) [see (A.1), (A.2) and (A.3)]. Therefore, according to (R1), the sequence $\frac{1}{2}\|\boldsymbol{y}^{(k)} - \tilde{\hat{\boldsymbol{y}}}\|_{\boldsymbol{G}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \hat{\boldsymbol{u}}\|_2^2$ converges. Furthermore, since $\frac{1}{2}\|\boldsymbol{y}^{(k)} - \tilde{\hat{\boldsymbol{y}}}\|_{\boldsymbol{G}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \hat{\boldsymbol{u}}\|_2^2$ has a limit value equal to zero, we conclude

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

that $\frac{1}{2}\|\boldsymbol{y}^{(k)}-\tilde{\hat{\boldsymbol{y}}}\|_{\boldsymbol{G}}^2+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\hat{\boldsymbol{u}}\|_2^2$ in fact converges to zero. This says that $\boldsymbol{y}_i^{(k)} \to \hat{\boldsymbol{y}}\ \forall\ i \in V$ and $\boldsymbol{u}^{(k+1)} \to \hat{\boldsymbol{u}}$. The proof is thus complete. ■

Proof of Theorem 1(b): Let $0<\alpha<1$ be some positive number and rewrite (A.19) as

$$\begin{aligned}&\left(\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star\|_2^2\right)+\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{(1-\alpha)\mathbf{M}}\\&+\|\boldsymbol{y}^{(k-1)}-\tilde{\boldsymbol{y}}^\star\|^2_{\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^{(k)}\|_2^2\\&+(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T\left[\frac{1}{2}\boldsymbol{G}-\frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}\right](\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})\\&\leq\left(\|\boldsymbol{y}^{(k-1)}-\tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k)}-\boldsymbol{u}^\star\|_2^2\right).\end{aligned}\tag{A.26}$$

Then, in order to prove linear convergence rate, i.e., for some $\delta>0$,

$$\begin{aligned}&\big(\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star\|_2^2\big)\\&\leq\frac{1}{1+\delta}\big(\|\boldsymbol{y}^{(k-1)}-\tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k)}-\boldsymbol{u}^\star\|_2^2\big),\end{aligned}$$

it is sufficient to show that

$$\begin{aligned}&\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{(1-\alpha)\mathbf{M}}+\|\boldsymbol{y}^{(k-1)}-\tilde{\boldsymbol{y}}^\star\|^2_{\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^{(k)}\|_2^2\\&+(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})^T\left[\frac{1}{2}\boldsymbol{G}-\frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}\right](\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})\\&\geq\delta\left(\|\boldsymbol{y}^{(k)}-\tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star\|_2^2\right).\end{aligned}\tag{A.27}$$

Recall from (A.5) and (A.10) that

$$\begin{aligned}&\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k-1)})-\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}^\star)+\beta_i(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}_i^{(k-1)})\\&+\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)}-\boldsymbol{u}_{ij}^\star)+\sum_{j\in\mathcal{N}_i}(\boldsymbol{v}_{ji}^{(k+1)}-\boldsymbol{v}_{ji}^\star)\\&+2c\textstyle\sum_{j\in\mathcal{N}_i}\left(\frac{\boldsymbol{y}_i^{(k)}+\boldsymbol{y}_j^{(k)}}{2}-\frac{\boldsymbol{y}_i^{(k-1)}+\boldsymbol{y}_j^{(k-1)}}{2}\right)=\boldsymbol{0}.\end{aligned}\tag{A.28}$$

By applying (A.12) and (A.13), (A.28) can be expressed as

$$\begin{aligned}&\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}_i^{(k-1)})-\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{y}^\star)+\beta_i(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}_i^{(k-1)})\\&+\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)}-\boldsymbol{u}_{ji}^{(k+1)}-\boldsymbol{u}_{ij}^\star+\boldsymbol{u}_{ji}^\star)\\&+c\textstyle\sum_{j\in\mathcal{N}_i}\left(\boldsymbol{y}_i^{(k)}-\boldsymbol{y}_i^{(k-1)}+\boldsymbol{y}_j^{(k)}-\boldsymbol{y}_j^{(k-1)}\right)=\boldsymbol{0}.\end{aligned}\tag{A.29}$$

After stacking (A.29) for $i=1,\ldots,N$, one obtains

$$\begin{aligned}&\tilde{\boldsymbol{A}}^T(\nabla\boldsymbol{f}(\tilde{\boldsymbol{A}}\boldsymbol{y}^{(k-1)})-\nabla\boldsymbol{f}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}^\star))+\boldsymbol{G}(\boldsymbol{y}^{(k)}-\boldsymbol{y}^{(k-1)})\\&\qquad+\boldsymbol{\Upsilon}(\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star)=\boldsymbol{0}.\end{aligned}\tag{A.30}$$

where $\nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\boldsymbol{y}^{(k)}) \triangleq [(\nabla f_1(\boldsymbol{A}_1\boldsymbol{y}_1^{(k)}))^T, \ldots, (\nabla f_1(\boldsymbol{A}_N\boldsymbol{y}_N^{(k)}))^T]^T$ and $\boldsymbol{\Upsilon} \in \mathbb{R}^{KN\times 2|\mathcal{E}|K}$ is a linear mapping matrix satisfying

$$\begin{bmatrix} \sum_{j\in\mathcal{N}_1}(\boldsymbol{u}_{1j}^{(k+1)} - \boldsymbol{u}_{j1}^{(k+1)}) \\ \vdots \\ \sum_{j\in\mathcal{N}_N}(\boldsymbol{u}_{Nj}^{(k+1)} - \boldsymbol{u}_{jN}^{(k+1)}) \end{bmatrix} = \boldsymbol{\Upsilon}\boldsymbol{u}^{(k+1)}. \tag{A.31}$$

