

DC-DistADMM: ADMM Algorithm for Constrained Optimization Over Directed Graphs

Vivek Khatana  and Murti V. Salapaka , *Fellow, IEEE*

Abstract—This article reports an algorithm for multiagent distributed optimization problems with a common decision variable, local linear equality, and inequality, constraints and set constraints with convergence rate guarantees. The algorithm accrues all the benefits of the alternating direction method of multipliers (ADMM) approach. It also overcomes the limitations of existing methods on convex optimization problems with linear inequality, equality, and set constraints by allowing directed communication topologies. Moreover, the algorithm can be synthesized distributively. The developed algorithm has: first, a $O(1/k)$ rate of convergence, where k is the iteration counter, when individual functions are convex but not-necessarily differentiable, and second, a geometric rate of convergence to any arbitrary small neighborhood of the optimal solution, when the objective functions are smooth and restricted strongly convex at the optimal solution. The efficacy of the algorithm is evaluated by a comparison with state-of-the-art constrained optimization algorithms in solving a constrained distributed ℓ_1 -regularized logistic regression problem, and unconstrained optimization algorithms in solving a ℓ_1 -regularized Huber loss minimization problem. Additionally, a comparison of the algorithm's performance with other algorithms in the literature that utilize multiple communication steps is provided.

Index Terms—Alternating direction method of multipliers (ADMM), constrained optimization, directed graphs, finite-time consensus, distributed optimization, multiagent networks.

I. INTRODUCTION

CONSIDER a group of n agents connected through a directed graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of vertices and directed edges, respectively. Each agent can transmit information to other agents restricted by the directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; an agent i can transmit to agent j if a directed link $i \rightarrow j$ exists

Manuscript received 11 February 2022; revised 24 August 2022; accepted 29 October 2022. Date of publication 14 November 2022; date of current version 30 August 2023. This work was supported by the Advanced Research Projects Agency-Energy OPEN through the project titled "Rapidly Viable Sustained Grid" under Grant DE-AR0001016. Recommended by Associate Editor J. Lavaei. (Corresponding author: Vivek Khatana.)

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: khata010@umn.edu; murtis@umn.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3221856>.

Digital Object Identifier 10.1109/TAC.2022.3221856

in $\mathcal{E}$. The agents focus on solving the following distributed optimization problem:

$$\begin{aligned} & \underset{\tilde{x} \in \mathbb{R}^p}{\text{minimize}} && \tilde{f}(\tilde{x}) = \sum_{i=1}^n \tilde{f}_i(\tilde{x}) \\ & \text{subject to} && C_i \tilde{x} = c_i, D_i \tilde{x} \leq d_i \quad \forall i \in \mathcal{V}, \quad \tilde{x} \in \bigcap_{i=1}^n \mathcal{X}_i \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^p$ is a global optimization variable. $\tilde{f}_i : \mathbb{R}^p \rightarrow \mathbb{R}$ is the local objective function of agent i . $\mathcal{X}_i$ is a convex constraint set associated with the variables of agent i . $C_i \tilde{x} = c_i$ with $C_i \in \mathbb{R}^{m_1 \times p}$, $c_i \in \mathbb{R}^{m_1}$ and $D_i \tilde{x} \leq d_i$ with $D_i \in \mathbb{R}^{m_2 \times p}$, $d_i \in \mathbb{R}^{m_2}$ are the local equality and inequality constraints of agent i . Many problems in various engineering fields, such as wireless systems, multiagent coordination and control [1], and machine learning [2] can be posed in the form of problem (1). Unlike a class of distributed optimization problems that involve global coupling constraints (see, for example, [3], [4], [5]), in problem (1) agents seek to determine a common solution $\tilde{x}$ that is required to be known by each agent.

