Broadcast-based Distributed Alternating Direction Method of Multipliers

Ali Makhdoumi and Asuman Ozdaglar MIT, Cambridge, MA 02139
Emails: makhdoum@mit.edu, asuman@mit.edu

***Abstract*—We consider a multi agent optimization problem where a network of agents collectively solves a global optimization problem with the objective function given by the sum of locally known convex functions. We propose a fully distributed broadcast-based Alternating Direction Method of Multipliers (ADMM), in which each agent broadcasts the outcome of his local processing to all his neighbors. We show that both the objective function values and the feasibility violation converge with rate $O(\frac{1}{T})$, where T is the number of iterations. This improves upon the $O(\frac{\log T}{\sqrt{T}})$ convergence rate of subgradient-based methods. We also characterize the effect of network structure and the choice of communication matrix on the convergence speed. Because of its broadcast nature, the storage requirements of our algorithm are much more modest compared to distributed algorithms that use edge-based pairwise communication between agents.**

I. Introduction

A. Motivation and Contributions

Many of today's data processing takes place over large-scale networks in which information is gathered and processed locally. A prominent example is a machine learning application where a network of agents aims to estimate a model (or a parameter) to optimize a global loss function using decentralized computations based on locally collected data. This has motivated a recent burgeoning literature on designing decentralized optimization algorithms for efficient processing of this local information. Many of these problems are formulated as a global optimization problem where the objective function is the sum of local objective functions of agents that are connected through a network [26], [39], [38], [27], [25], [19], [18]. Many of the distributed algorithms studied in the literature consider each agent performing a local optimization according to subgradient-based methods and then communicating the outcome of his processing with other agents through edge-based pairwise exchanges.

In this paper, we consider a multi-agent optimization problem where a network of agents collectively minimizes the sum of locally known convex functions $f_i(x)$ using information exchange over the network.[1] The contribution of this work is threefold. We first reformulate the multiagent optimization problem as a constrained optimization problem. We denote the variable of node i by x_i. The requirement that all x_i's are equal is imposed by introducing a constraint per node that ensures each x_i is equal to a weighted average of the variables of the neighbors x_j. These weights define the communication matrix, where each row corresponds to a node and the non-zero entries in that row contain the (negative) weights given to the neighbors of that node (diagonal entries is given by the sum of the weights, hence the row sum of the communication matrix is equal to zero). The communication matrix is a generalized Laplacian matrix (Laplacian of a weighted graph [1]) of the underlying graph, and provides flexibility in choosing the weights over a given network to improve the performance of the algorithm. Our algorithm do not require any centralized processing to obtain the communication matrix. In fact, agents can implement the algorithm only with knowing their own degree. We then separate each node constraint, that is specified in terms of a linear equality between variables of the neighbors of that node, into multiple constraints that contain only one of the neighbor variables. This allows each node variable to be updated simultaneously without the need for an order among these variables in each iteration of ADMM iteration (see [3], [39], [11], [42]). We provide a reduction that enables implementation of the algorithm by broadcasting node-based dual variables, instead of maintaining and updating edge-based primal and dual variables. This lead to an algorithm with much more modest storage requirements compared to othe existing ADMM-based distributed algorithms (see [34]). Second, we show that the difference between objective function values of the iterates and the optimal objective values as well as the norm of the feasibility violation of the iterates converge with rate $O(\frac{1}{T})$, where T is the number of iterations. Third, we characterize the dependence of the underlying network structure and choice of communication matrix on the performance (speed of convergence) of our proposed algorithm. In particular, we provide bounds on the objective function value improvement and feasibility violation as a function of the number of edges, maximum degree in the network and the singular values of the communication matrix. We also provide numerical results illustrating the effect of network structure as well as the choice of communication matrix on the performance of the algorithm.

B. Related Works

Our paper is related to a large recent literature on distributed methods for solving multi agent optimization problem over networks. Much of this literature builds on the seminal works [36], [37], which proposed gradient methods that can parallelize computation across multiple processors. A number of recent papers considered the multi agent optimization problem and proposed subgradient type distributed methods that use consensus or averaging mechanisms for

[1] Throughout the paper, we use the terms node and agent interchangeably.

aggregating information among the agents over the networks (see [25], [27], [19], [19], [30], [17], [35], [20], [24], [43], [7], [9]). These papers are related to the vast literature on the consensus problem, where agents have a more specific goal of aligning their estimates (see [14], [28], [40], [29]). For convex local objective functions, these methods achieve $O(\frac{\log T}{\sqrt{T}})$ convergence rate, where T is the number of iterations. The recent paper [41] adopted stronger assumptions on the local objective functions, i.e., they are convex with Lipschitz continuous gradients, and show that that a distributed gradient method with averaging converges at rate $O(\frac{1}{T})$ to an error neighborhood (with the further assumption of strongly convex local objective functions, linear rate is achieved to an error neighborhood). Another contribution [15] assumed local objective functions have continuous and bounded gradients and used Nesterov's acceleration to design a distributed algorithm with rate $O(\frac{\log T}{T})$. The recent paper [32], proposes a gradient-based distributed algorithm for convex objective function with Lipschitz continuous gradients and show its convergence with rate $O(\frac{1}{T})$.

There is another recent strand of literature that uses ADMM-based algorithms for multi agent optimization (see [6] and [10]). Several papers have demonstrated computationally the remarkable potential of ADMM for handling distributed optimization problems for decentralized estimation and compressive sensing applications (see [31], [22], [21], [16], [23]). For the case when the objective function is the sum of two convex functions, the convergence rate of ADMM was shown to be $O(\frac{1}{T})$ in [12]. Recent papers [13], [8] established global linear convergence rate under stronger assumption (i.e., strongly convex functions or error bound conditions). The paper [33] considered using ADMM for distributed multiagent optimization with strongly convex objective functions. Our paper is most related to [38], [34] which studied synchronous and asynchronous ADMM algorithms for multiagent optimization and showed $O(\frac{1}{T})$ convergence rate. These algorithms maintain edge-based primal and dual variables and involve pairwise communication. In contrast, in this paper, we propose a node-based formulation which leads to a broadcast based implementation and eliminates the significant increase in the number of variables because of an edge-based formulation.

C. Outline

The organization of the paper is as follows. In Section II, we give the problem formulation and the reformulation used to develop a distributed algorithm. In Section III, we propose a distributed broadcast-based ADMM algorithm. In Section IV, we consider the sequence of ergodic (time) averages of the estimates generated by this algorithm. We show that both the difference of the objective function values from the optimal value and the feasibility violation at the ergodic average sequence converge to 0 at rate $O(\frac{1}{T})$. In Section V, we show that the performance of our algorithm depends on the network structure through the singular values of the communication matrix. We also discuss the choice of communication matrix. In Section VI, we report the performance of our algorithm for networks with different structures and different communication matrices to illustrate the dependence of convergence rate on both network structure and the choice of communication matrix. Section VII contains our concluding remarks.

D. Basic Notations

A vector $\mathbf{x}$ is viewed as a column vector. For a matrix A, we write $[A]_i$ to denote the ith column of matrix A, $[A]^i$ to denote the ith row of matrix A, and A_{ij} to denote the entry at ith row and jth column. For a vector $\mathbf{x}$, x_i denotes the ith component of the vector. We use $\mathbf{x}'$ and A' to denote the transpose of a vector $\mathbf{x}$ and a matrix A, respectively. We use standard Euclidean norm for a vector $\mathbf{x}$ in $\mathbb{R}^n$, i.e., we have that $||\mathbf{x}|| = \sqrt{\sum_{i=1}^n x_i^2}$. For a set S, $|S|$ denotes the number of elements of S.

II. Problem Formulation

Consider a connected undirected network represented by the graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of agents and E is the set of edges. For any i, let $N(i)$ denote the set of neighbors of agent i including agent i itself, i.e., $N(i) = \{j \mid (i,j) \in E\} \cup \{i\}$ and let d_i denote the degree of agent i, i.e., $|N(i)| = d_i + 1$.

The goal of the agents is to collectively solve the following optimization problem

$$\min_{x\in\mathbb{R}} \sum_{i=1}^n f_i(x), \tag{1}$$

where $f_i : \mathbb{R} \to \mathbb{R}$ is a convex function. We assume that information is distributed across agents, i.e., the function f_i is only known to agent i. We therefore refer to f_i as the local objective function of agent i.[2]

We introduce a variable x_i for each i and write the objective function of problem (1) as $\sum_{i=1}^n f_i(x_i)$, so that the objective function is decoupled across the agents. The constraint that all the x_i's are equal can be imposed using the following matrix.