According to [26][7], both $\boldsymbol{u}^{(k+1)}$ and $\boldsymbol{u}^\star$ lie in the range space of $\boldsymbol{\Upsilon}^T$. Hence, one can show that

$$\|\boldsymbol{\Upsilon}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)\|^2 \geq \sigma_{\min}^2(\boldsymbol{\Upsilon})\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \tag{A.32}$$

where $\sigma_{\min}(\boldsymbol{\Upsilon}) > 0$ is the minimum nonzero singular value of $\boldsymbol{\Upsilon}$. From (A.30), we have that

$$\begin{aligned}
&\|\boldsymbol{G}(\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)})\|_2^2 \\
&= \| - \tilde{\boldsymbol{A}}^T(\nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\boldsymbol{y}^{(k-1)}) - \nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}^\star)) - \boldsymbol{\Upsilon}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)\|_2^2 \\
&\geq (1-\mu)\|\tilde{\boldsymbol{A}}^T(\nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\boldsymbol{y}^{(k)}) - \nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}^\star))\|_2^2 \\
&\qquad\qquad + (1-\frac{1}{\mu})\|\boldsymbol{\Upsilon}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)\|_2^2 \\
&\geq (1-\mu)\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})\|(\nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\boldsymbol{y}^{(k)}) - \nabla \boldsymbol{f}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}^\star))\|_2^2 \\
&\qquad\qquad + (1-\frac{1}{\mu})\sigma_{\min}^2(\boldsymbol{\Upsilon})\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \\
&\geq (1-\mu)\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})\|(\boldsymbol{y}^{(k-1)} - \boldsymbol{y}^\star)\|_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\tilde{\boldsymbol{A}}}^2 \\
&\qquad + (1-\frac{1}{\mu})\sigma_{\min}^2(\boldsymbol{\Upsilon})\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2,
\end{aligned} \tag{A.33}$$

where the first inequality is due to the fact that

$$\|\boldsymbol{a} + \boldsymbol{q}\|_2^2 \geq (1-\mu)\|\boldsymbol{a}\|_2^2 + (1-\frac{1}{\mu})\|\boldsymbol{q}\|_2^2 \tag{A.34}$$

for any $\boldsymbol{a}, \boldsymbol{q}$ and $\mu > 0$, the second inequality is obtained by setting $\mu > 1$ and (A.32), and the last inequality is by (12). Equation (A.33) implies that

$$\begin{aligned}
&\frac{\delta}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \leq \frac{\delta}{c(1-\frac{1}{\mu})\sigma_{\min}^2(\boldsymbol{\Upsilon})}\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\|_{\boldsymbol{G}^T\boldsymbol{G}}^2 \\
&+ \frac{\delta(\mu-1)\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})}{c(1-\frac{1}{\mu})\sigma_{\min}^2(\boldsymbol{\Upsilon})}\|(\boldsymbol{y}^{(k-1)} - \tilde{\boldsymbol{y}}^\star)\|_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\tilde{\boldsymbol{A}}}^2.
\end{aligned} \tag{A.35}$$

[7]Note that the matrix $\boldsymbol{\Upsilon}$ corresponds to matrix M_- in [26].

According to (A.35), (A.27) can hold true if

$$\|\boldsymbol{y}^{(k)} - \tilde{\boldsymbol{y}}^\star\|^2_{(1-\alpha)\mathbf{M}} \geq \delta\|\boldsymbol{y}^{(k)} - \tilde{\boldsymbol{y}}^\star\|^2_{\frac{1}{2}\boldsymbol{G}+\alpha\mathbf{M}},$$

$$(\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)})^T \left[\frac{1}{2}\boldsymbol{G} - \frac{1}{2}\tilde{\boldsymbol{A}}^T \boldsymbol{D}_{L_f} \boldsymbol{D}_\rho^{-1} \tilde{\boldsymbol{A}}\right] (\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)})$$

$$\geq \frac{\delta}{c(1-\frac{1}{\mu})\sigma^2_{\min}(\boldsymbol{\Upsilon})} \|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\|^2_{\boldsymbol{G}^T\boldsymbol{G}},$$

$$\|\boldsymbol{y}^{(k-1)} - \tilde{\boldsymbol{y}}^\star\|^2_{\alpha\mathbf{M}}$$

$$\geq \frac{\delta(\mu-1)\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})}{c(1-\frac{1}{\mu})\sigma^2_{\min}(\boldsymbol{\Upsilon})} \|(\boldsymbol{y}^{(k-1)} - \tilde{\boldsymbol{y}}^\star)\|^2_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\tilde{\boldsymbol{A}}},$$

which are respectively satisfied if the following three conditions can be satisfied for some $\delta > 0$