Early works on the distributed optimization problem can be found in seminal papers [6], [7]. Current approaches for solving optimization problems can be broadly classified as follows: 1) primal methods that update the agent estimates of the solution by utilizing a step-size rule based on the gradient information (at the current estimate) of the objective function to steer towards the optimal solution, and 2) dual-based optimization methods that employ Lagrange multipliers. For optimization problems that are unconstrained [with no equality, inequality and set constraints in problem (1)], examples [8], [9], [10], [11], [12] (and references therein) take a primal based approach, whereas [13], [14], [15], [16], [17], [18], [19], [20] take a dual based approach. The works in [14], [15], [16], [17], [18], [19], [20] motivated by the advantages of parallelizability and good convergence results, (see [21]) adopt the alternating direction method of multipliers (ADMM) dual-based approach; Wei and Ozdaglar [14] proposed the ADMM-based algorithm for convex objective functions with an $O(1/k)$ rate of convergence, the works in [15] and [16] established linear rate of convergence with globally strongly convex objective functions, the work in [18] is based on minimizing a proximal first-order approximation of smooth and globally strongly convex functions, and Iutzeler et al. [19] provided a linear rate of convergence for a twice continuously differentiable and locally strongly convex functions. In contrast to the dual-based approaches [13], [14], [15], [16], [17], [18], [19], [20] for unconstrained problems, this article

provides an algorithm with provable convergence guarantees, which is not restricted by assumption of bidirectional (or undirected) communication topologies and does not need centralized information for designing the algorithm.

The works in [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], and [36] consider constrained distributed optimization problems similar to (1). For solving constrained convex optimization problems of the form (1) distributively, many of the existing state-of-the-art algorithms [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33] utilize primal algorithms based on projections onto the constraint set and the (sub)gradient information of the individual objective functions. Here, the works in [27], [28], [29], [31], [34], [35], and [36] focused only on set constraints; equality constraints are considered in [26] and [30] while inequality constraints are the focus in [22], [23], [24], [25], [26], [32], and [33]. To establish convergence, the work in [25] assumes the differentiability of the individual objective functions. The assumption of individual functions being Lipschitz continuous and convex quadratic over the constraint set is required in [28]. The works in [29] and [32] require the assumption of the set on which gradient of the individual functions is zero to be bounded and a compact optimal solution set, respectively. Twice continuous differentiability and bounded Hessian matrix for the individual functions is assumed in [30]. The works in [31] and [33], require bounded (sub)gradients. In contrast to the assumptions above in the primal-based approaches in [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], and [33], the method reported in this article does not require differentiability nor does it require bounded (sub)gradients. The works in [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33] established convergence to an optimal solution; however, unlike the focus of this article, no convergence rate/iteration complexity estimates are determined. Moreover, this article is based on ADMM and, thus, accrues its advantages; ADMM is shown to have a better empirical performance than the primal (projected) subgradient methods [13].

The works in [34], [35], and [36] utilized a primal-dual approach for constrained problems. Here, Yuan et al. [35] required bounded (sub)gradients to establish an $O(k^{-1/4})$ rate of convergence for the objective function residual and Lei et al. [36] does not establish rate of convergence in the presence of constraints. As alluded to earlier, ADMM-based approach (which is a Lagrangian dual based approach) has several advantages (see [13]); however, results on distributed constrained optimization problems using ADMM are sparse. Here, Mota et al. [34], which considers dual ADMM-like method, does not provide a convergence rate/iteration complexity analysis.

Note that most existing constrained optimization schemes [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [34], [35], [36] are designed using centralized information and work under the assumption of bi-directional (or undirected) communication networks. Moreover, the algorithms in [22], [23], [25], [26], [27], [28], [33], [35], and [36] are continuous-time algorithms that pose the need for further analysis under discretization schemes needed for implementation. Compared to these our proposed algorithm is based on discrete-time iterations, which can be easily implemented in a practical setting. However, unlike some existing work in the literature [23], [24], in this work, we do not consider nonlinear inequality constraints and time-varying graphs. We now summarize the discussion above to delineate the main contribution of the article as follows.

The main contribution of the article is an ADMM-based algorithm called directed constrained-distributed alternating direction method of multipliers (DC-DistADMM) that solves problem (1). The novel features of the algorithm for unconstrained and constrained optimization problems are summarized as follows.