Definition 1. Let A be a matrix whose entries satisfy the following: for any $i = 1, \ldots, n$, $A_{ij} = 0$ for $j \notin N(i)$, $A_{ij} < 0$ for $j \in N(i) \setminus \{i\}$, and $A_{ii} = -\sum_{j\in N(i)\setminus\{i\}} A_{ij}$. We refer to A as the *communication matrix*. Note that sum of each row of A is zero.

Matrix A is the *Laplacian matrix* of an undirected possibly weighted graph and generalizes the standard Laplacian matrix of a graph as described in the following example (see e.g. [1] for details on the generalized Laplacian matrix).

For example, if we let the entries of the communication matrix A be $A_{ii} = d_i$ and $A_{ij} = -1$ for any $i = 1, \ldots, n$ and $j \in N(i) \setminus \{i\}$. With this choice of weights, the A matrix becomes the standard Laplacian matrix of the underlying graph.

Let $\mathbf{x} \in \mathbb{R}^n$ shows a vector whose ith element is x_i. In the following lemma, we show that the constraint $A\mathbf{x} = 0$ for any communication matrix ensures that $x_i = x$ for some $x \in \mathbb{R}$ and for any $i = 1, \ldots, n$.

[2]In this paper, we assume $x \in \mathbb{R}$. All the results can be extended to the case where $\mathbf{x} \in \mathbb{R}^m$ for some m.

Lemma 1. *For any $\mathbf{x} \in \mathbb{R}^n$, if $A\mathbf{x} = 0$, then all x_i's are equal.*

Proof: Let x_{i_0} be the largest number among $x_1, \ldots, x_n$. From $A\mathbf{x} = 0$, we have that

$$x_{i_0} = \frac{1}{\sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j}} \sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j} x_j.$$

This shows that

$$\begin{aligned} x_{i_0} &= \frac{1}{\sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j}} \sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j} x_j \\ &\geq \frac{1}{\sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j}} \sum_{j \in N(i_0) \setminus \{i_0\}} A_{i_0 j} x_{i_0} = x_{i_0}, \end{aligned}$$

and the equality holds if and only if for all $j \in N(i_0)$, we have $x_j = x_{i_0}$. Since the graph is connected, repeating the same argument shows that $x_i = x_{i_0}$ for all $i \in V$. ■

Using Lemma 1, we can reformulate optimization problem (1) as

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} & \sum_{i=1}^n f_i(x_i) \\ \text{s.t. } & A\mathbf{x} = 0. \end{aligned} \tag{2}$$

Assumption 1. The optimal solution set of problem (2) is non-empty.

Note 1. Since the function $\sum_{i=1}^n f_i(x_i)$ is convex and the constraints are linear, under Assumption 1, the dual problem of (2) has an optimal solution and there is no duality gap (see section 6 of [2]). Note that a primal-dual optimal solution $(\mathbf{x}^*, \mathbf{p}^*)$ is a saddle point of the Lagrangian function $L(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^n f_i(x_i) + \mathbf{p}'(A\mathbf{x})$, i. e.,

$$L(\mathbf{x}^*, \mathbf{p}) \leq L(\mathbf{x}^*, \mathbf{p}^*) \leq L(\mathbf{x}, \mathbf{p}^*), \tag{3}$$

for all $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{p} \in \mathbb{R}^n$.

III. Broadcast-Based Distributed ADMM

In this section, we propose a broadcast-based implementation of ADMM algorithm to solve problem (2). We first use a reformulation technique used in [3] (see section 3.4 of [3]), which allows us to separate each constraint associated with a node into multiple constraints that involve only the variable corresponding to one of its neighboring nodes. We expand the constraint $A\mathbf{x} = 0$ so that for each node i, we have

$$\sum_{j \in N(i)} A_{ij} x_j = 0.$$

We let $A_{ij} x_j = z_{ij}$ to obtain the following reformulation:

$$\min_{x_i,\ z_{ij}} \sum_{i=1}^n f_i(x_i) \tag{4}$$

$$\text{s.t. } A_{ij} x_j = z_{ij}, \quad \text{for } i = 1, \ldots, n,\ j \in N(i), \tag{5}$$

$$\sum_{j \in N(i)} z_{ij} = 0, \quad \text{for } i = 1, \ldots, n.$$

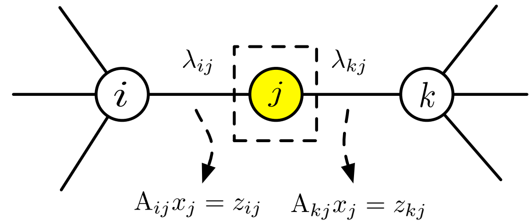

Fig. 1: Constraints variable x_j appears in with the corresponding Lagrange multipliers.

For each equality constraint in (5), we let λ_{ij} be the corresponding Lagrange multiplier and form the augmented Lagrangian function by adding a quadratic penalty for feasibility violation to the Lagrangian function as

$$\begin{aligned} L_c(\mathbf{x}, \{z_{ij}\}, \{\lambda_{ij}\}) &= \sum_{i=1}^n f_i(x_i) + \sum_{i=1}^n \sum_{j \in N(i)} \lambda_{ij}(A_{ij} x_j - z_{ij}) \\ &+ \frac{c}{2} \sum_{i=1}^n \sum_{j \in N(i)} (A_{ij} x_j - z_{ij})^2. \end{aligned}$$

Note that for each j, variable x_j appears in constraints associated with his neighboring nodes (see Figure 1 for an illustration together with the associated Lagrange multipliers).

We now use ADMM algorithm to solve (4). ADMM algorithm generates primal-dual sequences $\{x_j(t)\}, \{z_{ij}(t)\}$, and $\{\lambda_{ij}(t)\}$ which at iteration $t+1$ are updated as follows:

1) For any $j = 1, \ldots, n$, we update x_j as

$$x_j(t+1) \in \text{argmin}_{x_j \in \mathbb{R}} \left[f_j(x_j) + \sum_{i \in N(j)} \lambda_{ij}(t) A_{ij} x_j + \frac{c}{2} \left(A_{ij} x_j - z_{ij}(t)\right)^2 \right]. \tag{6}$$

2) For any $i = 1, \ldots, n$, we update the vector $\mathbf{z}_i = [z_{ij}]_{j \in N(i)}$ as

$$\mathbf{z}_i(t+1) \in \text{argmin}_{\mathbf{z}_i \in Z_i} \left[\sum_{j \in N(i)} \left(-\lambda_{ij}(t) z_{ij} + \frac{c}{2} (A_{ij} x_j(t+1) - z_{ij})^2 \right) \right], \tag{7}$$

where $Z_i = \{\mathbf{z}_i \mid \sum_{j \in N(i)} z_{ij} = 0\}$.

3) For $i = 1, \ldots, n$ and $j \in N(i)$ we update λ_{ij} as

$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + c(A_{ij} x_j(t+1) - z_{ij}(t+1)) \tag{8}$$

Note that the quadratic optimization (7) in step 2 can be solved in closed form as

$$z_{ij}(t+1) = A_{ij} x_j(t+1) + \frac{\lambda_{ij}(t) - p_i(t+1)}{c}, \tag{9}$$

for $i = 1, \ldots, n$ and $j \in N(i)$, where $p_i(t+1)$ is the Lagrange multiplier associated with the constraint $\sum_{j \in N(i)} z_{ij}(t+1) =$

0. Taking the summation of (9) for all $j \in N(i)$, we obtain

$$\sum_{j\in N(i)} z_{ij}(t+1) = 0 = \sum_{j\in N(i)} A_{ij}x_j(t+1) + \frac{1}{c}\left(\sum_{j\in N(i)} \lambda_{ij}(t)\right) - \frac{1}{c}(d_i+1)p_i(t+1),$$

which results in

$$p_i(t+1) = \frac{1}{d_i+1}\sum_{j\in N(i)} \lambda_{ij}(t) + \frac{c}{d_i+1}\sum_{j\in N(i)} A_{ij}x_j(t+1). \tag{10}$$

Comparing (9) and (8), we obtain $\lambda_{ij}(t+1) = p_i(t+1)$ for all $i = 1,\ldots,n$ and $j \in N(i)$. Substituting this equation into (10), we obtain

$$\begin{aligned} p_i(t+1) &= p_i(t) + \frac{c}{d_i+1}\sum_{j\in N(i)} A_{ij}x_j(t+1) \\ &= p_i(t) + \frac{c}{d_i+1}\left([A]^i\right)' \mathbf{x}(t+1), \end{aligned} \tag{11}$$

for $i = 1,\ldots,n$. Substituting (11) and $\lambda_{ij}(t) = p_i(t)$ into (9), we obtain $z_{ij}(t+1) = A_{ij}x_j(t+1) - \frac{1}{d_i+1}\left([A]^i\right)' \mathbf{x}(t+1)$. Writing this equation for t, we have

$$z_{ij}(t) = A_{ij}x_j(t) - \frac{1}{d_i+1}\left([A]^i\right)' \mathbf{x}(t). \tag{12}$$

