

D-DistADMM: A $O(1/k)$ Distributed ADMM for Distributed Optimization in Directed Graph Topologies

Vivek Khatana¹ and Murti V. Salapaka¹

Abstract—We focus on the problem of minimizing a finite sum $f(x) = \sum_{i=1}^n f_i(x)$ of n functions f_i , where f_i are convex and available only locally to an agent i . The n agents are connected in a directed network $\mathcal{G}(\mathbf{V}, \mathbf{E})$. In this article, we present the Directed-Distributed Alternating Direction Method of Multiplier (D-DistADMM) Algorithm, which is an Alternating Direction Method of Multiplier (ADMM) based scheme and utilizes a finite-time “approximate” consensus method to solve the above optimization problem distributively. At each iteration of the proposed scheme the agents solve their local optimization problem and utilize an approximate consensus protocol to update a local estimate of the global optimization variable. We show that for convex and not-necessarily differentiable objective functions the proposed *D-DistADMM* method converges at a rate $O(1/k)$, where k is the iteration counter, in terms the difference between the Lagrangian function evaluated at any iteration k of the *D-DistADMM* algorithm and the optimal solution. We further demonstrate the features of our algorithm by solving a distributed least squares problem.

keywords: *Distributed optimization, distributed gradient descent, multi-agent networks, finite-time consensus.*

I. INTRODUCTION

This article, focuses on a set of n agents with a directed interconnection topology determined by a directed graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ and $\mathbf{E}$ are sets of vertices and directed edges respectively. The aim is to solve the following distributed optimization problem:

$$\operatorname{argmin}_{x \in \mathbb{R}^p} f(x) = \sum_{i=1}^n f_i(x), \quad (1)$$

where, $x \in \mathbb{R}^p$ is a global optimization variable, and each function $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ is a convex objective known only locally to agent i . Each agent can only communicate to any other agent that is connected to it through a directed edge in $\mathcal{G}(\mathbf{V}, \mathbf{E})$. Different problems that arise in many fields of engineering such as sensor networks [1], multi-agent coordination and control [2], machine learning [3], and robotics [4] can be reduced to a form of the problem (1). Early works on the distributed optimization problem can be found in [5], [6]. Since then, the distributed optimization problem (1) has received significant attention [7]–[9]. The methods to solve (1) can be divided into two groups: primal based and the dual methods. The first group includes the

(sub)-gradient based methods and a number of existing schemes to minimize finite sum of convex function lie in this category with beginnings in the Distributed Gradient Descent (DGD) [7], the stochastic and asynchronous versions of DGD [10], [11] and [8]. A recent paper [12] provides a comprehensive survey on the first order primal methods. The work in the present article is mostly related to the second category of dual based methods. These methods are based on the Lagrange multipliers. Particularly, Alternating Direction of Multiplier Method (ADMM) has received significant attention due to its amenable structure for distributed implementation. Researchers have proposed many variants of ADMM [13]–[18]. The reference [15] proposed a distributed ADMM algorithm to solve (1) with a $O(1/k)$ rate of convergence for convex f_i 's. The authors in [14] have established the linear convergence rate of distributed ADMM under strong convexity of the individual objective functions. However, the results in [15] and [14] are valid for undirected graph topologies which imply every communication channel is necessarily bidirectional. [19] has introduced a communication efficient ADMM scheme under restricted assumptions such as the underlying graph being bipartite or strongly convex objective functions. The reference [16] present a distributed ADMM scheme that converges at a rate of $O(1/k)$, when objective functions are convex. However, the communication weight matrix require a system-wide information which makes it unsuitable for applications where distributed synthesis is required. The work in [18] is based on minimizing a proximal first-order approximation of the smooth and strongly convex function f_i 's over an undirected network.

In this article, we propose a novel Directed-Distributed ADMM *D-DistADMM* method to solve (1) over directed networks. The *D-DistADMM* scheme utilizes a finite-time ε -consensus protocol along with an ADMM based optimization scheme, where $\varepsilon > 0$ is a pre-specified tolerance that is independent of the network topology and the problem (1). At each iteration of the *D-DistADMM* algorithm every agent i , updates its local optimization variable using an ADMM update and then compute the global parameter update using the ε -consensus protocol. Under the assumption of the individual function f_i 's being convex we prove that the proposed algorithm converges at a rate $O(1/k)$ in terms the difference between the Lagrangian function evaluated at any iteration k of the *D-DistADMM* algorithm and the optimal solution. Further, the numerical simulation results demonstrate the performance of the proposed *D-DistADMM* algorithm in solving a distributed optimization. The rest of the paper is organized as follows. In Section II, we provide

This work is supported by Advanced Research Projects Agency-Energy OPEN through the project titled “Rapidly Viable Sustained Grid” via grant no. DE-AR0001016.

¹ Vivek Khatana {khatav010@umn.edu} and Murti V. Salapaka {murtis@umn.edu} are with Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA,

the basic definitions and notation used in the article. We also present the framework to develop the *D-DistADMM* algorithm. The proposed *D-DistADMM* algorithm is discussed in Section III along with the ε -consensus protocol used with *D-DistADMM*. We present the convergence analysis for the proposed algorithm in Section IV. In Section V the results of numerical simulations utilizing *D-DistADMM* to solve a distributed least squares problem are presented. Section VI provides the conclusion.

II. DEFINITIONS, PROBLEM FORMULATION AND ASSUMPTIONS

A. Definition, Notations and Assumptions

In this section, we present definitions and notations that is used later in the analysis. Detailed description of these notions can be found in [20] and [21].