- 1) With respect to unconstrained optimization problems, DC-DistADMM is a Lagrangian dual-based algorithm:
 - a) that accommodates directed graphs allowing for nonsymmetric communication topologies. This feature considerably widens the applicability as in most application scenarios, agents do not have the same range of communication;
 - b) that is amenable to distributed synthesis scenarios and extends the applicability of ADMM method to applications where a plug and play operation is required [37], [38] (distributed synthesis is detailed in, [49] Appendix A). The DC-DistADMM algorithm has column stochastic updates and does not require the agents to know global network interconnection information.
- 2) With respect to the constrained optimization problems, DC-DistADMM algorithm:
 - a) is a Lagrangian dual-based algorithm for constrained optimization problems that accommodates directed graphs and allows for distributed synthesis;
 - b) to the best of the authors' knowledge DC-DistADMM has the best convergence rates in the constrained optimization literature. Most works [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [36] do not provide convergence rate guarantees while the work in [35] provides a worse one. The DC-DistADMM algorithm converges to an optimal solution of problem (1) under mild assumptions (Theorem 3). The algorithm has a $O(1/k)$ rate of convergence, where k is the iteration counter, when the individual functions $\tilde{f}_i$ are convex but not-necessarily differentiable. It has a geometric rate of convergence to any arbitrary neighborhood of the optimal solution when the objective functions are smooth and restricted strongly convex at the optimal solution of problem (1). DC-DistADMM does not require the individual functions $\tilde{f}_i$ to be differentiable for convergence to an optimal solution neither bounded subgradients and globally strongly convex or twice continuously differentiable is assumed for the derivation of convergence rate estimates unlike existing analysis in the literature (see [11], [15], [16], [19], [25], [28], [29], [30], [31], [32], [33], [35], [39]).
- 3) ADMM-based approaches and more generally Lagrangian dual-based approaches for distributed constrained problems involve in each iteration an information mixing step among agents and a step that improves the objective. An important technical contribution of the article is that it transforms the original problem (1) to an equivalent problem [see (7)] that allows for carrying out

almost *complete* mixing via any consensus strategy for the information mixing step to be followed by the objective improvement step. This reformulation with its advantages is being employed by other groups including works in [40], [41], [42], and [43] with consensus strategies other than the one employed in this article.

- 4) The total number of communication steps for DC-DistADMM by the k th iterate is within a factor of $\log k$ of the optimal lower bound in obtaining a $O(1/k)$ rate of convergence. Empirical data corroborates that DC-DistADMM, with respect to the computational performance, outperforms other distributed constrained and unconstrained optimization approaches.

The authors have introduced a preliminary version [44] of the method developed here termed as D-DistADMM. The current work presents a significant generalization, DC-DistADMM, where a *constrained* distributed optimization problem is considered. The constrained optimization problem poses significant technical challenges over the unconstrained case; here, stronger and more comprehensive results under less restrictive assumptions are established. In contrast to [44] this article provides the convergence rate guarantees, establishes upper bounds on the total communication steps required for achieving the convergence rate estimates (remarks 2 and 5), and validation of the DC-DistADMM algorithm's applicability using detailed numerical tests and comparison with other state-of-the-art algorithms is provided.

The rest of this article is organized as follows. In Section II, the problem under consideration is discussed in detail and some basic definitions and notations used in the article are presented. Section III presents the DC-DistADMM algorithm along with the finite-time ε -consensus protocol in detail; and provide supporting analysis for the ε -consensus protocol. The convergence analysis of the proposed algorithm is provided in Section IV. In Section V, numerical simulations comparing the existing state-of-the-art methods and DC-DistADMM algorithm in solving: (i) a ℓ_1 regularized distributed logistic regression problem and (ii) a ℓ_1 regularized distributed Huber loss minimization problem, are provided. The comparison results verify theoretical claims and provide a discussion on the effectiveness and suitability of the proposed algorithm. Finally, Section VI concludes this article.

II. DEFINITIONS, PROBLEM FORMULATION, AND ASSUMPTIONS

A. Definition, Notations, and Assumptions

In this section, definitions and notations that are used later in the analysis are presented. Detailed description of most of these notions can be found in [45], [46], and [47].