Substituting this equation and $\lambda_{ij}(t) = p_i(t)$ into (6), we obtain

$$x_j(t+1) \in \text{argmin}_{x_j\in\mathbb{R}} f_j(x_j) + \sum_{i\in N(j)} [p_i(t)A_{ij}x_j + \frac{c}{2}(A_{ij}(x_j - x_j(t)) + y_i(t))^2], \tag{13}$$

where $y_i(t) = \frac{1}{d_i+1}\left([A]^i\right)' \mathbf{x}(t)$. Note that by this definition of $y_i(t)$, (12) becomes

$$z_{ij}(t) = A_{ij}x_j(t) - y_i(t). \tag{14}$$

This reduction shows that the algorithm need not maintain primal and dual variables $z_{ij}(t)$ and $\lambda_{ij}(t)$ for each i and its neighbors j, but instead can operate with the lower dimensional node-based dual variable $p_i(t)$. The dual variable $p_i(t)$ can be updated as in (11) using primal variables $x_j(t)$ for all $j \in N(i)$. The primal variable $x_j(t)$ can be updated as in (13) using $p_i(t)$ and $y_i(t)$ for $i \in N(j)$, where $y_i(t)$ is node i's estimate of the primal variable (obtained as the average of primal variables of his own neighbors). The steps of the algorithm are summarized in Algorithm 1.

This algorithm can be implemented in a distributed way using the following information structure and communication pattern. Each node i maintains local variables $x_i(t)$, $y_i(t)$, and $p_i(t)$ and updates these variables using communication with its neighbors as follows:

- At the end of iteration t, each node i sends out $p_i(t)$ and $y_i(t)$ to all of its neighbors and then each node such as j uses $y_i(t)$ and $p_i(t)$ of all $i \in N(j)$ to solve the optimization problem

$$x_j(t+1) \in \text{argmin}_{x_j\in\mathbb{R}} f_j(x_j) + \sum_{i\in N(j)} p_i(t)A_{ij}x_j + \frac{c}{2}\sum_{i\in N(j)} (y_i(t) + A_{ij}(x_j - x_j(t)))^2,$$

  in order to find $x_j(t+1)$.
- Each node j sends out $x_j(t+1)$ to all of its neighbors and then each node such as i computes

$$y_i(t+1) = \frac{1}{d_i+1}\sum_{j\in N(i)} A_{ij}x_j(t+1).$$

- Each node i computes

$$p_i(t+1) = p_i(t) + cy_i(t+1).$$

Algorithm 1 Broadcast-Based Distributed ADMM

- **Initialization:** $x_i(0)$, $y_i(0)$, and $p_i(0)$, for any $i \in V$ and A.
- **Algorithm:**
  1) for $i = 1,\ldots,n$, let
  $$x_i(t+1) \in \text{argmin}_{x_i\in\mathbb{R}} f_i(x_i) + \sum_{j\in N(i)} p_j(t)A_{ji}x_i + \frac{c}{2}(y_j(t) + A_{ji}(x_i - x_i(t)))^2.$$
  2) for $i = 1,\ldots,n$, let
  $$y_i(t+1) = \frac{1}{d_i+1}\sum_{j\in N(i)} A_{ij}x_j(t+1).$$
  3) for $i = 1,\ldots,n$, let $p_i(t+1) = p_i(t) + cy_i(t+1)$.
- **Output:** $\{x_i(t)\}_{t=0}^{\infty}$ for any $i \in V$.

We next contrast the communication requirements of this algorithm with those of the edge-based distributed ADMM algorithm of [34]. Following the previous description, each node broadcasts 3 different variables (i.e., performs 3 communications) per iteration to its neighbors and the algorithm therefore requires $3n$ communications per iteration, whereas [34] requires n communications. However, the storage requirement of our algorithm is $3n$ for all nodes as node i requires to keep $x_i(i)$, $y_i(t)$ and $p_i(t)$, whereas [34] requires to store $\sum_{i=1}^{n}(d_i+1) = n + 2|E|$ many variables. Therefore, we significantly gain in terms of the storage requirement (we gain a factor of n when $|E| = \Theta(n^2)$) and lose a factor of 3 in terms of the communication requirement, using a broadcast-based algorithm for solving problem (1).[3]

[3]Note that the broadcast-based ADMM can be implemented in a distributed fashion in terms of the original variables $\lambda_{ij}(t)$ and $z_{ij}(t)$ as follows: each node j maintains the primal variable $x_j(t)$ as well as the variables $\lambda_{ij}(t)$ and $z_{ij}(t)$ for each of its neighboring nodes (see Figure 1). Each j updates $x_j(t)$ using his local variables $\lambda_{ij}(t)$ and $z_{ij}(t)$ (see (6)). Node j can then update $z_{ij}(t)$ using primal variables $x_i(t)$ received from his neighbors (see (12)) and $\lambda_{ij}(t)$ using his local variables $x_j(t+1)$ and $z_{ij}(t+1)$ (see (8)). It can be seen that this implementation has the same communication and storage requirements as the edge-based ADMM algorithm of [34]. Hence the reduction in primal and dual variables $\lambda_{ij}(t)$ and $z_{ij}(t)$ in the implementation of the broadcast ADMM is key in realizing the storage gain.

IV. Convergence Analysis of Broadcast-based Distributed ADMM

A. Convergence Analysis: Preliminary Results

In this section, we study the convergence rate of our algorithm and show that the maximum degree, the number of edges, and the singular values of the normalized communication matrix defined over the underlying graph impact the algorithm performance.

First, we show a lemma which we will use to provide a bound on the improvement of Lagrangian function (see Corollary 1).

Lemma 2. *Let $x_i = x^*$ be an optimal solution of problem* (4) *and $z_{ij}^* = A_{ij}x^*$ for all $i = 1, \ldots, n$ and $j \in N(i)$. For any $j = 1, \ldots, n$ and $\mathbf{p} \in \mathbb{R}^n$, we have*

$$\begin{aligned} & f_j(x_j(t+1)) - f_j(x^*) + \sum_{i \in N(j)} p_i(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} \frac{1}{2c}[(p_i(t+1) - p_i)^2 - (p_i(t) - p_i)^2] \\ & + \sum_{i \in N(j)} \frac{c}{2}[(z_{ij}(t+1) - z_{ij}^*)^2 - (z_{ij}(t) - z_{ij}^*)^2] \\ & + \sum_{i \in N(j)} p_i(t+1)(z_{ij}(t+1) - z_{ij}^*) \leq 0. \end{aligned} \tag{15}$$

Proof: From (6), for each j there exist a subgradient $h_j(x_j(t+1))$ in $\partial f_j(x_j(t+1))$ such that

$$\begin{aligned} & h_j(x_j(t+1)) \\ & \quad + \sum_{i \in N(j)} [\lambda_{ij}(t)A_{ij} + cA_{ij}(A_{ij}x_j(t+1) - z_{ij}(t))] = 0. \end{aligned}$$

Therefore, we have that

$$\begin{aligned} & (x_j - x_j(t+1))h_j(x_j(t+1) + (x_j - x_j(t+1)) \\ & \sum_{i \in N(j)} [\lambda_{ij}(t)A_{ij} + cA_{ij}\left(A_{ij}x_j(t+1) - z_{ij}(t)\right)] = 0, \end{aligned}$$

for any $x_j \in \mathbb{R}$. By definition of subgradient, for any $x_j \in \mathbb{R}$, we have

$$f_j(x_j) - f_j(x_j(t+1)) \geq (x_j - x_j(t+1))h_j(x_j(t+1)).$$

The above two relations yield

$$\begin{aligned} & f_j(x_j(t+1)) + (x_j(t+1) - x_j) \\ & \sum_{i \in N(j)} [\lambda_{ij}(t)A_{ij} + cA_{ij}\left(A_{ij}x_j(t+1) - z_{ij}(t)\right)] \leq f_j(x_j), \end{aligned}$$

for any $x_j \in \mathbb{R}$. By letting $x_j = x^*$ and using (8), we obtain

$$\begin{aligned} & f_j(x_j(t+1)) - f_j(x^*) \leq \\ & \sum_{i \in N(j)} \left(A_{ij}(x_j(t+1) - x^*)\right)\left(\lambda_{ij}(t+1) + c(z_{ij}(t+1) - z_{ij}(t)\right). \end{aligned} \tag{16}$$