A directed graph $\mathcal{G}$ is a pair $(\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is a set of vertices (or nodes) and $\mathbf{E}$ is a set of edges, which are ordered subsets of two distinct elements of $\mathbf{V}$. If an edge from $j \in \mathbf{V}$ to $i \in \mathbf{V}$ exists then it is denoted as $(i, j) \in \mathbf{E}$. In a directed graph, a directed path from node i to j exists if there is a sequence of distinct directed edges of $\mathcal{G}$ of the form $(e_1, i), (e_2, e_1), \dots, (j, e_m)$. A directed graph is strongly connected if it has a directed path between each pair of distinct nodes i and j . The diameter of a directed graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$ is the longest shortest directed path between any two nodes in the the graph. The set of in-neighbors of node $i \in \mathbf{V}$ is called the in-neighborhood of node i and is denoted by $\mathcal{N}_i^- = \{j \mid (i, j) \in \mathbf{E}\}$ not including the node i and $|\mathcal{N}_i^-|$ denote the in-degree (number of in-neighbors) of node $i \in \mathbf{V}$. The set of out-neighbors of node $i \in \mathbf{V}$ is called the out-neighborhood of node i and is denoted by $\mathcal{N}_i^+ = \{j \mid (j, i) \in \mathbf{E}\}$ not including the node i and $|\mathcal{N}_i^+|$ denote the out-degree (number of out-neighbors) of node $i \in \mathbf{V}$. A matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$ is called a column-stochastic matrix if $0 \leq w_{ij} \leq 1$ for all $i, j \in \{1, 2, \dots, n\}$ and $\sum_{i=1}^n w_{ij} = 1$. A matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$ is said to be irreducible if for any $i, j \in \{1, \dots, n\}$, there exist $\ell \in \mathbb{N}$ such that $[\mathbf{W}^\ell]_{ij} > 0$. A non-negative matrix $\mathbf{W}$ is primitive if it is irreducible and has only one eigenvalue of maximum modulus. A function f is said to be convex if the domain of $\text{dom } f$ is convex and for all $x, y \in \text{dom } f$ and $0 \leq \alpha \leq 1$,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

In this article, each agent $i \in \{1, 2, \dots, n\}$ maintains a local estimate $x_i(k) \in \mathbb{R}^p$ and a dual variable $\lambda_i(k)$ at any iteration $k \geq 0$ of the *D-DistADMM* algorithm. Let $\mathbf{X}(k) = \{x_i(k) \mid 1 \leq i \leq n\}$ denote the collection of estimates at the k^{th} iteration of the algorithm. Define, the vectorization operation for a given collection $\mathbf{X}(k)$ as $\mathbf{x}_{[1:n]} = [x_1^T \ x_2^T \ \dots \ x_n^T]^T \in \mathbb{R}^{np}$, where $x_i \in \mathbb{R}^p$ for all $i \in \{1, \dots, n\}$. Similarly, represent the collection of dual variables at the k^{th} iteration of the algorithm as $\mathbf{\Lambda}(k) = \{\lambda_i(k) \mid 1 \leq i \leq n\}$ with the vectorized output $\lambda_{[1:n]} = [\lambda_1^T \ \lambda_2^T \ \dots \ \lambda_n^T]^T \in \mathbb{R}^{np}$. We model the interconnection between the n agents using a directed graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$. We will consider $\mathbf{D}$ as an upper bound on the diameter of the graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$ throughout the rest of the article.

Assumption 1. The directed graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$ is strongly-connected. Let $\mathbf{P} = [p_{ij}]$ be the weighted adjacency matrix associated with the digraph $\mathcal{G}(\mathbf{V}, \mathbf{E})$ be a primitive, column-stochastic matrix with $p_{ij} \geq 0$ for all $i, j \in \mathbf{V}$.

B. Problem Formulation

Consider the distributed optimization problem given in (1). The optimal solution x^* of (1) is a common decision variable which is required to be known to every agent in the network. To utilize the structure of the ADMM scheme at the same time impose this agreement in the solution of all the agents we express (1) as:

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} \quad x_i = x_j \text{ for all } i, j \in \mathbf{V}. \end{aligned} \quad (2)$$

We reformulate (2) to relax the exact consensus constraint to a requirement of a ε -consensus among the estimates of all the agents as follows

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} \quad \|x_i - x_j\| \leq \varepsilon \text{ for all } i, j \in \mathbf{V}. \end{aligned} \quad (3)$$

For most applications such a relaxation poses no issues. Here, the ε -consensus constraint requires the estimate of each agent i to remain ε -close of each other. Let, $\mathbf{x}_{[1:n]} = [x_1^T \ \dots \ x_n^T]^T \in \mathbb{R}^{np}$ represent the vector formed from the entries of $\mathbf{X}$ (by applying the vectorization operator discussed above to $\mathbf{X}$). Let, $\mathcal{C} = \{[x_1^T \ \dots \ x_n^T]^T \in \mathbb{R}^{np} \mid \|x_i - x_j\| \leq \varepsilon, \text{ for all } i, j = 1, 2, \dots, n\}$. The set $\mathcal{C}$ is a collection of vectors $\mathbf{x}_{[1:n]} \in \mathbb{R}^{np}$ such that the norm of the difference between any two sub-vectors of $\mathbf{x}_{[1:n]}$ with dimension n is less than ε . Using this description of set $\mathcal{C}$ and the vector $\mathbf{x}_{[1:n]}$, we write (3) as,

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} \quad \mathbf{x}_{[1:n]} = y, \ y \in \mathcal{C}. \end{aligned} \quad (4)$$