Definition 1 (Directed Graph): A directed graph $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices (or nodes) and $\mathcal{E}$ is a set of edges, which are ordered subsets of two distinct elements of $\mathcal{V}$. If an edge from $j \in \mathcal{V}$ to $i \in \mathcal{V}$ exists then it is denoted as $(i, j) \in \mathcal{E}$.

Definition 2 (Strongly Connected Graph): A directed graph is strongly connected if for any pair (i, j) , $i \neq j$, there is a directed path from node i to node j .

Definition 3 (Column-Stochastic Matrix): A matrix $M = [m_{ij}] \in \mathbb{R}^{n \times n}$ is called a column-stochastic matrix if $0 \leq m_{ij} \leq 1$ and $\sum_{i=1}^n m_{ij} = 1$ for all $1 \leq j \leq n$.

Definition 4 (Diameter of a Graph): The diameter of a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is the longest shortest directed path between any two nodes in the graph.

Definition 5 (In-Neighborhood): The set of in-neighbors of node $i \in \mathcal{V}$ is called the in-neighborhood of node i and is denoted by $\mathcal{N}_i^- = \{j \mid (i, j) \in \mathcal{E}\}$ not including the node i .

Definition 6 (Out-Neighborhood): The set of out-neighbors of node $i \in \mathcal{V}$ is called the out-neighborhood of node i and is denoted by $\mathcal{N}_i^+ = \{j \mid (j, i) \in \mathcal{E}\}$ not including the node i .

Definition 7 (Lipschitz Differentiability): A differentiable function $f: \mathbb{R}^p \rightarrow \mathbb{R}$ is called Lipschitz differentiable with constant $L > 0$, if the following inequality holds:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^p.$$

Definition 8 (Restricted Strongly Convex Function [48]): Given, $\tilde{x} \in \mathbb{R}^p$, $C \subseteq \mathbb{R}^p$, a differentiable convex function $f: \mathbb{R}^p \rightarrow \mathbb{R}$ is called restricted strongly convex with respect to $\tilde{x}$ on C with parameter $\sigma > 0$, if the following inequality holds:

$$\langle \nabla f(x) - \nabla f(\tilde{x}), x - \tilde{x} \rangle \geq \sigma \|x - \tilde{x}\|^2 \quad \forall x \in C.$$

Definition 9 (Diameter of a set): For a norm $\|\cdot\|$ and a set $K \subset \mathbb{R}^p$ define the diameter of K with respect to the norm $\|\cdot\|$, $\text{diam}_{\|\cdot\|}(K) := \sup_{x, y \in K} \|x - y\|$.

Definition 10 (Affine hull of a set): The affine hull $\text{Aff}(X)$ of a set X is the set of all affine combinations of elements of X

$$\text{Aff}(X) = \left\{ \sum_{i=1}^k \theta_i x_i \mid k > 0, x_i \in X, \theta_i \in \mathbb{R}, \sum_{i=1}^k \theta_i = 1 \right\}.$$

Definition 11 (Relative interior of a set): Let $X \subset \mathbb{R}^n$. A point $x \in X$ is in the relative interior of X , if X contains the intersection of a small enough ball centered at x with the $\text{Aff}(X)$, that is, there exists $r > 0$ such that

$$B_r(x) \cap \text{Aff}(X) := \{y \mid y \in \text{Aff}(X), \|y - x\| \leq r\} \subset X.$$

The set of all relative interior points of X is called its relative interior denoted by $\text{relint}(X)$.