$$(1-\alpha)\mathbf{M} \succeq \delta\left(\frac{1}{2}\boldsymbol{G} + \alpha\mathbf{M}\right), \tag{A.36a}$$

$$\frac{1}{2}\boldsymbol{G} - \frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}} \succeq \frac{\delta}{c(1-\frac{1}{\mu})\sigma^2_{\min}(\boldsymbol{\Upsilon})}\boldsymbol{G}^T\boldsymbol{G}, \tag{A.36b}$$

$$\alpha(\boldsymbol{D}_{\sigma_f} - \frac{1}{2}\boldsymbol{D}_\rho) \succeq \delta\frac{(\mu-1)\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})}{c(1-\frac{1}{\mu})\sigma^2_{\min}(\boldsymbol{\Upsilon})}\boldsymbol{D}_{L_f}. \tag{A.36c}$$

Note that, given β_i's in (23), we have $\boldsymbol{D}_{\sigma_f} - \frac{1}{2}\boldsymbol{D}_\rho \succ \mathbf{0}$ and $\boldsymbol{G} - \frac{1}{\rho}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}} \succ \mathbf{0}$ (see (A.20) and (A.21)); moreover, since $\boldsymbol{A}_i$'s are full column rank, we have $\mathbf{M} \succ \mathbf{0}$. Hence there must exist some $\delta > 0$ such that the three conditions in (A.36) all hold true. ■

Appendix B
Proof of Theorem 3

Proof of Theorem 3(a): Let $\boldsymbol{x}^\star \triangleq [(\boldsymbol{x}_1^\star)^T, \ldots, (\boldsymbol{x}_N^\star)^T]^T$ and $\boldsymbol{\nu}^\star$ be a pair of optimal primal and dual solutions to (P2), and let $\tilde{\boldsymbol{\nu}}^\star \triangleq [(\boldsymbol{\nu}_1^\star)^T, \ldots, (\boldsymbol{\nu}_N^\star)^T]^T$ and $\{\boldsymbol{u}_{ij}^\star, \boldsymbol{v}_{ij}^\star, j \in \mathcal{N}_i\}_{i=1}^N$ be a pair of optimal primal and dual solutions to problem (26). Then they respectively satisfy the following optimality conditions

$$\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{x}_i^\star) + \partial g_i(\boldsymbol{x}_i^\star) + \mathbf{E}_i^T\boldsymbol{\nu}^\star = \mathbf{0}, i \in V, \tag{A.37}$$

$$\textstyle\sum_{i=1}^N \mathbf{E}_i\boldsymbol{x}_i^\star = \boldsymbol{q}, \tag{A.38}$$

$$\partial\varphi_i(\boldsymbol{\nu}_i^\star) + \tfrac{1}{N}\boldsymbol{q} + \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ji}^\star) = \mathbf{0},\ i \in V, \tag{A.39}$$

$$\boldsymbol{\nu}_i^\star = \boldsymbol{\nu}_j^\star\ \forall j \in \mathcal{N}_i, i \in V, \tag{A.40}$$

$$\boldsymbol{u}_{ij}^\star + \boldsymbol{v}_{ij}^\star = \mathbf{0}\ \forall j \in \mathcal{N}_i, i \in V. \tag{A.41}$$

where $\partial\varphi_i(\boldsymbol{\nu}_i^\star) = -\mathbf{E}_i\boldsymbol{x}_i^\star$ as $\boldsymbol{x}_i^\star$ is a maximizer to (6) with $\boldsymbol{\nu} = \boldsymbol{\nu}_i^\star$ [42]. Under Assumption 1, (A.2) implies that $\boldsymbol{\nu}^\star \triangleq \boldsymbol{\nu}_1^\star = \cdots = \boldsymbol{\nu}_N^\star$ and $\tilde{\boldsymbol{\nu}}^\star = \mathbf{1}_N \otimes \boldsymbol{\nu}^\star$.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Firstly, by recalling that $\boldsymbol{p}_i^{(k)} = \sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)})$, it follows from (41) and (A.39) that

$$\begin{aligned}\mathbf{0} = &-(\mathbf{E}_i\boldsymbol{x}_i^{(k)} - \tfrac{1}{N}\boldsymbol{q}) + \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)} + \boldsymbol{v}_{ji}^{(k+1)})\\ &+ c\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)}) \qquad \text{(A.42)}\\ = &-\mathbf{E}_i\boldsymbol{x}_i^\star + \tfrac{1}{N}\boldsymbol{q} + \textstyle\sum_{j\in\mathcal{N}_i}\boldsymbol{u}_{ij}^\star + \sum_{j\in\mathcal{N}_i}\boldsymbol{v}_{ji}^\star. \qquad \text{(A.43)}\end{aligned}$$

By multiplying $\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star$ to the both sides of (A.43), we obtain

$$\begin{aligned}&\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)} + \boldsymbol{v}_{ji}^{(k+1)} - \boldsymbol{u}_{ij}^\star - \boldsymbol{v}_{ji}^\star)^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star)\\ &+ c\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star)\\ &- (\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^\star)^T\mathbf{E}_i^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star) = \mathbf{0}. \qquad \text{(A.44)}\end{aligned}$$