By using the indicator function for the set $\mathcal{C}$ we get,

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n f_i(x_i) + g(y) \\ & \text{subject to} \quad \mathbf{x}_{[1:n]} = y, \end{aligned} \quad (5)$$

where, $g(y)$ is the indicator function of set $\mathcal{C}$:

$$g(y) = \begin{cases} 0 & \text{if } y \in \mathcal{C} \\ \infty & \text{otherwise.} \end{cases} \quad (6)$$

Let, $F(\mathbf{x}_{[1:n]}) = \sum_{i=1}^n f_i(x_i)$, $\mathbf{x}_{[1:n]} = [x_1^T \ \dots \ x_n^T]^T \in \mathbb{R}^{np}$. We denote the Lagrangian function as

$$L(\mathbf{x}_{[1:n]}, y, \lambda) = F(\mathbf{x}_{[1:n]}) + \lambda^T (\mathbf{x}_{[1:n]} - y), \quad (7)$$

where, $\lambda \in \mathbb{R}^{np}$ is the Lagrange multiplier associated with the constraint $\mathbf{x}_{[1:n]} - y = 0$.

C. Standard ADMM Method

The standard ADMM method [13] solves the following optimization problem: for $A \in \mathbb{R}^{n \times p}, B \in \mathbb{R}^{n \times q}, c \in \mathbb{R}^n$,

$$\begin{aligned} & \text{minimize}_{x \in \mathbb{R}^p, y \in \mathbb{R}^q} \quad f(x) + g(y) \\ & \text{subject to} \quad Ax + By = c, \end{aligned} \quad (8)$$

Consider, the augmented Lagrangian with the positive scalar γ and the Lagrange multiplier λ ,

$$L_\gamma(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\gamma}{2}\|Ax + By - c\|^2. \quad (9)$$

ADMM updates of the primal and dual variables are: starting with the initial guess $(x(0), y(0), \lambda(0))$, at each iteration,

$$x(k+1) = \underset{x}{\operatorname{argmin}} L_\gamma(x, y(k), \lambda(k)) \quad (10)$$

$$y(k+1) = \underset{y}{\operatorname{argmin}} L_\gamma(x(k+1), y, \lambda(k)) \quad (11)$$

$$\lambda(k+1) = \lambda(k) + \gamma(Ax(k+1) - By(k+1) - c). \quad (12)$$

III. THE PROPOSED *D-DistADMM* METHOD

In this section, we present the proposed *D-DistADMM* method. Consider, the augmented Lagrangian associated with (5) at any iteration $k \geq 0$ with the positive scalar γ and the Lagrange multiplier $\lambda(k)$,

$$\begin{aligned} L_\gamma(\mathbf{x}_{[1:n]}(k), y(k), \lambda_{[1:n]}(k)) &= \sum_{i=1}^n f_i(x_i(k)) + g(y(k)) \\ &\quad + \frac{\gamma}{2}\|\mathbf{x}_{[1:n]}(k) - y(k)\|^2 + \lambda^T(k)(\mathbf{x}_{[1:n]}(k) - y(k)) \\ &= \sum_{i=1}^n [f_i(x_i(k)) + \frac{\gamma}{2}\|x_i(k) - y_i(k)\|^2 \\ &\quad + \lambda_i^T(k)(x_i(k) - y_i(k))] + g(y(k)), \end{aligned} \quad (13)$$

where, $y_i(k) \in \mathbb{R}^p$ is the i^{th} sub-vector of $y(k) \in \mathbb{R}^{np}$. The ADMM iterations for (13) at each agent i (where, the quantities that do not affect the solution are omitted) are:

$$x_i(k+1) = \underset{x_i}{\operatorname{argmin}} L_\gamma(\mathbf{x}_{[1:n]}, y(k), \lambda_{[1:n]}(k)) \quad (14)$$

$$= \underset{x_i}{\operatorname{argmin}} [f_i(x_i) + \frac{\gamma}{2}\|x_i - y_i(k)\|^2 + \lambda_i^T(k)(x_i - y_i(k))]$$

$$y(k+1) = \underset{y}{\operatorname{argmin}} L_\gamma(\mathbf{x}_{[1:n]}(k+1), y, \lambda_{[1:n]}(k))$$

$$= \underset{y}{\operatorname{argmin}} [g(y) + \frac{\gamma}{2}\|\mathbf{x}_{[1:n]}(k+1) - y\|^2 \quad (15)$$

$$+ \lambda_{[1:n]}^T(k)(\mathbf{x}_{[1:n]}(k+1) - y)]$$

$$= \underset{y}{\operatorname{argmin}} [g(y) + \frac{\gamma}{2}\|\mathbf{x}_{[1:n]}(k+1) - y + \frac{1}{\gamma}\lambda_{[1:n]}(k)\|^2]$$

$$\lambda_i(k+1) = \lambda_i(k) + \gamma(x_i(k+1) - y_i(k+1)). \quad (16)$$

The update in (14) can be solved using a classical methods such as gradient descent [22]. The $y(k+1)$ update in (15) is the projection of $\mathbf{x}_{[1:n]}(k+1) + \frac{1}{\gamma}\lambda_{[1:n]}(k)$ onto the set $\mathcal{C}$, whereas (16) gives the dual update at each agent i . To find a solution of (15) we employ a distributed finite-time ε -consensus protocol in higher-dimensions.