Using (9), we have that

$$-(\lambda_{ij}(t) + c(A_{ij}x_j(t+1) - z_{ij}(t+1) + p_i(t+1))) = 0.$$

Multiplying by $(z_{ij}(t+1) - z_{ij}^*)$ and taking the summation over all $i \in N(j)$, we have

$$\begin{aligned} & \sum_{i \in N(j)} -(z_{ij}(t+1) - z_{ij}^*) \\ & (\lambda_{ij}(t) + c(A_{ij}x_j(t+1) - z_{ij}(t+1) - p_i(t+1))) \leq 0. \end{aligned}$$

Using (8), we obtain

$$\sum_{i \in N(j)} \left(-\lambda_{ij}(t+1) + p_i(t+1))\right)(z_{ij}(t+1) - z_{ij}^*) \leq 0. \tag{17}$$

Taking the summation of (16) and (17), we have

$$\begin{aligned} & f_j(x_j(t+1)) + \sum_{i \in N(j)} \lambda_{ij}(t+1)(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} c(z_{ij}(t+1) - z_{ij}(t))(z_{ij}(t+1) - z_{ij}^*) \\ & + \sum_{i \in N(j)} c(z_{ij}(t+1) - z_{ij}(t))(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} p_i(t+1)(z_{ij}(t+1) - z_{ij}^*) \leq f_j(x^*). \end{aligned}$$

We add and subtract the term

$$\sum_{i \in N(j)} p_i(A_{ij}x_j(t+1) - z_{ij}(t+1))$$

for some arbitrary $\mathbf{p} \in \mathbb{R}^n$ and substitute $\lambda_{ij}(t+1) = p_i(t+1)$ to obtain

$$\begin{aligned} & f_j(x_j(t+1)) - f_j(x^*) + \sum_{i \in N(j)} p_i(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} (p_i(t+1) - p_i)(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} c(z_{ij}(t+1) - z_{ij}(t))(z_{ij}(t+1) - z_{ij}^*) \\ & + \sum_{i \in N(j)} c(z_{ij}(t+1) - z_{ij}(t))(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & + \sum_{i \in N(j)} p_i(t+1)(z_{ij}(t+1) - z_{ij}^*) \leq 0. \end{aligned} \tag{18}$$

The second line of (18) can be written as

$$\begin{aligned} & \sum_{i \in N(j)} (p_i(t+1) - p_i)(A_{ij}x_j(t+1) - z_{ij}(t+1)) \\ & = \sum_{i \in N(j)} \frac{1}{2c}[(p_i(t+1) - p_i)^2 \\ & - (p_i(t) - p_i)^2 + (A_{ij}x_j(t+1) - z_i(t+1))^2]. \end{aligned} \tag{19}$$

The third line of (18) can be written as

$$\begin{aligned} & \sum_{i \in N(j)} c(z_{ij}(t+1) - z_{ij}(t))(z_{ij}(t+1) - z_{ij}^*) \\ & = \sum_{i \in N(j)} \frac{c}{2}[(z_{ij}(t+1) - z_{ij}^*)^2 \\ & - (z_{ij}(t) - z_{ij}^*)^2 + (z_{ij}(t+1) - z_{ij}(t))^2]. \end{aligned} \tag{20}$$

The fourth line of (18) can be written as

$$\begin{aligned}&\sum_{i\in N(j)} c(z_{ij}(t+1)-z_{ij}(t))(A_{ij}x_j(t+1)-z_{ij}(t+1))\\&=\sum_{i\in N(j)}\frac{c}{2}[(A_{ij}x_j(t+1)-z_{ij}(t))^2\\&-(A_{ij}x_j(t+1)-z_{ij}(t+1))^2-(z_{ij}(t+1)-z_{ij}(t))^2].\end{aligned}\tag{21}$$

Substituting (19), (20), and (21) in (18) and ignoring the positive term $\sum_{i\in N(j)}\frac{c}{2}(A_{ij}x_j(t+1)-z_{ij}(t))^2$, we obtain

$$\begin{aligned}&f_j(x_j(t+1))-f_j(x^*)+\sum_{i\in N(j)} p_i(A_{ij}x_j(t+1)-z_{ij}(t+1))\\&+\sum_{i\in N(j)}\frac{1}{2c}[(p_i(t+1)-p_i)^2-(p_i(t)-p_i)^2]\\&+\sum_{i\in N(j)}\frac{c}{2}[(z_{ij}(t+1)-z^*_{ij})^2-(z_{ij}(t)-z^*_{ij})^2]\\&+\sum_{i\in N(j)} p_i(t+1)(z_{ij}(t+1)-z^*_{ij})\le 0.\end{aligned}\tag{22}$$

■

Let $\mathbf{x}(t)=(x_1(t),\ldots,x_n(t))$, $\mathbf{x}^*=(x^*,\ldots,x^*)\in\mathbb{R}^n$, and

$$F(\mathbf{x})=\sum_{j=1}^n f_j(x_j).$$

In the following corollary, we will provide a bound on the difference of Lagrange function at primal $\mathbf{x}(t+1)$ and at the optimum $\mathbf{x}^*$ for an arbitrary dual vector $\mathbf{p}$, i.e., $L(\mathbf{x}(t+1),\mathbf{p})-L(\mathbf{x}^*,\mathbf{p})$. Note that since $A\mathbf{x}^*=0$, we have that $L(\mathbf{x}^*,\mathbf{p})=F(\mathbf{x}^*)$.

Corollary 1. *Let $x_i=x^*$ be an optimal solution of problem* (4) *and $z^*_{ij}=A_{ij}x^*$ for all $i=1,\ldots,n$ and $j\in N(i)$. For any $\mathbf{p}\in\mathbb{R}^n$, we have*

$$\begin{aligned}&F(\mathbf{x}(t+1))-F(\mathbf{x}^*)+\mathbf{p}'A\mathbf{x}(t+1)\\&+\sum_{i=1}^n(d_i+1)\frac{1}{2c}[(p_i(t+1)-p_i)^2-(p_i(t)-p_i)^2]\\&+\sum_{j=1}^n\sum_{i\in N(j)}\frac{c}{2}[(z_{ij}(t+1)-z^*_{ij})^2-(z_{ij}(t)-z^*_{ij})^2]\le 0.\end{aligned}\tag{23}$$

Proof: Using Lemma 2 and taking the summation of (15), we obtain

$$\begin{aligned}&F(\mathbf{x}(t+1))-F(\mathbf{x}^*)\\&+\sum_{j=1}^n\sum_{i\in N(j)} p_i(A_{ij}x_j(t+1)-z_{ij}(t+1))\\&+\sum_{j=1}^n\sum_{i\in N(j)}\frac{1}{2c}[(p_i(t+1)-p_i)^2-(p_i(t)-p_i)^2]\\&+\sum_{j=1}^n\sum_{i\in N(j)}\frac{c}{2}[(z_{ij}(t+1)-z^*_{ij})^2-(z_{ij}(t)-z^*_{ij})^2]\\&+\sum_{j=1}^n\sum_{i\in N(j)} p_i(t+1)(z_{ij}(t+1)-z^*_{ij})\le 0.\end{aligned}\tag{24}$$

We have that

$$\sum_{j=1}^n\sum_{i\in N(j)} p_iA_{ij}x_j(t+1)=\mathbf{p}'A\mathbf{x}(t+1).$$

We can also write

$$\begin{aligned}&\sum_{j=1}^n\sum_{i\in N(j)} p_i(t+1)z_{ij}(t+1)\\&=\sum_{i=1}^n p_i(t+1)\left(\sum_{j\in N(i)} z_{ij}(t+1)\right)=0,\end{aligned}$$

where we used the constraint $\sum_{j\in N(i)} z_{ij}(t+1)=0$, for any $i=1,\ldots,n$. Similarly, we have that $\sum_{j=1}^n\sum_{i\in N(j)} p_iz_{ij}(t+1)=0$ and $\sum_{j=1}^n\sum_{i\in N(j)} p_iz^*_{ij}=0$. Using the previous four relations in (24) completes the proof. ■

We consider the performance of the algorithm at the ergodic vector defined as

$$\hat{\mathbf{x}}(T)=(\hat{x}_1(T),\ldots,\hat{x}_n(T)),$$

where

$$\hat{x}_i(T)=\frac{1}{T}\sum_{t=1}^T x_i(t),$$

for any $i=1,\ldots,n$. Note that each agent i can construct this vector by simple recursive time-averaging of its estimate $x_i(t)$.