Secondly, from the optimality of (43), we have that

$$\begin{aligned}\mathbf{0} = &\,\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) + \partial g(\boldsymbol{x}_i^{(k)})\\ &+ \tfrac{1}{2|\mathcal{N}_i|}\mathbf{E}_i^T\Big[\tfrac{1}{c}(\mathbf{E}_i\boldsymbol{x}_i^{(k)} - \tfrac{1}{N}\boldsymbol{q}) - \tfrac{1}{c}\textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)} + \boldsymbol{v}_{ji}^{(k)})\\ &+ \textstyle\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k-1)} + \boldsymbol{\nu}_j^{(k-1)})\Big] + \boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)})\\ = &\,\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) + \partial g(\boldsymbol{x}_i^{(k)}) + \mathbf{E}_i^T\boldsymbol{\nu}_i^{(k)}\\ &\qquad + \boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)}) \qquad \text{(A.45)}\\ = &\,\boldsymbol{A}_i^T(\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) - \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k)})) + \boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k)})\\ &+ \partial g(\boldsymbol{x}_i^{(k)}) + \mathbf{E}_i^T\boldsymbol{\nu}_i^{(k)} + \boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)}) \qquad \text{(A.46)}\\ = &\,\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^\star) + \partial g(\boldsymbol{x}_i^\star) + \mathbf{E}_i^T\boldsymbol{\nu}^\star, \qquad \text{(A.47)}\end{aligned}$$

where, in the first equality, we have added and subtracted $\frac{1}{2c|\mathcal{N}_i|}\mathbf{E}_i^T\mathbf{E}_i\boldsymbol{x}_i^{(k)}$ and defined

$$\boldsymbol{P}_i \triangleq \beta_i\boldsymbol{I}_K - \tfrac{1}{2c|\mathcal{N}_i|}\mathbf{E}_i^T\mathbf{E}_i; \qquad \text{(A.48)}$$

the second equality is due to (33); and the last equality is because $\boldsymbol{x}_i^\star$ is a maximizer to (6) with $\boldsymbol{\nu} = \boldsymbol{\nu}_i^\star$. Multiplying both (A.46) and (A.47) with $\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^\star$, combining with (A.44), and summing for

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

$i = 1, \ldots, N$, yields

$$\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) - \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k)}))^T\boldsymbol{A}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^\star) + \sum_{i=1}^{N}(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)})^T\boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^\star) +$$
$$\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k)}) - \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^\star))^T\boldsymbol{A}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^\star) + \sum_{i=1}^{N}(\partial g(\boldsymbol{x}_i^{(k)}) - \partial g(\boldsymbol{x}_i^\star))^T(\boldsymbol{x}_i^{(k)} - xb_i^\star)$$
$$+ \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)} + \boldsymbol{v}_{ji}^{(k+1)} - \boldsymbol{u}_{ij}^\star - \boldsymbol{v}_{ji}^\star)^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star)$$
$$+ c\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} + \boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_i^{(k-1)} - \boldsymbol{\nu}_j^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) = \boldsymbol{0}. \tag{A.49}$$

Similar to (A.15) and by (29), the fifth term in the LHS of (A.49) can be expressed as

$$\begin{aligned} &\textstyle\sum_{i=1}^N \sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k+1)} + \boldsymbol{v}_{ji}^{(k+1)} - \boldsymbol{u}_{ij}^\star - \boldsymbol{v}_{ji}^\star)^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) \\ =&\frac{2}{c}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)^T(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}). \end{aligned} \tag{A.50}$$

Moreover, the sixth term in the LHS of (A.49) can be shown as

$$\begin{aligned} &c\textstyle\sum_{i=1}^N \sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) \\ &\qquad + c\textstyle\sum_{i=1}^N \sum_{j\in\mathcal{N}_i}(\boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_j^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) \\ &= c\textstyle\sum_{i=1}^N |\mathcal{N}_i|(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) \\ &\qquad + c\textstyle\sum_{i=1}^N \sum_{j=1}^N [\boldsymbol{W}]_{i,j}(\boldsymbol{\nu}_j^{(k)} - \boldsymbol{\nu}_j^{(k-1)})^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^\star) \\ &= c(\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)})^T\boldsymbol{Q}(\boldsymbol{\nu}^{(k)} - \tilde{\boldsymbol{\nu}}^\star), \end{aligned} \tag{A.51}$$

where $\boldsymbol{Q} \triangleq (\boldsymbol{D} + \boldsymbol{W}) \otimes \boldsymbol{I}_M$. By applying (A.7), (A.8), (A.9), (A.50) and (A.51) to (A.49), one obtains

$$\begin{aligned} &\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{\mathbf{M}}^2 - \frac{1}{2}\|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}\|^2_{\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}} \\ &\qquad +(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)})^T\boldsymbol{P}(\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star) \\ &\qquad +c(\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)})^T\boldsymbol{Q}(\boldsymbol{\nu}^{(k)} - \tilde{\boldsymbol{\nu}}^\star) \\ &+ \frac{2}{c}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star)^T(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}) \leq 0, \end{aligned} \tag{A.52}$$