A. Finite-time ε -consensus protocol

In this subsection, we will present a finite-time “approximate” consensus protocol which we call the ε -consensus protocol. Consider, a set of n agents with the interconnection between them modeled as a directed graph $\mathcal{G}(\mathbf{V}, \mathbf{E})$. Each agent has a vector $u_i(0) \in \mathbb{R}^p$. The agents’ objective is to determine the average $\bar{u} = \frac{1}{n} \sum_{i=1}^n u_i(0)$, of the vectors distributively. To this end, the agents follow an iterative

protocol, involving an additional scalar variable v_i , and having the following updates for $k = 0, 1, \dots$

$$u_i(k+1) = p_{ii}u_i(k) + \sum_{j \in \mathcal{N}_i^-} p_{ij}u_j(k) \quad (17)$$

$$v_i(k+1) = p_{ii}v_i(k) + \sum_{j \in \mathcal{N}_i^-} p_{ij}v_j(k) \quad (18)$$

$$w_i(k+1) = \frac{1}{v_i(k+1)}u_i(k+1), \quad (19)$$

with, $p_{ij} \geq 0$ for all $i, j \in \mathbf{V}$, $\sum_j p_{ij} = 1$ and, $v_i(0) = 1$ for all $i \in \mathbf{V}$. The variable $\frac{1}{v_i(k)}u_i(k) \in \mathbb{R}^{np}$ is an estimate of $\bar{u}$ with each agent at any iteration k . It has been established in the literature that the vector estimates $\frac{1}{v_i(k)}u_i(k)$ converge to the average $\bar{u}$ as time progresses, this is stated formally in the next theorem.

Theorem 1. *Let Assumption 1 hold. Let $\{u_i(k)\}, \{v_i(k)\}$ and $\{w_i(k)\}$ for $k = 1, 2, \dots$ be the sequences generated by (17), (18), and (19) at each agent $i \in \mathbf{V}$. Then $\frac{1}{v_i(k)}u_i(k)$ asymptotically converges to $\bar{u} = \frac{1}{n} \sum_{i=1}^n u_i(0)$ for all $i \in \mathbf{V}$, that is*

$$\lim_{k \rightarrow \infty} \frac{1}{v_i(k)}u_i(k) = \frac{1}{n} \sum_{i=1}^n u_i(0), \text{ for all } i \in \mathbf{V}.$$

Proof. Refer [23], [24] and [25] for the proof. $\square$

Theorem 1 suggests that after some large iteration t the agent states’ will become close enough to $\bar{u}$. The proposed ε -consensus protocol gives a distributed criterion to determine this iteration t such that the agent states’ are ε -close to each other. Our ε -consensus algorithm is a distributed finite-time (global) termination protocol for consensus algorithms for vector states which was first introduced in [25]. This protocol is a generalization of the works in [26], [27] proposed originally for consensus on a scalar value. Here, each agent maintains an auxiliary scalar value R_i called the radius of the agent i . The radius variable is designed to track the ball with a minimal radius enclosing all the agent states’ $\frac{1}{v_i(k)}u_i(k)$. Each agent updates its radius R_i based on the states $\frac{1}{v_j(k)}u_j(k)$ of its neighbors as follows:

$$R_i(k+1, \ell) = \max_{j \in \mathcal{N}_i^-} \{\|w_i(k+1, \ell) - w_j(k, \ell)\| + R_j(k, \ell)\}, \quad (20)$$

where, ℓ is any starting time-instant and $R_i(0, \ell) := 0$, for all $i \in \mathbf{V}$. Denote, $\mathcal{B}(w_i(k, \ell), R_i(k, \ell))$ as the ball of radius $R_i(k, \ell)$ centered at $w_i(k, \ell)$. At each iteration k , R_i is updated according to (20). After $\mathbf{D}$ number of iteration starting at an iteration ℓ , the ball $\mathcal{B}(w_i(\mathbf{D} + \ell), R_i(\mathbf{D} + \ell))$ encloses the states $\frac{1}{v_i(\ell)}u_i(\ell)$ of all the agents. This is stated formally in the following Lemma,

Lemma 1. *Let $\{u_i(k)\}, \{v_i(k)\}$ and $\{w_i(k)\}$ for $k = 1, 2, \dots$ be the sequences generated by (17), (18), and (19) at each agent $i \in \mathbf{V}$. Let $R_i(k, \ell)$ for a fixed $\ell \geq 0$ be updated as in (20), for all $k \geq 0$ and $i \in \mathbf{V}$. Then, under Assumption 1,*

$$w_j(\ell) \in \mathcal{B}(w_i(\mathbf{D} + \ell), R_i(\mathbf{D}, \ell)), \text{ for all } j, i \in \mathbf{V}.$$

Proof. Refer [25] for the complete proof. $\square$

Based on the update (20) and Lemma 1, Algorithm 1 suggest a protocol to determine the radius of the minimal enclosing ball distributively. Starting from the initial iteration $k = 0$ each agent follow (20). At every iteration of the form $\ell\mathbf{D}$, for $\ell \geq 0$, $R_i(\mathbf{D}, \ell\mathbf{D})$ is compared to the parameter ε , if $R_i(\mathbf{D}, \ell\mathbf{D}) < \varepsilon$ then the agent states are ε -close to $\bar{u}$ and the iteration stops else $R_i(\mathbf{D}, \ell\mathbf{D})$ is re-initialized and the protocol continues.