Throughout this article, vectors are assumed to be column vectors unless stated otherwise. Each agent $i \in \mathcal{V}$ maintains two local primal variables $x_i^k \in \mathbb{R}^{(m+p)}$, $y_i^k \in \mathbb{R}^{(m+p)}$ and two dual variables $\lambda_i^k \in \mathbb{R}^{(m+p)}$, $\mu_i^k \in \mathbb{R}^m$ at all iterations $k \geq 0$ of the DC-DistADMM algorithm. $\mathbf{x}^k = [x_1^k \ x_2^k \ \dots \ x_n^k]^\top \in \mathbb{R}^{n(m+p)}$, $\mathbf{y}^k = [y_1^k \ y_2^k \ \dots \ y_n^k]^\top \in \mathbb{R}^{n(m+p)}$, $\boldsymbol{\lambda}^k = [\lambda_1^k \ \lambda_2^k \ \dots \ \lambda_n^k]^\top \in \mathbb{R}^{n(m+p)}$, and $\boldsymbol{\mu}^k = [\mu_1^k \ \mu_2^k \ \dots \ \mu_n^k]^\top \in \mathbb{R}^{nm}$, respectively. $\mathcal{D}$ denotes an upper bound on the diameter of the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. $\mathbb{R}_{\geq 0}^m$ denotes the non-negative subspace of $\mathbb{R}^m$, $\|\cdot\|$ denotes the 2-norm of the vector input unless stated otherwise and $\lceil \cdot \rceil$ denotes the least integer function or the ceiling function, defined as: given $x \in \mathbb{R}$, $\lceil x \rceil = \min\{m \in \mathbb{Z} \mid m \geq x\}$, where $\mathbb{Z}$ is the set of integers. Given a set S , the indicator function of set S is defined as

$$\mathcal{I}_S(x) = \begin{cases} 0 & \text{if } x \in S \\ +\infty & \text{otherwise.} \end{cases}$$

B. Problem Formulation

Consider the slack variable $\tilde{x}_s$ and rewrite the inequality constraint $D_i \tilde{x} \leq d_i$ as

$$\begin{bmatrix} D_i & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{x}_s \end{bmatrix} = d_i, \quad \tilde{x}_s \geq 0 \quad \forall i \in \mathcal{V}$$

where $\mathbf{I}$ is the identity matrix of appropriate dimension. Let $x = [\tilde{x}^\top \ \tilde{x}_s^\top]^\top \in \mathbb{R}^{(m+p)}$. Using the notation and the definition

of the indicator function of $\mathbb{R}_{\geq 0}^m$ problem (1) can be rewritten as

$$\begin{aligned} & \underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \sum_{i=1}^n [\tilde{f}_i(\tilde{x}) + \mathcal{I}_{\mathbb{R}_{\geq 0}^m}(\tilde{x}_s)] \\ & \text{subject to} \quad A_i x = b_i \quad \forall i \in \mathcal{V}, \quad \tilde{x} \in \bigcap_{i=1}^n \mathcal{X}_i \end{aligned} \quad (2)$$

where $A_i := [C_i + D_i \quad \mathbf{I}]^1$, $b_i := c_i + d_i$. Problem (2) is recast by creating local copies x_i for all $i \in \mathcal{V}$, of the global variable x and imposing the agreement of the solutions of all the agents via consensus constraint. This leads to an equivalent formulation of (2) as described as follows:

$$\begin{aligned} & \underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \sum_{i=1}^n [\tilde{f}_i(\tilde{x}_i) + \mathcal{I}_{\mathbb{R}_{\geq 0}^m}(\tilde{x}_{s_i})] \\ & \text{subject to} \quad A_i x_i = b_i, \quad \tilde{x}_i \in \mathcal{X}_i \quad \forall i \in \mathcal{V} \\ & \quad \quad \quad x_i = x_j \quad \forall i, j \in \mathcal{V} \end{aligned} \quad (3)$$

The constraints $\tilde{x}_i \in \mathcal{X}_i$ can be integrated into the objective function using the indicator functions of the sets $\mathcal{X}_i$, as

$$\begin{aligned} & \underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} \quad A_i x_i = b_i \quad \forall i \in \mathcal{V} \\ & \quad \quad \quad x_i = x_j \quad \forall i, j \in \mathcal{V} \end{aligned} \quad (4)$$

where $f_i(x_i) := \tilde{f}_i(\tilde{x}_i) + \mathcal{I}_{\mathbb{R}_{\geq 0}^m}(\tilde{x}_{s_i}) + \mathcal{I}_{\mathcal{X}_i}(\tilde{x}_i)$ and $\mathcal{I}_{\mathcal{X}_i}$ is the indicator function of set $\mathcal{X}_i$