where $\boldsymbol{\nu}^{(k)} = [(\boldsymbol{\nu}_1^{(k)})^T, \ldots, (\boldsymbol{\nu}_N^{(k)})^T]^T$, $\boldsymbol{P} = \mathrm{blkdiag}\{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_N\} \succ \boldsymbol{0}$, and $\boldsymbol{D}_{\sigma_f}$, $\boldsymbol{D}_{L_f}$, $\boldsymbol{D}_\rho$, $\tilde{\boldsymbol{A}}$ and $\mathbf{M}$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

are all defined below (A.11). After applying (A.17) to (A.52), we obtain

$$
\begin{aligned}
&\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^\star\|^2_{\mathbf{M}+\frac{1}{2}\boldsymbol{P}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star\|_2^2+\frac{c}{2}\|\boldsymbol{\nu}^{(k)}-\tilde{\boldsymbol{\nu}}^\star\|^2_{\boldsymbol{Q}}\\
&\leq \|\boldsymbol{x}^{(k-1)}-\boldsymbol{x}^\star\|^2_{\frac{1}{2}\boldsymbol{P}}+\frac{1}{c}\|\boldsymbol{u}^{(k)}-\boldsymbol{u}^\star\|_2^2+\frac{c}{2}\|\boldsymbol{\nu}^{(k-1)}-\tilde{\boldsymbol{\nu}}^\star\|^2_{\boldsymbol{Q}}\\
&\quad-\frac{1}{2}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|^2_{\boldsymbol{P}-\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}}-\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^{(k)}\|_2^2\\
&\quad-\frac{c}{2}\|\boldsymbol{\nu}^{(k)}-\boldsymbol{\nu}^{(k-1)}\|^2_{\boldsymbol{Q}}.
\end{aligned}
\tag{A.53}
$$

It is easy to show that, under (47), it holds true that

$$
\sigma^2_{f,i}-\frac{\rho_i}{2}>0,\ \boldsymbol{P}_i-\frac{L^2_{f,i}}{\rho_i}\boldsymbol{A}_i^T\boldsymbol{A}_i\succ\mathbf{0},\ \forall\ i\in V,
\tag{A.54}
$$

for some $\sigma^2_{f,i}\leq\rho_i<2\sigma^2_{f,i}\ \forall i\in V$, which implies that

$$
\boldsymbol{P}\succ\mathbf{0},\ \boldsymbol{P}-\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}\succ\mathbf{0}.
$$

Thus, (A.53) implies that (R1) the sequence $\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^\star\|^2_{\mathbf{M}+\frac{1}{2}\boldsymbol{P}}+\frac{1}{c}\|\boldsymbol{u}^{(k+1)}-\boldsymbol{u}^\star\|_2^2+\frac{c}{2}\|\boldsymbol{\nu}^{(k)}-\tilde{\boldsymbol{\nu}}^\star\|^2_{\boldsymbol{Q}}$ converges for any optimal $\boldsymbol{x}^\star$ to (P2), and optimal $\tilde{\boldsymbol{\nu}}^\star$ and $\boldsymbol{u}^\star$ to problem (26); and (R2)

$$
\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\to\mathbf{0},\ \boldsymbol{u}^{(k+1)}-\boldsymbol{u}^{(k)}\to\mathbf{0},
\tag{A.55}
$$

$$
\|\boldsymbol{\nu}^{(k)}-\boldsymbol{\nu}^{(k-1)}\|^2_{\boldsymbol{Q}}\to\mathbf{0}.
\tag{A.56}
$$

Let $\hat{\boldsymbol{x}}=[(\hat{\boldsymbol{x}}_1)^T,\ldots,(\hat{\boldsymbol{x}}_N)^T]^T$, $\tilde{\hat{\boldsymbol{\nu}}}=[(\hat{\boldsymbol{\nu}}_1)^T,\ldots,(\hat{\boldsymbol{\nu}}_N)^T]^T$, $\hat{\boldsymbol{u}}_{ij}$ and $\hat{\boldsymbol{v}}_{ij}$ be a set of limit points of $\{\boldsymbol{x}^{(k)}\}$, $\{\boldsymbol{\nu}_1^{(k)},\ldots,\boldsymbol{\nu}_N^{(k)}\}$, $\{\boldsymbol{u}_{ij}^{(k)}\}$ and $\{\boldsymbol{v}_{ij}^{(k)}\}$, respectively. Firstly, by applying the fact of $\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\to\mathbf{0}$ to (A.46), we have

$$
\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\hat{\boldsymbol{x}}_i)+\partial g(\hat{\boldsymbol{x}}_i)+\mathbf{E}_i^T\hat{\boldsymbol{\nu}}_i=\mathbf{0},\ \forall\ i\in V.
\tag{A.57}
$$

Secondly, by (A.13), we have

$$
\hat{\boldsymbol{u}}_{ij}+\hat{\boldsymbol{v}}_{ij}=\mathbf{0}\ \forall j,i.
\tag{A.58}
$$

Thirdly, applying the fact of $\boldsymbol{u}_{ij}^{(k+1)}-\boldsymbol{u}_{ij}^{(k)}\to\mathbf{0}$ to (29a) yields

$$
\boldsymbol{\nu}_i^{(k)}-\boldsymbol{\nu}_j^{(k)}\to\mathbf{0}\Longrightarrow\hat{\boldsymbol{\nu}}\triangleq\hat{\boldsymbol{\nu}}_i=\hat{\boldsymbol{\nu}}_j\ \forall j\in\mathcal{N}_i, i\in V
\tag{A.59}
$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