Algorithm 1: Finite-time ε -consensus protocol at each node $i \in \mathbf{V}$

Input:

| Tolerance $\varepsilon > 0$; Diameter upper bound $\mathbf{D}$;

Initialize:

| $u_i(0)$; $v_i(0) = 1$; $R_i(0) = 0$; $\ell := 1$

Repeat for $k = 0, 1, 2, \dots$

| /* consensus updates (17)–(19) */

| $u_i(k+1) = p_{ii}u_i(k) + \sum_{j \in \mathcal{N}_i^-} p_{ji}u_j(k)$

| $v_i(k+1) = p_{ii}v_i(k) + \sum_{j \in \mathcal{N}_i^-} p_{ji}v_j(k)$

| $w_i(k+1) = \frac{1}{v_i(k+1)}u_i(k+1)$

| /* radius update (20) */

| $R_i(k+1) = \max_{j \in \mathcal{N}_i^-} \{\|w_i(k+1) - w_j(k)\| + R_j(k)\}$

if $k = \ell\mathbf{D}$ **then**

| $R_i(\mathbf{D}, \ell\mathbf{D}) = R_i(k+1)$;

| **if** $R_i(\mathbf{D}, \ell\mathbf{D}) < \varepsilon$ **then**

| | **break** ; // converged

| **else**

| | $R_i(k+1) = 0$;

| | $\ell = \ell + 1$;

end

B. y variable updates in D -DistADMM

The ε -consensus protocol projects the input vector $\mathbf{u} = [u_1^T(0), \dots, u_n^T(0)]^T \in \mathbb{R}^{np}$ onto the set $\mathcal{C}$. This property of the ε -consensus protocol can be used to find a solution which is ε -close to the actual (optimal) solution of (15) in a distributed manner. The following Lemma establish this.

Lemma 2. *Let, $\{x_i(k+1)\}_{i=1, \dots, n}$, $\{\lambda_i(k+1)\}_{i=1, \dots, n}$ be the estimates produced by the update rules (14) and (16) for $k \geq 0$. Then, for any iteration k of Algorithm 2, the ε -consensus protocol with the initialization $u_i(0) = x_i(k+1) + \frac{1}{\gamma}\lambda_i(k)$ and $v_i(0) = 1$, for all $i \in \mathbf{V}$ and given $\varepsilon > 0$, generates a set of outputs $\{y_i(k+1)\}_{i=1, \dots, n}$ where $y_i(k+1) \in \mathbb{R}^p$, such that*

$$\|y_i(k+1) - y_j(k+1)\| \leq \varepsilon, \text{ for all } i, j \in \mathbf{V}.$$

Further, the vector $y(k+1) = [y_1(k+1)^T, \dots, y_n(k+1)^T]^T \in \mathbb{R}^{np}$ is ε -close to the (optimal) solution $\frac{1}{n} \sum_{i=1}^n [x_i(k+1) + \lambda_i(k)]$ of the update (15).

Proof. It directly follows from Theorem 1, there exist $t > 0$ such that

$$\left\| \frac{1}{v_i(t)}u_i(t) - \frac{1}{v_j(t)}u_j(t) \right\| \leq \varepsilon, \text{ for all } i, j \in \mathbf{V}.$$

Letting $y_i(k+1) = \frac{1}{v_i(t)}u_i(t)$ for all $i \in \mathbf{V}$, gives the desired inequality. Since, $\|y_i(k+1) - y_j(k+1)\| \leq \varepsilon$,

for all $i, j \in \mathbf{V}$ therefore, the vector $y(k+1) = [y_1^T(k+1), \dots, y_n^T(k+1)]^T \in \mathcal{C}$. Thus, by Theorem 1 $y(k+1) = \left[\frac{1}{v_1(t)}u_1^T(t), \dots, \frac{1}{v_n(t)}u_n^T(t) \right]^T$ is such that,

$$\left\| \frac{1}{v_i(t)}u_i(t) - \frac{1}{n} \sum_{i=1}^n [x_i(k+1) + \lambda_i(k)] \right\| \leq \varepsilon,$$

Note, that $\tilde{y} := \frac{1}{n} \sum_{i=1}^n [x_i(k+1) + \lambda_i(k)]$ is the actual (optimal) solution to (15). Thus, Lemma 2 suggests that utilizing the finite-time ε -consensus protocol we can produce a solution $y(k+1)$ which is ε -close to $\tilde{y}$, that is,

$$\tilde{y} = y(k+1) + e(k+1), \text{ with } \|e(k+1)\| \leq \sqrt{n}\varepsilon. \quad (21)$$

This completes the proof. $\square$

The proposed D -DistADMM algorithm is presented in Algorithm 2.

Algorithm 2: D -DistADMM Algorithm

Input:

| choose $\gamma > 0, \varepsilon > 0, k = 1$

Initialize:

| For all $i \in \mathbf{V}$, pick $x_i(0), y_i(0), \lambda_i(0), \in \mathbb{R}^p$

Repeat for $k = 0, 1, 2, \dots$

| **for** $i = 1, 2, \dots, n$, (**In parallel**) **do**

$$x_i(k+1) = \underset{x_i}{\operatorname{argmin}} [f_i(x_i) + \frac{\gamma}{2} \|x_i - y_i(k)\|^2$$

$$+ \lambda_i^T(k)(x_i - y_i(k))]$$

| send $x_i(k+1) + \frac{1}{\gamma}\lambda_i(k)$ to $j \in \mathcal{N}_i^+$

$$y_i(k+1) \xleftarrow{\varepsilon\text{-consensus}} (x_i(k+1) + \frac{1}{\gamma}\lambda_i(k), \varepsilon)$$

$$\lambda_i(k+1) = \lambda_i(k) + \gamma(x_i(k+1) - y_i(k+1))$$

| **end**

| **until** a stopping criterion is met

IV. CONVERGENCE RESULTS FOR D -DistADMM

Let, Ω^* denote the set of solutions to (5). The results in this paper holds under the following assumptions:

Assumption 2. *The optimal solution set Ω^* is non-empty and compact.*

Assumption 3. *For all $i \in \{1, 2, \dots, n\}$, f_i is a not-necessarily differentiable, proper closed-convex function having a lower bound.*

In what follows we will make use of the following identities:

1. Sub-gradient inequality: Let $f : \mathbb{R}^p \rightarrow \mathbb{R}$ be a convex function. Let $h(x)$ be a sub-gradient of f at $x \in \operatorname{dom} f$, then for all $z \in \operatorname{dom} f$,

$$f(z) \geq f(x) + h^T(x)(z - x) \quad (22)$$

2. For $w_1, w_2, w_3, w_4 \in \mathbb{R}^p$

$$(w_1 - w_2)^T(w_3 - w_4) = \frac{1}{2}(\|w_1 - w_4\|^2 - \|w_1 - w_3\|^2) + \frac{1}{2}(\|w_2 - w_3\|^2 - \|w_2 - w_4\|^2) \quad (23)$$

Theorem 2. *Let $\{x_i(k)\}_{i=1, \dots, n}$, $\{\lambda_i(k)\}_{i=1, \dots, n}$, $k = 1, 2, \dots$, be the sequence of estimates and the dual variables generated by the D -DistADMM algorithm. Further, let $\{y(k) \in \mathbb{R}^{np}\}$, $k = 1, 2, \dots$ be the outputs generated*

by the ε -consensus protocol as given in Lemma 2. Under Assumptions 2 and 3 for any fixed $\lambda \in \mathbb{R}^{np}$, the following holds for all $\ell > 0$,

$$L(\hat{\mathbf{x}}(\ell), \hat{y}(\ell), \lambda) - L(\mathbf{x}^*, y^*, \lambda) \leq \frac{\|\lambda - \lambda_{[1:n]}(0)\|^2}{2\gamma\ell} + \frac{\gamma\|\mathbf{x}^* - y(0)\|^2}{2\ell} + O(\varepsilon),$$

where, $(\mathbf{x}^*, y^*) \in \Omega^*$ is any optimal solution of (5), $\hat{\mathbf{x}}(\ell) := \frac{1}{\ell} \sum_{k=0}^{\ell-1} \mathbf{x}_{[1:n]}(k+1)$, $\hat{y}(\ell) := \frac{1}{\ell} \sum_{k=0}^{\ell-1} y(k+1)$, and ε is a pre-specified tolerance chosen independent of the problem.

Proof. See Subsection VI-A $\square$

Thus, Theorem 2 establishes an ergodic $O(1/k)$ rate of convergence for the proposed *D-DistADMM* algorithm in terms of the difference of the Lagrangian functions.

V. NUMERICAL SIMULATIONS

In this section, we present the results of numerical simulations obtained by applying the *D-DistADMM* method of Algorithm 2 to solve distributed least squares problem. All the numerical simulations in this section are done on a directed network of containing 20 nodes. Consider the following distributed least squares problem

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^n \|b_i - A_i x\|^2. \quad (24)$$

Here, each agent $i \in \mathbf{V}$ has a measured data $b_i \in \mathbb{R}^n$ a scaling matrix $A_i \in \mathbb{R}^{n \times p}$. The unknown quantity (or the true state) which is required to be estimated is x . The entries of the A_i matrices, and the measured data b_i are chosen from i.i.d samples of standard normal distribution $\mathcal{N}(0, 1)$. The underlying true signal x is also generated from an i.i.d sample of the standard normal distribution.

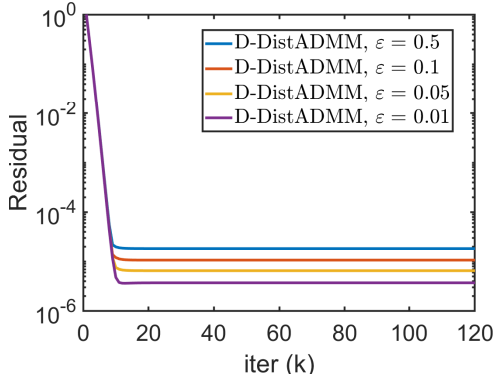


Fig. 1: Plots of residuals $f_i(\hat{\mathbf{x}}_i(k)) - f^*$ against the number of iterations for different values of ε .

Fig. 1 provides the trajectories of the residual $f_i(\hat{\mathbf{x}}_i(k)) - f^*$ versus the iteration counter k . The residual decreases continuously until reaching an $O(\varepsilon)$ neighborhood. Also, as expected the error from the optimal value f^* becomes less and less as a smaller value of the parameter ε is chosen.

VI. CONCLUSION

In this article, we present a novel distributed ADMM based algorithm to solve the distributed optimization problem in directed networks in contrast to the existing schemes applicable only for undirected networks. Our scheme is based on a finite-time ε -consensus that enable the agents to find approximate optimal solution in a finite-number of iterations by just utilizing local communications. We prove that the proposed *D-DistADMM* algorithm converges at a $O(1/k)$ rate even when the individual functions f_i are convex but not necessarily differentiable. The proposed *D-DistADMM* algorithm eliminates the need of the optimization step involving the global variable in the standard ADMM setup and replaces it with a less computation intensive ε -consensus protocol. The efficacy of the *D-DistADMM* algorithm is demonstrated by solving a distributed least squares problem.