Let, $\mathcal{C}_\eta := \left\{ \mathbf{y} = [y_1^\top \dots y_n^\top]^\top \in \mathbb{R}^{n(m+p)} \text{ such that} \right.$

$$\left. \|y_i - y_j\| \leq 2\eta, 1 \leq i, j \leq n \right\} \quad (5)$$

to be a set of vectors $\mathbf{y} \in \mathbb{R}^{n(m+p)}$ such that the norm of the difference between any two $(m+p)$ dimensional subvectors of $\mathbf{y}$ is less than η .

Using definition of set $\mathcal{C}_\eta$, (4) is equivalent to

$$\begin{aligned} & \underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} \quad A_i x_i = b_i \quad \forall i \in \mathcal{V} \\ & \quad \quad \quad \mathbf{x} = [x_1^\top x_2^\top \dots x_n^\top]^\top, \quad \mathbf{x} = \mathbf{y}, \quad \mathbf{y} \in \mathcal{C}_0. \end{aligned} \quad (6)$$

Problem (6) can be reformulated as

$$\begin{aligned} & \underset{x \in \mathbb{R}^p, y \in \mathbb{R}^q}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i) + \mathcal{I}_{\mathcal{C}_0}(\mathbf{y}) \\ & \text{subject to} \quad A_i x_i = b_i \quad \forall i \in \mathcal{V} \\ & \quad \quad \quad \mathbf{x} = [x_1^\top x_2^\top \dots x_n^\top]^\top, \quad \mathbf{x} = \mathbf{y} \end{aligned} \quad (7)$$

where $\mathcal{I}_{\mathcal{C}_0}$ is the indicator function of the set $\mathcal{C}_0$.

C. Standard ADMM Method

Here, a brief review of the standard ADMM method [13] utilized to solve optimization problems of the following form:

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

¹Without loss of generality here we assume $m_1 = m_2 = m$.

where $S \in \mathbb{R}^{n \times p}$, $T \in \mathbb{R}^{n \times q}$, and $c \in \mathbb{R}^n$ are provided. Consider, the augmented Lagrangian with the Lagrange multiplier λ and positive scalar γ

$$\begin{aligned} \mathcal{L}_\gamma(x, y, \lambda) &= f(x) + g(y) + \lambda^\top (Sx + Ty - c) \\ &\quad + \frac{\gamma}{2} \|Sx + Ty - c\|^2. \end{aligned} \quad (9)$$

The primal and dual variable updates in ADMM are given as follows: starting with the initial guess (x^0, y^0, λ^0) , at each iteration

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

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

$$\lambda^{k+1} = \lambda^k + \gamma(Sx^{k+1} + Ty^{k+1} - c). \quad (12)$$

III. PROPOSED DC-DISTADMM METHOD

Define $\mathbf{F}(\mathbf{x}) := \sum_{i=1}^n f_i(x_i)$, $\mathbf{A} := \mathbf{I} \otimes A_i$, $i = 1, \dots, n$, $\in \mathbb{R}^{nm \times n(m+p)}$, where $\mathbf{I}$ is $n \times n$ identity matrix and $\otimes$ denote the matrix Kronecker product, and $\mathbf{b} := [b_1^\top \dots b_n^\top]^\top \in \mathbb{R}^{nm}$. The Lagrangian function $\mathcal{L}$ for problem (7) is given by

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{y}, \lambda, \mu) &= \mathbf{F}(\mathbf{x}) + \mathcal{I}_{\mathcal{C}_0}(\mathbf{y}) + \lambda^\top (\mathbf{x} - \mathbf{y}) \\ &\quad + \mu^\top (\mathbf{A}\mathbf{x} - \mathbf{b}). \end{aligned} \quad (13)$$