The result of $\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_j^{(k)} \to \mathbf{0}\ \forall j, i$ and Assumption 1 implies that $\boldsymbol{\nu}^{(k)} - \mathbf{1}_N \otimes \boldsymbol{\nu}_i^{(k)} \to \mathbf{0}$ for any $i \in V$. Since the Laplacian matrix $\boldsymbol{L}\mathbf{1}_N = \mathbf{0}$ [32], one obtains

$$\begin{aligned}
&\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\|_{\boldsymbol{Q}}^2 \\
&\to (\mathbf{1}_N \otimes (\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)}))^T \boldsymbol{Q} (\mathbf{1}_N \otimes (\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)})) \\
&= (\mathbf{1}_N^T (\boldsymbol{D} + \boldsymbol{W}) \mathbf{1}_N) \|\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)}\|_2^2 \\
&= (\mathbf{1}_N^T (2\boldsymbol{D} - \boldsymbol{L}) \mathbf{1}_N) \|\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)}\|_2^2 \\
&= (2 \textstyle\sum_{j=1}^N |\mathcal{N}_j|) \|\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)}\|_2^2,
\end{aligned} \tag{A.60}$$

which, when combined with (A.56), further implies that

$$\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}_i^{(k-1)} \to \mathbf{0}\ \forall i \in V. \tag{A.61}$$

By applying (A.61) to (A.42), one obtains

$$\mathbf{0} = -\mathbf{E}_i \hat{\boldsymbol{x}}_i + \tfrac{1}{N}\boldsymbol{q} + \textstyle\sum_{j \in \mathcal{N}_i} \hat{\boldsymbol{u}}_{ij} + \sum_{j \in \mathcal{N}_i} \hat{\boldsymbol{v}}_{ji} \tag{A.62}$$

$$= \partial\varphi_i(\hat{\boldsymbol{\nu}}_i) + \tfrac{1}{N}\boldsymbol{q} + \textstyle\sum_{j \in \mathcal{N}_i} \hat{\boldsymbol{u}}_{ij} + \sum_{j \in \mathcal{N}_i} \hat{\boldsymbol{v}}_{ji}, \tag{A.63}$$

where $\partial\varphi_i(\hat{\boldsymbol{\nu}}_i) = -\mathbf{E}_i \hat{\boldsymbol{x}}_i$ since (A.57) implies that $\hat{\boldsymbol{x}}_i$ is a maximizer to (6) with $\boldsymbol{\nu} = \hat{\boldsymbol{\nu}}_i$ [42]. Finally, by summing (A.62) for $i = 1, \ldots, N$, followed by applying (A.14) and (A.58), one obtains

$$\textstyle\sum_{i=1}^N \mathbf{E}_i \hat{\boldsymbol{x}}_i = \boldsymbol{q}. \tag{A.64}$$

The results in (A.57), (A.58), (A.59), (A.63) and (A.64) imply that $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\nu}}$ are in fact a pair of optimal primal and dual solutions to (P2), and $\tilde{\hat{\boldsymbol{\nu}}}$ and $\{\hat{\boldsymbol{u}}_{ij}, \hat{\boldsymbol{v}}_{ij}\}$ are a pair of optimal primal and dual solutions to problem (26) [see (A.37) to (A.41)]. Thus, according to (R1), the sequence $\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{\mathbf{M}+\frac{1}{2}\boldsymbol{P}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \hat{\boldsymbol{u}}\|_2^2 + \frac{c}{2}\|\boldsymbol{\nu}^{(k)} - \tilde{\hat{\boldsymbol{\nu}}}\|_{\boldsymbol{Q}}^2$ in fact converges to zero and thereby $\boldsymbol{x}^{(k)} \to \hat{\boldsymbol{x}}$, $\boldsymbol{u}^{(k+1)} \to \hat{\boldsymbol{u}}$ and $\boldsymbol{\nu}_i^{(k)} \to \hat{\boldsymbol{\nu}}\ \forall i \in V$. ■

Proof of Theorem 3(b): We assume that $\phi_i(\boldsymbol{x}_i) = f_i(\boldsymbol{A}_i \boldsymbol{x}_i)$, $\boldsymbol{A}_i$ has full column rank and $\mathbf{E}_i$ has full row rank, for all $i \in V$. Denote $\boldsymbol{r}^{(k)} \triangleq \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{\alpha\mathbf{M}+\frac{1}{2}\boldsymbol{P}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 + \frac{c}{2}\|\boldsymbol{\nu}^{(k)} - \tilde{\boldsymbol{\nu}}^\star\|_{\boldsymbol{Q}}^2$ for some $\alpha > 0$. One can write (A.53) as follows

$$\begin{aligned}
&\boldsymbol{r}^{(k)} + \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{(1-\alpha)\mathbf{M}}^2 + \|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^\star\|_{\alpha\mathbf{M}}^2 \\
&\leq \boldsymbol{r}^{(k-1)} - \frac{1}{2}\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_{\boldsymbol{P}-\frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}}^2 \\
&\quad - \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2^2 - \frac{c}{2}\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\|_{\boldsymbol{Q}}^2.
\end{aligned}$$