A. Proof of Theorem 2

Suppose $(\mathbf{x}^*, y^*)$ is an optimal solution of (5). At any iteration $k+1$ of the *D-DistADMM* algorithm, by the optimality conditions [22] for all $i \in \mathbf{V}$,

$$[\mathbf{x} - \mathbf{x}_{[1:n]}(k+1)]^T [\mathbf{h}(\mathbf{x}_{[1:n]}(k+1)) + \lambda_{[1:n]}(k+1) + \gamma(y(k+1) - y(k))] \geq 0,$$

where, $\mathbf{h}(\mathbf{x}_{[1:n]}(k+1)) = [h_1^T(x_1(k+1)), \dots, h_n^T(x_n(k+1))]^T$ is a vector of subgradients $h_i(x_i(k+1))$ stacked together. Let, $\tilde{y}$ denote the solution to (15) given in (21). By the optimality conditions we have,

$$\begin{aligned} d(\tilde{y}) - \lambda_{[1:n]}(k) - \gamma(\mathbf{x}_{[1:n]}(k+1) - \tilde{y}) &= 0, \\ d(\tilde{y}) - \lambda_{[1:n]}(k) - \gamma(\mathbf{x}_{[1:n]}(k+1) + y(k+1) - e(k+1)) &= 0, \end{aligned}$$

where, $d(\tilde{y})$ is the sub-gradient of g at $\tilde{y}$. Noticing, F and g are convex and using sub-gradient inequality (22) we have,

$$\begin{aligned} &F(\mathbf{x}_{[1:n]}(k+1)) - F(\mathbf{x}) + g(\tilde{y}) - g(y) \\ &\leq -[\mathbf{x} - \mathbf{x}_{[1:n]}(k+1)]^T \mathbf{h}(\mathbf{x}_{[1:n]}(k+1)) - [y - \tilde{y}]^T d(\tilde{y}) \\ &\leq [\mathbf{x} - \mathbf{x}_{[1:n]}(k+1)]^T [\lambda_{[1:n]}(k+1) + \gamma(y(k+1) - y(k))] \\ &\quad + [y - \tilde{y}]^T [-\lambda_{[1:n]}(k+1)] + \gamma[y - \tilde{y}]^T e(k+1) \\ &\leq [\mathbf{x} - \mathbf{x}_{[1:n]}(k+1)]^T [\lambda_{[1:n]}(k+1) + \gamma(y(k+1) - y(k))] \\ &\quad + [y - y(k+1)]^T [-\lambda_{[1:n]}(k+1)] + \gamma\varepsilon^2 + \gamma\varepsilon\|y\| \\ &\quad + e(k+1)^T [\lambda(k+1) + \gamma y(k+1)] \\ &\leq \lambda_{[1:n]}^T(k+1)[(\mathbf{x} - \mathbf{x}_{[1:n]}(k+1)) + (y(k+1) - y)] \\ &\quad + \gamma(\mathbf{x} - \mathbf{x}_{[1:n]}(k+1))^T (y(k+1) - y(k)) + \varepsilon\|\lambda(0)\| \\ &\quad + \varepsilon\gamma R_d + \gamma\varepsilon^2 + \gamma\varepsilon\|y\| + \gamma\varepsilon\|\mathbf{x}^*\| \end{aligned}$$

Here, $R_d = \sup_{\alpha} R_{d_{\alpha}}$, where, $R_{d_{\alpha}}$ is the diameter of the level set $X_{\alpha} = \{x \in \mathbb{R}^{np} | f(x) \leq \alpha\}$. Note that the supremum exists and is finite due to Assumptions 2 and 3. Setting $(\mathbf{x}, y) = (\mathbf{x}^*, y^*)$ gives,

$$\begin{aligned} &F(\mathbf{x}_{[1:n]}(k+1)) - F(\mathbf{x}^*) + g(y(k+1)) - g(y^*) \\ &\leq \lambda_{[1:n]}^T(k+1)(y(k+1) - \mathbf{x}_{[1:n]}(k+1)) \\ &\quad + \gamma(\mathbf{x}^* - \mathbf{x}_{[1:n]}(k+1))^T (y(k+1) - y(k)) + \gamma\varepsilon^2 \\ &\quad + \gamma\varepsilon\|y^*\| + \gamma\varepsilon\|\mathbf{x}^*\| + \varepsilon\|\lambda(0)\| + \varepsilon\gamma R_d \end{aligned}$$

For a fixed λ , we have

$$\begin{aligned}
& F(\mathbf{x}_{[1:n]}(k+1)) - F(\mathbf{x}^*) + g(y(k+1)) - g(y^*) \\
& + \lambda^T(\mathbf{x}_{[1:n]}(k+1) - y(k+1)) \\
& \leq [\lambda_{[1:n]}(k+1) - \lambda]^T(y(k+1) - \mathbf{x}_{[1:n]}(k+1)) \\
& \quad + \gamma(\mathbf{x}^* - \mathbf{x}_{[1:n]}(k+1))^T(y(k+1) - y(k)) \\
& \quad + \gamma\epsilon^2 + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d \\
& = \frac{1}{\gamma}[\lambda - \lambda_{[1:n]}(k+1)]^T[\lambda_{[1:n]}(k+1) - \lambda_{[1:n]}(k)] \\
& \quad + \gamma(\mathbf{x}^* - \mathbf{x}_{[1:n]}(k+1))^T(y(k+1) - y(k)) \\
& \quad + \gamma\epsilon^2 + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d \quad (25)
\end{aligned}$$