Lemma 3. *Let $x_i=x^*$ be an optimal solution of problem* (4) *and $z^*_{ij}=A_{ij}x^*$ for all $i=1,\ldots,n$ and $j\in N(i)$. For any T and $\mathbf{p}\in\mathbb{R}^n$, we have that*

$$\begin{aligned}&F(\hat{\mathbf{x}}(T))-F(\mathbf{x}^*)+\mathbf{p}'A\hat{\mathbf{x}}(T)\\&\le\sum_{i=1}^n(d_i+1)\frac{1}{2cT}(p_i(0)-p_i)^2\\&+\sum_{j=1}^n\sum_{i\in N(j)}\frac{c}{2T}(z_{ij}(0)-z^*_{ij})^2.\end{aligned}$$

Proof: Using Corollary 1 and taking the summation of the telescopic equation given in (23) for $t=0$ to T and then dividing by $T+1$, we obtain

$$\sum_{i=1}^{n}\left(\frac{1}{T+1}\sum_{t=0}^{T}f_i(x_i(t+1))\right)-\sum_{i=1}^{n}f_i(x^*)$$
$$+\frac{1}{T+1}\sum_{t=0}^{T+1}\mathbf{p}'A\mathbf{x}(t+1)$$
$$+\sum_{i=1}^{n}(d_i+1)\frac{1}{2c(T+1)}[(p_i(T+1)-p_i)^2-(p_i(0)-p_i)^2]$$
$$+\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2(T+1)}[(z_{ij}(T+1)-z_{ij}^*)^2-(z_{ij}(0)-z_{ij}^*)^2]$$
$$\leq 0.$$

Next, we use the convexity of the functions f_i's for $i=1,\ldots,n$ to further lower bound the left-hand side of the previous equation by

$$f_i(\hat{x}_i(T+1))\leq\frac{1}{T+1}\sum_{t=0}^{T}f_i(x_i(t+1)).$$

We obtain

$$\sum_{i=1}^{n}f_i(\hat{x}_i(T+1))-\sum_{i=1}^{n}f_i(x^*)+\mathbf{p}'A\hat{\mathbf{x}}(T+1) \tag{25}$$
$$+\sum_{i=1}^{n}(d_i+1)\frac{1}{2c(T+1)}[(p_i(T+1)-p_i)^2-(p_i(0)-p_i)^2]$$
$$+\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2(T+1)}[(z_{ij}(T+1)-z_{ij}^*)^2-(z_{ij}(0)-z_{ij}^*)^2]$$
$$\leq 0, \tag{26}$$

where we used the linearity of $\mathbf{p}'A\mathbf{x}(t+1)$ to write $\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{p}'A\mathbf{x}(t+1)=\mathbf{p}'A\hat{\mathbf{x}}(T+1)$. Rewriting (25) and ignoring the positive terms $\sum_{i=1}^{n}(d_i+1)\frac{1}{2c(T+1)}[(p_i(T+1)-p_i)^2]$ and $\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2(T+1)}[(z_{ij}(T+1)-z_{ij}^*)^2]$, we yield

$$F(\hat{\mathbf{x}}(T+1))-F(\mathbf{x}^*)+\mathbf{p}'A\hat{\mathbf{x}}(T+1)$$
$$\leq\sum_{i=1}^{n}(d_i+1)\frac{1}{2c(T+1)}(p_i(0)-p_i)^2$$
$$+\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2(T+1)}(z_{ij}(0)-z_{ij}^*)^2.$$

Writing this equation for T completes the proof. ■

B. Convergence Analysis: Optimality Convergence

In the next theorem, we show that the difference of the objective function values at the ergodic vector from the optimal value converges to 0 with rate $O(\frac{1}{T})$.[4]

Theorem 1. *Let $(\mathbf{x}^*,\mathbf{p}^*)$ be a primal-dual optimal solution of* (2). *Let $\{\hat{\mathbf{x}}(T)\}$ be the ergodic sequence generated by the broadcast-based distributed ADMM algorithm. Then,*

$$|F(\hat{\mathbf{x}}(T))-F(\mathbf{x}^*)| \tag{27}$$
$$\leq\frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)}((A_{ij}x_j(0)-y_i(0))-A_{ij}x^*)^2$$
$$+\frac{1}{2cT}\max\{\sum_{i=1}^{n}(d_i+1)(p_i(0)-2p_i^*)^2,\sum_{i=1}^{n}(d_i+1)p_i(0)^2\}, \tag{28}$$

where $x_i(0)$, $p_i(0)$, and $y_i(0)$ for $i=1,\ldots,n$ are the initial values of Algorithm 1.

Proof: Using Lemma 3 with $\mathbf{p}=0$, we obtain

$$F(\hat{\mathbf{x}}(T))-F(\mathbf{x}^*)\leq\sum_{i=1}^{n}(d_i+1)\frac{1}{2cT}p_i(0)^2$$
$$+\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2T}(z_{ij}(0)-z_{ij}^*)^2. \tag{29}$$

Since $(\mathbf{x}^*,\mathbf{p}^*)$ is a primal-dual optimal solution, using Note 1 and (3), we have that

$$F(\mathbf{x}^*)\leq F(\hat{\mathbf{x}}(T))+\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T), \tag{30}$$

which implies

$$F(\mathbf{x}^*)-F(\hat{\mathbf{x}}(T))\leq\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T). \tag{31}$$

Next, we will bound the term $\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T)$. We add the term $\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T)$ to both sides of (30) to obtain

$$\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T)\leq F(\hat{\mathbf{x}}(T))-F(\mathbf{x}^*)+2\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T). \tag{32}$$

Again, using Lemma 3 to bound the right-hand side of (32), we obtain

$$\mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T)\leq\sum_{i=1}^{n}(d_i+1)\frac{1}{2cT}[(p_i(0)-2p_i^*)^2]$$
$$+\sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2T}(z_{ij}(0)-z_{ij}^*)^2. \tag{33}$$

Using (33) to bound the right-hand side of (31), we obtain

$$F(\mathbf{x}^*)-F(\hat{\mathbf{x}}(T))\leq\sum_{i=1}^{n}(d_i+1)\frac{1}{2cT}(p_i(0)-2p_i^*)^2$$
$$+\frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)}(z_{ij}(0)-z_{ij}^*)^2. \tag{34}$$

Combining (29) and (34), we obtain

$$|F(\hat{\mathbf{x}}(T))-F(\mathbf{x}^*)|\leq\frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)}(z_{ij}(0)-z_{ij}^*)^2$$
$$+\frac{1}{2cT}\max\{\sum_{i=1}^{n}(d_i+1)(p_i(0)-2p_i^*)^2,\sum_{i=1}^{n}(d_i+1)p_i(0)^2\}.$$

Substituting $z_{ij}(0)=A_{ij}x_j(0)-y_i(0)$ (see (14)) and $z_{ij}^*=A_{ij}x^*$ in the previous relation, completes the proof. ■

[4]We say that $f(T)=O(g(T))$ if there exist a constant C and T_0 such that $f(T)\leq Cg(T)$ for all $T\geq T_0$.

C. Convergence Analysis: Feasibility Convergence

In the next theorem, we will show that the feasibility violation converges to zero with rate $O(\frac{1}{T})$.