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

Therefore, it suffices to show that, for some $\delta > 0$,

$$\begin{aligned}
&\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|^2_{(1-\alpha)\mathbf{M}} + \|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^\star\|^2_{\alpha\mathbf{M}} \\
&+ \frac{1}{2}\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|^2_{\boldsymbol{P}-\frac{1}{2}\tilde{\boldsymbol{A}}^T\boldsymbol{D}_{L_f}\boldsymbol{D}_\rho^{-1}\tilde{\boldsymbol{A}}} + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2^2 \\
&+ \frac{c}{2}\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\|^2_{\boldsymbol{Q}} \geq \delta \boldsymbol{r}^{(k)}.
\end{aligned} \tag{A.65}$$

Firstly, from (A.45) and (A.47), we have that (without g_i's)

$$\begin{aligned}
\boldsymbol{A}_i^T(\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) - \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^\star)) + \mathbf{E}_i^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star)& \\
+ \boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)}) = \mathbf{0}.&
\end{aligned} \tag{A.66}$$

By applying (A.34) to (A.66), we have, for some $\mu_1 > 1$,

$$\begin{aligned}
&\|\boldsymbol{P}_i(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^{(k-1)})\|^2 \\
&\geq (1-\mu_1)\|\boldsymbol{A}_i^T(\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) - \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^\star))\|^2 \\
&\qquad + (1-\frac{1}{\mu_1})\|\mathbf{E}_i^T(\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star)\|_2^2 \\
&\geq (1-\mu_1)L_{f,i}\lambda^2_{\max}(\boldsymbol{A}_i^T\boldsymbol{A}_i)\|\boldsymbol{x}_i^{(k-1)} - \boldsymbol{x}_i^\star)\|_2^2 \\
&\qquad + (1-\frac{1}{\mu_1})\lambda_{\min}(\mathbf{E}_i\mathbf{E}_i^T)\|\boldsymbol{\nu}_i^{(k)} - \boldsymbol{\nu}^\star\|_2^2,
\end{aligned} \tag{A.67}$$

where the second inequality is obtained by (12). Note that $\boldsymbol{D} + \boldsymbol{W} = 2\boldsymbol{D} - \boldsymbol{L} \preceq 2\boldsymbol{D}$ as $\boldsymbol{L} \succeq \mathbf{0}$ [32]. Hence, we have

$$\begin{aligned}
&\frac{c\delta}{2}\|\boldsymbol{\nu}^{(k)} - \tilde{\boldsymbol{\nu}}^\star\|^2_{\boldsymbol{Q}} \leq c\delta\|\boldsymbol{\nu}^{(k)} - \tilde{\boldsymbol{\nu}}^\star\|^2_{\boldsymbol{D}\otimes\boldsymbol{I}_M} \\
&\leq c\delta\tau_1\|(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)})\|^2_{\boldsymbol{P}^T\boldsymbol{P}} + c\delta\tau_2\|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^\star)\|_2^2,
\end{aligned} \tag{A.68}$$

where the second inequality is due to (A.67), $\tau_1 = \max_{i\in V}\left\{\frac{|\mathcal{N}_i|}{(1-\frac{1}{\mu_1})\lambda_{\min}(\mathbf{E}_i\mathbf{E}_i^T)}\right\} > 0$ and $\tau_2 = \max_{i\in V}\left\{\frac{(\mu_1-1)\lambda^2_{\max}(\boldsymbol{A}_i^T\boldsymbol{A}_i)|\mathcal{N}}{(1-\frac{1}{\mu_1})\lambda_{\min}(\mathbf{E}_i\mathbf{E}_i^T}\right.$ 0 are finite given that $\mathbf{E}_i$'s have full row rank.

Secondly, upon stacking (A.43) for all $i \in V$ and applying (A.3) and (A.12), one obtains

$$\begin{aligned}
\boldsymbol{\Upsilon}(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star) + c\boldsymbol{Q}(\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)})& \\
- \mathbf{E}(\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star) = \mathbf{0},&
\end{aligned} \tag{A.69}$$

where $\mathbf{E} = \mathrm{blkdiag}\{\mathbf{E}_1, \ldots, \mathbf{E}_N\}$ and $\boldsymbol{\Upsilon}$ is given in (A.31). Analogously, by applying (A.34) to (A.69) and by (A.32), one can show that, for some $\mu_2 > 1$,

$$\frac{\delta}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|_2^2 \leq \frac{\delta}{c\tau_3}\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{\mathbf{E}^T\mathbf{E}}^2 + \frac{\delta(\mu_2-1)c}{\tau_3}\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\|_{\boldsymbol{Q}}^2, \tag{A.70}$$

where $\tau_3 = (1-\frac{1}{\mu_2})\sigma_{\min}^2(\boldsymbol{\Upsilon}) > 0$. By (A.68) and (A.70), sufficient conditions for satisfying (A.65) are therefore given by: $\forall i \in V$,