Using (25) and (23) we get,

$$\begin{aligned}
& F(\mathbf{x}_{[1:n]}(k+1)) - F(\mathbf{x}^*) + g(y(k+1)) - g(y^*) \\
& + \lambda^T(\mathbf{x}_{[1:n]}(k+1) - y(k+1)) \\
& \leq \frac{1}{2\gamma}(\|\lambda - \lambda_{[1:n]}(k)\|^2 - \|\lambda - \lambda_{[1:n]}(k+1)\|^2) \\
& \quad - \|\lambda_{[1:n]}(k+1) - \lambda_{[1:n]}(k)\|^2 \\
& \quad + \frac{\gamma}{2}(\|\mathbf{x}^* - y(k)\|^2 - \|\mathbf{x}^* - y(k+1)\|^2) \\
& + \|\mathbf{x}_{[1:n]}(k+1) - y(k+1)\|^2 - \|\mathbf{x}_{[1:n]}(k+1) - y(k)\|^2 \\
& \quad + \gamma\epsilon^2 + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d \\
& \leq \frac{1}{2\gamma}(\|\lambda - \lambda_{[1:n]}(k)\|^2 - \|\lambda - \lambda_{[1:n]}(k+1)\|^2) \\
& \quad + \frac{\gamma}{2}(\|\mathbf{x}^* - y(k)\|^2 - \|\mathbf{x}^* - y(k+1)\|^2) \\
& \quad + \frac{\gamma\epsilon^2}{2} + \gamma\epsilon^2 + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d,
\end{aligned}$$

the last inequality follows from (16) and dropping the negative term. Summing from $k = 0$ to $k = \ell - 1$,

$$\begin{aligned}
& \sum_{k=0}^{\ell-1} [F(\mathbf{x}_{[1:n]}(k+1)) - F(\mathbf{x}^*) + g(y(k+1)) - g(y^*) \\
& \quad + \lambda^T(\mathbf{x}_{[1:n]}(k+1) - y(k+1))] \\
& \leq \frac{\|\lambda - \lambda_{[1:n]}(0)\|^2}{2\gamma} + \frac{\gamma\|\mathbf{x}^* - y(0)\|^2}{2} + \frac{\gamma\epsilon^2}{2} + \gamma\epsilon^2 \\
& \quad + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d
\end{aligned}$$

Let, $\hat{\mathbf{x}}(\ell) := \frac{1}{\ell} \sum_{k=0}^{\ell-1} \mathbf{x}_{[1:n]}(k+1)$, $\hat{y}(\ell) := \frac{1}{\ell} \sum_{k=0}^{\ell-1} y(k+1)$. Since, F and g are convex we have, $\sum_{k=0}^{\ell-1} F(\mathbf{x}_{[1:n]}(k+1)) \geq \ell F(\hat{\mathbf{x}}(\ell))$ and $\sum_{k=0}^{\ell-1} g(y(k+1)) \geq \ell g(\hat{y}(\ell))$. Therefore, for any $\lambda \in \mathbb{R}^{np}$ we have,

$$\begin{aligned}
& L(\hat{\mathbf{x}}(\ell), \hat{y}(\ell), \lambda) - L(\mathbf{x}^*, y^*, \lambda) \\
& \leq \frac{\|\lambda - \lambda_{[1:n]}(0)\|^2}{2\gamma\ell} + \frac{\gamma\|\mathbf{x}^* - y(0)\|^2}{2\ell} \\
& \quad + \frac{\gamma\epsilon^2}{2} + \gamma\epsilon^2 + \gamma\epsilon\|\mathbf{y}^*\| + \gamma\epsilon\|\mathbf{x}^*\| + \epsilon\|\lambda(0)\| + \epsilon\gamma R_d \\
& = \frac{\|\lambda - \lambda_{[1:n]}(0)\|^2}{2\gamma\ell} + \frac{\gamma\|\mathbf{x}^* - y(0)\|^2}{2\ell} + O(\epsilon).
\end{aligned}$$

REFERENCES

- [1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 20–27.
- [2] A. Olshevsky, "Efficient information aggregation strategies for distributed control and signal processing," *arXiv preprint arXiv:1009.6036*, 2010.
- [3] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *2012 50th annual allerton conference on communication, control, and computing (allerton)*. IEEE, 2012, pp. 1543–1550.
- [4] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control systems magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [5] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.
- [6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.
- [7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.
- [8] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2010.
- [9] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.
- [10] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.
- [11] R. Xin, S. Kar, and U. A. Khan, "Gradient tracking and variance reduction for decentralized optimization and machine learning," *arXiv preprint arXiv:2002.05373*, 2020.
- [12] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.
- [13] 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.
- [14] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.
- [15] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5445–5450.
- [16] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed admm over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.
- [17] F. Mansoori and E. Wei, "A flexible framework of first-order primal-dual algorithms for distributed optimization," *arXiv preprint arXiv:1912.07526*, 2019.
- [18] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.
- [19] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-admm: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.
- [20] R. Diestel, *Graph Theory*. Berlin, Germany: Springer-Verlag, 2006.
- [21] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [22] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.
- [23] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings*. IEEE, 2003, pp. 482–491.
- [24] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed strategies for average consensus in directed graphs," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 2124–2129.
- [25] J. Melbourne, G. Saraswat, V. Khatana, S. Patel, and M. V. Salapaka, "On the geometry of consensus algorithms with application to distributed termination in higher dimension," in *the proceedings of International Federation of Automatic Control (IFAC)*, 2020.
- [26] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *45th Annual Allerton Conf*, 2007, pp. 715–720.
- [27] M. Prakash, S. Talukdar, S. Attree, V. Yadav, and M. V. Salapaka, "Distributed stopping criterion for consensus in the presence of delays," *IEEE Transactions on Control of Network Systems*, 2019.