Theorem 2. *Let $(\mathbf{x}^*, \mathbf{p}^*)$ be a primal-dual optimal solution of (2). Let $\{\hat{\mathbf{x}}(T)\}$ be the ergodic vector generated by the broadcast-based distributed ADMM algorithm. Then,*

$$||A\hat{\mathbf{x}}(T)|| \leq \frac{1}{cT}\sum_{i=1}^{n}(d_i+1)\left(p_i(0)-p_i^*\right)^2 + \frac{1}{cT}(d_{max}+1) + \frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)}\left((A_{ij}x_j(0)-y_i(0))-A_{ij}x^*\right)^2, \quad (35)$$

Proof: Using Lemma 3 with $\mathbf{p} = \mathbf{p}^* + \frac{A\hat{\mathbf{x}}(T)}{||A\hat{\mathbf{x}}(T)||}$, we have

$$\begin{aligned} &F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + \mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T) + ||A\hat{\mathbf{x}}(T)|| \\ &\leq \sum_{i=1}^{n}(d_i+1)\frac{1}{2cT}\left(p_i(0)-p_i^* - \frac{(A\hat{\mathbf{x}}(T))_i}{||A\hat{\mathbf{x}}(T)||}\right)^2 \\ &+ \sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2T}(z_{ij}(0)-z_{ij}^*)^2. \end{aligned}$$

Since $(\mathbf{x}^*, \mathbf{p}^*)$ is a primal-dual optimal solution, using Note 1 and (3), we have that

$$F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + \mathbf{p}^{*\prime}A\hat{\mathbf{x}}(T) \geq 0.$$

Combining the two previous relations, we obtain

$$||A\hat{\mathbf{x}}(T)|| \leq \sum_{i=1}^{n}(d_i+1)\frac{1}{2cT}\left(p_i(0)-p_i^* - \frac{(A\hat{\mathbf{x}}(T))_i}{||A\hat{\mathbf{x}}(T)||}\right)^2 + \sum_{j=1}^{n}\sum_{i\in N(j)}\frac{c}{2T}(z_{ij}(0)-z_{ij}^*)^2.$$

Since $(a-b)^2 \leq 2a^2+2b^2$, for any $a, b \in \mathbb{R}$, we have that

$$\begin{aligned} &\left(p_i(0)-p_i^* - \frac{(A\hat{\mathbf{x}}(T))_i}{||A\hat{\mathbf{x}}(T)||}\right)^2 \\ &\leq 2(p_i(0)-p_i^*)^2 + 2\left(\frac{(A\hat{\mathbf{x}}(T))_i}{||A\hat{\mathbf{x}}(T)||}\right)^2. \end{aligned}$$

Using this inequality and $\sum_{i=1}^{n}\left(\frac{(A\hat{\mathbf{x}}(T))_i}{||A\hat{\mathbf{x}}(T)||}\right)^2 = 1$, we can further bound $||A\hat{\mathbf{x}}(T)||$ as

$$\begin{aligned} ||A\hat{\mathbf{x}}(T)|| &\leq \frac{1}{cT}\sum_{i=1}^{n}(d_i+1)\left(p_i(0)-p_i^*\right)^2 \\ &+ \frac{1}{cT}(d_{\max}+1) + \frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)}(z_{ij}(0)-z_{ij}^*)^2. \end{aligned}$$

Substituting $z_{ij}(0) = A_{ij}x_j(0) - y_i(0)$ (see (14)) and $z_{ij}^* = A_{ij}x^*$ in the previous relation completes the proof. ■

V. Network Effect and Choice of Communication Matrix

A. Network Effect

In this section, we will analyze the effect of network structure on the convergence rate of the broadcast-based distributed ADMM algorithm.

Lemma 4. *Let $\mathbf{x}^*$ be an optimal solution for problem (2). There exists an optimal dual solution $\tilde{\mathbf{p}}$ for problem (2) that satisfies*

$$\sum_{i=1}^{n}(d_i+1)\tilde{p}_i^2 \leq (d_{max}+1)\frac{L^2}{{\sigma_{min}}^2},$$

where L is a bound on the subgradients of the function F at $\mathbf{x}^*$, i.e., $\|\mathbf{v}\| \leq L$ for all $\mathbf{v} \in \partial F(\mathbf{x}^*)$, and σ_{min} is the smallest non-zero singular value of A. [5]

Proof: By Assumption 1 and equation (3), there exists an optimal primal-dual solution for problem (2) such that $(\mathbf{x}^*, \mathbf{p}^*)$ is a saddle point of the Lagrangian function, i.e., for any $\mathbf{x} \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} f_i(x^*) - \sum_{i=1}^{n} f_i(x_i) \leq \mathbf{p}^{*\prime}A(\mathbf{x}-\mathbf{x}^*). \quad (36)$$

Note that since $A\mathbf{x}^* = 0$, the inequality $L(\mathbf{x}^*, \mathbf{p}) \leq L(\mathbf{x}^*, \mathbf{p}^*)$ holds for any choice of $\mathbf{p}$ and $\mathbf{p}^*$ and $(\mathbf{x}^*, \mathbf{p}^*)$ satisfies saddle point inequality given in (3) if and only if it satisfies the inequality given in (36). Equation (36) shows that $\mathbf{p}^{*\prime}A \in \partial F(\mathbf{x}^*)$. Let $\mathbf{p}^{*\prime}A = \mathbf{v}' \in \partial F(\mathbf{x}^*)$. We will use this $\mathbf{p}^*$ to construct $\tilde{\mathbf{p}}$ such that $\tilde{\mathbf{p}}'A = \mathbf{v}'$ and hence we would have

$$\sum_{i=1}^{n} f_i(x^*) - \sum_{i=1}^{n} f_i(x_i) \leq \tilde{\mathbf{p}}'A(\mathbf{x}-\mathbf{x}^*),$$

meaning $(\mathbf{x}^*, \tilde{\mathbf{p}})$ satisfies the saddle point inequality given in (3). This shows that $(\mathbf{x}^*, \tilde{\mathbf{p}})$ is an optimal primal-dual solution (see section 6 of [2]). Moreover, we choose $\tilde{\mathbf{p}}$ to satisfy the constraint $\sum_{i=1}^{n}(d_i+1)\tilde{p}_i^2 \leq (d_{\max}+1)\frac{L^2}{{\sigma_{\min}}^2}$ as desired in the statement of lemma.

Let $A = \sum_{i=1}^{r}\mathbf{u}_i\sigma_i\mathbf{v}_i'$ be the singular value decomposition of A, where rank$(A) = r$. Since $\mathbf{p}^{*\prime}A = \mathbf{v}'$, $\mathbf{v}$ belongs to the span of $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ and can be written as $\mathbf{v} = \sum_{i=1}^{r} c_i\mathbf{v}_i$ for some coefficients c_i's . Let $\tilde{\mathbf{p}} = \sum_{i=1}^{r}\frac{c_i}{\sigma_i}\mathbf{u}_i$. By this choice of $\tilde{\mathbf{p}}$ we have $\tilde{\mathbf{p}}'A = \sum_{i=1}^{r} c_i\mathbf{v}_i = \mathbf{v}'$. This choice also yields

$$\begin{aligned} &\sum_{i=1}^{n}(d_i+1)\tilde{p}_i^2 \leq (d_{\max}+1)||\tilde{\mathbf{p}}||^2 = (d_{\max}+1)\sum_{i=1}^{r}\frac{c_i^2}{\sigma_i^2} \\ &\leq (d_{\max}+1)\sum_{i=1}^{r}\frac{c_i^2}{\sigma_{\min}^2} = (d_{\max}+1)\frac{1}{\sigma_{\min}^2}\sum_{i=1}^{r}c_i^2 \\ &= (d_{\max}+1)\frac{1}{\sigma_{\min}^2}||\mathbf{v}||^2 \leq (d_{\max}+1)\frac{1}{\sigma_{\min}^2}L^2, \end{aligned}$$

where we used the bound on the subgradient to obtain the last inequality. Since $\tilde{\mathbf{p}}'A = \mathbf{v}' \in \partial F(\mathbf{x}^*)$, $(\mathbf{x}^*, \tilde{\mathbf{p}})$ satisfies the saddle point inequality given in (3) and hence $\tilde{\mathbf{p}}$ is a dual optimal solution.

■

Next, we use Lemma 4 in Theorems 1 and 2 to analyze the network effect.

[5] The set of subgradients of F at the point x^* is called the subdifferential of F at x^*, and is denoted by $\partial F(x^*)$.

Theorem 3. *Let $\sigma_{min}(A)$ be the smallest non-zero singular value of A. Also let the initialization in Algorithm 1 be as $\mathbf{x}(0) = 0$, $\mathbf{p}(0) = 0$, and $\mathbf{y}(0) = 0$. We have*

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}||A||_F^2\ x^{*2} + (d_{max}+1)\frac{2}{cT}\frac{L^2}{\sigma_{min}(A)^2}, \tag{37}$$

and

$$||A\hat{\mathbf{x}}(T)|| \leq \frac{c}{2T}||A||_F^2\ x^{*2} + \frac{1}{cT}(d_{max}+1) + (d_{max}+1)\frac{1}{cT}\frac{L^2}{\sigma_{min}(A)^2},$$

where $||A||_F = \sum_{i,j} A_{ij}^2$ denotes Frobeniuos norm of A.