$$(1-\alpha-\delta\alpha)(\sigma_{f,i}^2 - \frac{\rho_i}{2})\boldsymbol{A}_i^T\boldsymbol{A}_i \succeq \frac{\delta}{2}\boldsymbol{P}_i + \frac{\delta}{c\tau_3}\boldsymbol{A}_i^T\boldsymbol{A}_i, \tag{A.71a}$$

$$\alpha(\sigma_{f,i}^2 - \frac{\rho_i}{2})\boldsymbol{A}_i^T\boldsymbol{A}_i \succeq c\delta\tau_2\boldsymbol{I}_K, \tag{A.71b}$$

$$\frac{1}{2}\boldsymbol{P}_i - \frac{L_{f,i}^2}{2\rho_i}\boldsymbol{A}_i^T\boldsymbol{A}_i \succeq c\delta\tau_1\boldsymbol{P}_i^T\boldsymbol{P}_i, \tag{A.71c}$$

$$\frac{1}{2} \geq \frac{\delta(\mu_2-1)}{\tau_3}. \tag{A.71d}$$

Under (A.54) and full column rank $\boldsymbol{A}_i$'s, we see that (A.71) is true for some $\delta > 0$. The proof is complete. ■

REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proc. Grid Computing Environments Workshop*, Austin, TX, USA, Nov. 12-16, 2008, pp. 1–10.

[2] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning- Parallel and Distributed Approaches*. Cambridge University Press, 2012.

[3] G. R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2007.

[4] S. Ghosh, *Distributed Systems- An Algorithmic Approach*. Chapman & Hall/CRC Computer & Information Science Series, 2007.

[5] A. Nedić, A. Ozdaglar, , and A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010.

[6] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *Proc. IEEE CDC*, Cancun, Mexica, Dec. 9-11, 2008, pp. 4185–4190.

[7] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Automatic Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.

[8] J. Chen and A. H. Sayed, "Diffusion adaption strategies for distributed optimization and learning networks," *IEEE. Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[9] M. Elad, *Sparse and Redundant Rerpesentations*. New York, NY, USA: Springer Science + Business Media, 2010.

[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, no. 1, pp. 1–122, 2011.

[11] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[12] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE. Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, April 2012.

[13] D. P. Bertsekas, *Network Optimization : Contribuous and Discrete Models*. Athena Scientific, 1998.

[14] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordianted beamforming with imperfect CSI: An ADMM approach," *IEEE Trans. Signal Processing*, vol. 60, no. 6, pp. 2988–3003, 2012.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[16] S. S. Ram, A. Nedić, and V. V. Veeravalli, "A new class of distributed optimization algorithm: Application of regression of distributed data," *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.

[17] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE. Trans. Automatic Control.*, vol. 59, no. 6, pp. 1524–1538, June 2014.

[18] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[19] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Dec. 2010.

[20] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Stat. Soc. B*, vol. 58, pp. 267–288, 1996.

[21] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE. Trans. Signal Process.*, vol. 60, no. 10, pp. 2718–2723, May 2013.

[22] E. Wei and A. Ozdaglar, "On the $O(1/K)$ convergence of asynchronous distributed alternating direction method of multipliers," in *Proc. IEEE GlobalSIP*, Austin, TX, USA, Dec. 3-5, 2013, pp. 551–554.

[23] ——, "Distributed alternating direction method of multipliers," in *Proc. IEEE CDC*, Maui, HI, USA, Dec. 10-13, 2012, pp. 5445–5450.

[24] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice CAAM technical report 12-14, 2012.

[25] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," available on arxiv.org.

[26] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, April 2014.

[27] D. Jakovetic, J. M. F. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," available on arxiv.org.

[28] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[29] B. He and X. Yuan, "Linearized alternating direction method of multipliers with gaussian back substitution for separable convex programming," *Numerial Algebra. Control and Optimization*, vol. 3, no. 2, pp. 247–260, 2013.

[30] S. Ma, "Alternating proximal gradient method for convex minimization," available on http://www.optimization-online.org/.

[31] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Large-scale sparse logistic regression," in *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, June 28 - July 1, 2009, pp. 547–556.

[32] F. R.-K. Chung, *Spectral graph theory*. CBMS Regional Conference Series in Mathematics, No. 92, 1996.

[33] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," available on arxiv.org.

[34] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," available on arxiv.org.

[35] B. He, Z. Peng, and X. Wang, "Proximal alternating direction-based contraction methods for separable linearly constrained convex optimization," *Frontiers of Math. in China*, vol. 6, no. 1, pp. 79–114, 2011.

[36] S. Ma and S. Zhang, "An extragradient-based alternating direction method for convex minimization," available on arxiv.org.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.

[37] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," in *Proc. IEEE ICASSP*, Florence, Italy, May 4-9, 2014, pp. 5445–5450.

[38] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex analysis and optimization*. Cambridge, Massachusetts: Athena Scientific, 2003.

[39] M. E. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3753–3764, 2008.

[40] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[41] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 1–112, 2013.

[42] S. Boyd and A. Mutapcic, "Subgradient methods," avaliable at www.stanford.edu/class/ee392o/subgrad_method.pdf.

This is a manuscript of an article from IEEE Transactions on Signal Processing 63 (2015):482,
doi: 10.1109/TSP.2014.2367458. Posted with permission.