Proof: Let $(\mathbf{x}^*, \tilde{\mathbf{p}})$ be a primal-dual optimal solution characterized in Lemma 4. Using Theorem 1 with $\mathbf{x}(0) = 0$, $\mathbf{p}(0) = 0$, and $\mathbf{y}(0) = 0$, we have that

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)} {A_{ij}}^2 x^{*2} + \frac{1}{2cT}\sum_{i=1}^{n}(d_i+1)4\tilde{p}_i^2.$$

Using Lemma 4, we obtain

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}||A||_F^2\ x^{*2} + (d_{\max}+1)\frac{2}{cT}\frac{L^2}{\sigma_{\min}(A)^2}.$$

Using Theorem 2 with $\mathbf{x}(0) = 0$, $\mathbf{p}(0) = 0$, and $\mathbf{y}(0) = 0$, we have that

$$||A\hat{\mathbf{x}}(T)|| \leq \frac{1}{cT}\sum_{i=1}^{n}(d_i+1)\tilde{p}_i^2 + \frac{1}{cT}(d_{\max}+1) + \frac{c}{2T}\sum_{j=1}^{n}\sum_{i\in N(j)} {A_{ij}}^2 x^{*2}.$$

Using Lemma 4, we obtain

$$||A\hat{\mathbf{x}}(T)|| \leq \frac{c}{2T}||A||_F^2\ x^{*2} + \frac{1}{cT}(d_{\max}+1) + (d_{\max}+1)\frac{1}{cT}\frac{L^2}{\sigma_{\min}(A)^2},$$

which completes the proof. ■

B. Choice of Communication Matrix

The performance bounds we presented hold for any communication matrix, i.e., any matrix that satisfies assumption of Definition 1. We next consider the problem of choosing the communication matrix that yields the best upper bound on the algorithm performance. We first analyze performance when we pick A to be the standard Laplacian of the underlying graph.

Algorithm 2 Broadcast-Based Distributed ADMM: Laplacian communication matrix

---

- **Initialization:** $x_i(0)$, $y_i(0)$, and $p_i(0)$, for any $i \in V$.
- **Algorithm:**
  1) For $i = 1, \ldots, n$, let
  $$x_i(t+1) \in \text{argmin}_{x_i\in\mathbb{R}} f_i(x_i) + \left(\sum_{j\in N(i)\setminus\{i\}} -p_j(t)x_i + \frac{c}{2}(y_j(t) - (x_i - x_i(t))^2\right) + d_i p_i(t) x_i + \frac{c}{2}(y_i(t) + d_i(x_i - x_i(t))^2$$
  2) For $i = 1, \ldots, n$, let
  $$y_i(t+1) = \frac{1}{d_i+1}\left(d_i x_i(t+1) + \sum_{j\in N(i)\setminus\{i\}} -x_j(t+1)\right).$$
  3) For $i = 1, \ldots, n$, let $p_i(t+1) = p_i(t) + cy_i(t+1)$.
- **Output:** $\{x_i(t)\}_{t=0}^{\infty}$ for any $i \in V$.

---

1) **Standard Laplacian Communication Matrix***:* Let $A_{ij} = -1$ and $A_{ii} = d_i$ for all $i = 1, \ldots, n$ and $j \in N(i)\setminus\{i\}$. This would lead to choosing Laplacian of the underlying graph as the communication matrix. By this choice of communication matrix, each node only requires to know its degree in order to implement the algorithm, as described in Algorithm 2.

Since the communication matrix A is the Laplacian of the underlying graph, $\sigma_{\min}(A)^2$ is the algebraic connectivity of graph denoted by $a(G)$. Moreover, we have that

$$||A||_F^2 = \sum_{i=1}^{n}\sum_{j\in N(i)} A_{ij}^2 = \sum_{i=1}^{n} d_i(d_i+1).$$

Using the previous two relations in Theorem 3, we obtain

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}x^{*2}\sum_{i=1}^{n} d_i(d_i+1) + (d_{\max}+1)\frac{2}{cT}\frac{L^2}{a(G)},$$

and

$$||A\hat{\mathbf{x}}(T)|| \leq \frac{c}{2T}x^{*2}\sum_{i=1}^{n} d_i(d_i+1) + \frac{1}{cT}(d_{\max}+1) + (d_{\max}+1)\frac{1}{cT}\frac{L^2}{a(G)}.$$

Thus our bounds show how degrees of nodes and algebraic connectivity of the graph impact the algorithm performance.

2) **Optimal Choice of Communication Matrix***:* We can choose matrix A to minimize the upper bounds on the optimality and feasibility violations. Both the feasibility and optimality violation bounds have the following term that depends on the choice of matrix A:

$$\frac{c}{2T}{x^*}^2||A||_F^2 + (d_{\max}+1)\frac{1}{cT}\frac{L^2}{\sigma_{\min}(A)^2}. \tag{38}$$

Here, we focus on the class of *symmetric generalized Laplacian matrices* in the form of $A = I - P$ for some symmetric stochastic matrix P.[6] This yields the following bound on the first term of (38) as follows

$$\frac{c}{2T}{x^*}^2||A||_F^2 = \frac{c}{2T}{x^*}^2||(I-P)||_F^2 \le \frac{c}{2T}{x^*}^2 2n.$$

Now we will find the best choice of A (or P) that minimizes the second term of (38). Note that since A is symmetric, $\sigma_{\min}(A)^2 = \lambda_{\min}(A)$, where $\lambda_{\min}(A)$ denotes the smallest non-zero eigenvalue of A. Moreover, λ is an eigenvalue of A if and only if $\mu = 1 - \lambda$ is an eigenvalue of P. Therefore, the problem of finding A that maximizes $\sigma_{\min}(A)^2$ is the same as the problem of finding P that minimizes $\mu_2(P)$ (second largest eigenvalue of P). This problem can be formulated as

$$\begin{aligned} \min \quad & \mu_2(P) \\ \text{s.t.} \quad & P\mathbf{1} = 0, \quad P = P' \\ & P_{ij} = 0, \quad (i,j) \notin E \\ & P \ge 0. \end{aligned} \tag{39}$$

It is shown in [4] that (39) is a convex optimization problem. Let μ^* be the optimal solution of (39). It follows that $\sigma_{\min}(A)^2 = 1 - \mu^*$. We let the initialization in Algorithm 1 be $\mathbf{p}(0) = 0$, $x_i(0)$, and $y_i(0) = 0$ for any i. Using Theorem 3, we obtain

$$\begin{aligned} |F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \le & \frac{c}{2T}{x^*}^2 2n + \frac{1}{cT}(d_{\max}+1) \\ & + (d_{\max}+1)\frac{1}{cT}\frac{L^2}{1-\mu^*}, \end{aligned}$$

and

$$\begin{aligned} ||A\hat{\mathbf{x}}(T)|| \le & \frac{c}{2T}{x^*}^2 2n + \frac{1}{cT}(d_{\max}+1) \\ & + (d_{\max}+1)\frac{1}{cT}\frac{L^2}{1-\mu^*}. \end{aligned}$$

Note 2. Since (39) is a convex optimization, it can be solved efficiently to obtain the communication matrix with maximum $\sigma_{\min}(A)^2$. Also note that solving (39) can be done offline through a distributed algorithm that uses pairwise communication according to gossip algorithm [5]. This could be done before our algorithm is implemented.

VI. Numerical Results

In this section, we show some simulations of the broadcast-based ADMM algorithm to demonstrate the performance of the algorithm and highlight the dependence on the network structure. We consider minimizing the function $F(x) = \frac{1}{2}\sum_{i=1}^{n}(x - a_i)^2$ where a_i is a scalar that is known only to node i. This problem appears in distributed estimation where the goal is to estimate the parameter x^*, using local measurements $a_i = x^* + N_i$ at each node $i = 1, \ldots, n$. Here N_i represents measurements noise, which we assume to be jointly Gaussian with mean zero and variance one. The maximum likelihood estimate is the minimizer x^* of $F(x)$.

[6] If A is communication matrix with $|A_{ij}| \le 1$ for all i, j, then A can be written as $A = I - P$, where P is a stochastic matrix. Since sum of each row of A is zero, sum of each row of P is one and $P_{ij} \ge 0$ for all i and j.

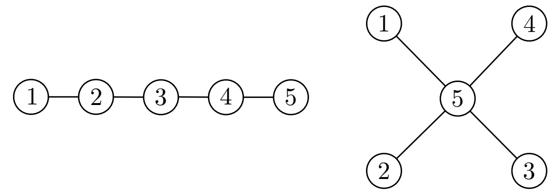

Fig. 2: Path and Star Networks used to illustrate the effect of network structure.

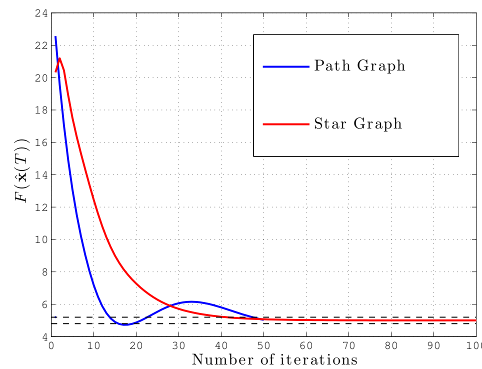

Fig. 3: Effect of network topology on optimality convergence: Star network with $a(G) = 1$ and path network with $a(G) = 0.38$ ($F(x^*) = 5$).

A. Effect of Network Topology

We consider a path network consisting of five nodes and a star network consisting of five nodes shown in Figure 2. We assume $a_i = i$ for $i = 1, \ldots, 5$ and apply broadcast-based ADMM on these networks. For this example we have that $F(x^*) = 5$ and $x^* = 3$. We initialize the algorithms with $x_i(0) = y_i(0) = p_i(0) = 0$ for $i = 1, \ldots, 5$. We also take the communication matrix to be Laplacian of the graph. The performance of the algorithm on two graphs are shown in Figures 3 and 4. Note that the algebraic connectivity of star graph and path graph are 1 and .38, respectively. Therefore, the performance bounds that we derived suggest the algorithm converges faster on the star graph. This is also illustrated by the simulation results in Figures 3 and 4. In Figure 3, in order to reach $\pm.2$ error, we need 40 iterations on the star graph and 50 iterations on the path graph. In Figure 4, in order to reach $\pm.1$ error, we need 10 iterations on the star graph and 40 iterations on the path graph

B. Effect of Communication Matrix

In Figure 6, we plot the performance of our algorithm over a sample network shown in Figure 5 with Laplacian matrix and optimal matrix A^* found by solving (39) as follows:

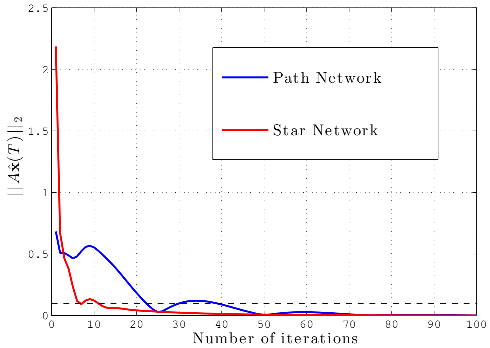

Fig. 4: Effect of network topology on feasibility convergence: Star network with $a(G) = 1$ and path network with $a(G) = 0.38$.

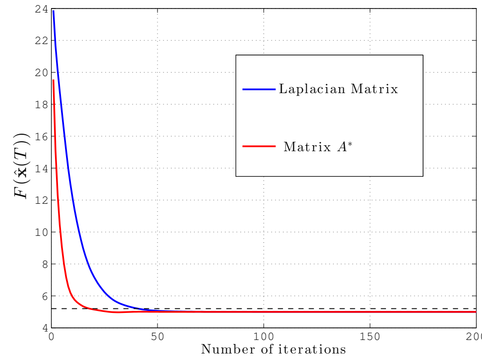

Fig. 6: Effect of communication matrix: performance with Laplacian and matrix A^* ($F(x^*) = 5$).

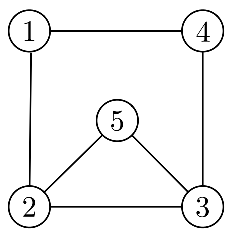

Fig. 5: Network used to illustrate the effect of communication matrix.

$$\text{Laplacian matrix} = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & -1 & -1 & 0 & 2 \end{bmatrix}$$

$$A^* = \begin{bmatrix} 0.68 & -0.29 & 0 & -0.39 & 0 \\ -0.29 & 0.81 & -0.21 & 0 & -0.31 \\ 0 & -0.21 & 0.81 & -0.29 & -0.31 \\ -0.39 & 0 & -0.29 & 0.68 & 0 \\ 0 & -0.31 & -0.31 & 0 & 0.62 \end{bmatrix}.$$

Note that we found A^* by solving convex problem (39) with the optimal solution P^*, and letting $A^* = I - P^*$. As Figure 6 shows by choosing the optimal communication, we obtain a faster convergence rate. In Figure 6, in order to reach $\pm.2$ error, we need 40 iterations using Laplacian matrix and 15 iterations using matrix A^*.

VII. Conclusion

We considered a multi agent optimization problem where a network of agents collectively solves a global optimization problem with the objective function given by the sum of locally known convex functions. In Algorithm 1, we proposed a fully distributed broadcast-based Alternating Direction Method of Multipliers (ADMM). In Theorems 1 and 2, we showed that both the objective function values and the feasibility violation converge with rate $O(\frac{1}{T})$, where T is the number of iterations . We also characterized the effect of network structure and the choice of communication matrix on the convergence speed in Theorem 3. Because of its broadcast nature, the storage requirements of our algorithm are much more modest compared to the distributed algorithms that use pairwise communication between agents. Moreover, we demonstrated the performance of our algorithm for networks with different structures and different communication matrices to show the dependence of convergence rate on both network structure and the choice of communication matrix.

References

[1] R. Agaev and P. Chebotarev. On the spectra of nonsymmetric Laplacian matrices. *Linear Algebra and its Applications*, 2005.

[2] D. P. Bertsekas, A. Nedić, and A. Ozdaglar. *Convex analysis and optimization*. Athena Scientific Belmont, 2003.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.

[4] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 2004.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *Information Theory, IEEE Transactions on*, 2006.

[6] 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*, 3(1):1–122, 2011.

[7] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *Signal Processing, IEEE Transactions on*, 2012.

[8] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, DTIC Document, 2012.

[9] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. *Automatic Control, IEEE Transactions on*, 2012.

[10] J. Eckstein. Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 2012.

[11] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista. Fast consensus by the alternating direction multipliers method. *Signal Processing, IEEE Transactions on*, 2011.

[12] B. He and X. Yuan. On the O(1/n) convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 2012.

[13] M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.

[14] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 2003.

[15] D. Jakovetic, J. Xavier, and J. Moura. Fast distributed gradient methods. 2011.

[16] D. Jakovetic, J. Xavier, and J. M. Moura. Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication. *Signal Processing, IEEE Transactions on*, 2011.

[17] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008.

[18] I. Lobel and A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *Automatic Control, IEEE Transactions on*, 2011.

[19] I. Lobel, A. Ozdaglar, and D. Feijer. Distributed multi-agent optimization with state-dependent communication. *Mathematical Programming*, 2011.

[20] I. Matei and J. S. Baras. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *Selected Topics in Signal Processing, IEEE Journal of*, 2011.

[21] G. Mateos, J. A. Bazerque, and G. B. Giannakis. Distributed sparse linear regression. *Signal Processing, IEEE Transactions on*, 2010.

[22] J. F. Mota, J. Xavier, P. M. Aguiar, and M. Puschel. Basis pursuit in sensor networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011.

[23] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel. D-admm: A communication-efficient distributed algorithm for separable optimization. *Signal Processing, IEEE Transactions on*, 2013.

[24] A. Nedic. Asynchronous broadcast-based convex optimization over a network. *Automatic Control, IEEE Transactions on*, 2011.

[25] A. Nedic and A. Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *Proceedings of IEEE CDC*, 2007.

[26] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 2009.

[27] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 2010.

[28] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 2004.

[29] A. Olshevsky and J. N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 2009.

[30] S. S. Ram, A. Nedić, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 2010.

[31] I. D. Schizas, A. Ribeiro, and G. B. Giannakis. Consensus in ad hoc wsns with noisy links?part i: Distributed estimation of deterministic signals. *Signal Processing, IEEE Transactions on*, 2008.

[32] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *arXiv preprint arXiv:1404.6264*, 2014.

[33] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin. On the linear convergence of the ADMM in decentralized consensus optimization. 2013.

[34] S. Shtern, E. Wei, and A. Ozdaglar. Distributed alternating direction method of multipliers (ADMM): Performance and network effects. Working paper, 2014.

[35] K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *CDC*, 2012.

[36] J. N. Tsitsiklis. Problems in decentralized decision making and computation. Technical report, DTIC Document, 1984.

[37] J. N. Tsitsiklis, D. P. Bertsekas, M. Athans, et al. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 1986.

[38] E. Wei and A. Ozdaglar. On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers. *arXiv preprint arXiv:1307.8254*, 2013.

[39] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *CDC*, 2012.

[40] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 2007.

[41] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *arXiv preprint arXiv:1310.7063*, 2013.

[42] H. Zhu, G. B. Giannakis, and A. Cano. Distributed in-network channel decoding. *Signal Processing, IEEE Transactions on*, 2009.

[43] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *Automatic Control, IEEE Transactions on*, 2012.