Note that the standard augmented Lagrangian associated with (7) at any iteration k is

$$\begin{aligned} \mathcal{L}_\gamma(\mathbf{x}^k, \mathbf{y}^k, \lambda^k, \mu^k) &= \mathcal{L}(\mathbf{x}^k, \mathbf{y}^k, \lambda^k, \mu^k) + \frac{\gamma}{2} \|\mathbf{x}^k - \mathbf{y}^k\|^2 \\ &\quad + \frac{\gamma}{2} \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|^2. \end{aligned} \quad (14)$$

Based on ADMM iterations (10) and (12), the primal and dual updates corresponding to augmented Lagrangian (14) are

$$\begin{aligned} \mathbf{x}^{k+1} &= \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \mathbf{F}(\mathbf{x}) + \lambda^{k\top} (\mathbf{x} - \mathbf{y}^k) + \mu^{k\top} (\mathbf{A}\mathbf{x} - \mathbf{b}) \right. \\ &\quad \left. + \frac{\gamma}{2} \|\mathbf{x} - \mathbf{y}^k\|^2 + \frac{\gamma}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \right\}. \end{aligned} \quad (15)$$

Note that the abovementioned $\mathbf{x}^{k+1}$ update can be obtained in a decentralized manner by computation at each agent as follows:

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} \left\{ f_i(x_i) + \lambda_i^{k\top} (x_i - y_i^k) + \mu_i^{k\top} (A_i x_i - b_i) \right. \\ &\quad \left. + \frac{\gamma}{2} \|x_i - y_i^k\|^2 + \frac{\gamma}{2} \|A_i x_i - b_i\|^2 \right\} \quad \forall i \in \mathcal{V}. \end{aligned} \quad (16)$$

The update for the primal variable $\mathbf{y}$ is given as

$$\begin{aligned} \bar{\mathbf{y}}^{k+1} &= \underset{\mathbf{y}}{\operatorname{argmin}} \\ &\quad \times \left\{ \mathcal{I}_{\mathcal{C}_0}(\mathbf{y}) + \lambda^{k\top} (\mathbf{x}^{k+1} - \mathbf{y}) + \frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{y}\|^2 \right\} \\ &= \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ \mathcal{I}_{\mathcal{C}_0}(\mathbf{y}) + \frac{\gamma}{2} \left\| \mathbf{x}^{k+1} - \mathbf{y} + \frac{1}{\gamma} \lambda^k \right\|^2 \right\}. \end{aligned} \quad (17)$$

The $\bar{\mathbf{y}}^{k+1}$ update in (17) is the projection of $\mathbf{x}^{k+1} + \frac{1}{\gamma} \lambda^k$ on the set $\mathcal{C}_0$. It can be verified (see [49], Appendix B) that

$$\begin{aligned} \bar{\mathbf{y}}^{k+1} &= \left[\bar{y}^{k+1\top} \dots \bar{y}^{k+1\top} \right]^\top \in \mathbb{R}^{n(m+p)}, \text{ where} \\ \bar{y}^{k+1} &= \frac{1}{n} \sum_{i=1}^n \left[x_i^{k+1} + \frac{1}{\gamma} \lambda_i^k \right] \end{aligned} \quad (18)$$

is the solution of the update (17). However, obtaining the optimal solution (the exact average of the local variables) in a distributive manner over a directed network where each agent only has access to its own local information is a challenge. Hence, the exact consensus constraint is relaxed to a requirement of a η_{k+1} -closeness among the variables of all the agents, i.e., the solution $\mathbf{y}^{k+1}$ is allowed us to lie in the set $\mathcal{C}_{\eta_{k+1}}$ and satisfy $\|y_i^{k+1} - y_j^{k+1}\| \leq 2\eta_{k+1}, \forall i, j \in \mathcal{V}$. The parameter η_{k+1} can be interpreted as a specified tolerance on the quality of consensus among the agent variables and can be chosen appropriately to find a corresponding near optimal solution of (17). A distributed finite-time terminated approximate consensus protocol is employed to find an *inexact* solution of (17), such that the obtained solution $\mathbf{y}^{k+1} \in \mathcal{C}_{k+1}$. The “approximate” consensus protocol is discussed in detail in the following section (Section III-A).

After the primal variables, each agent updates the dual variables in the following manner:

$$\lambda_i^{k+1} = \lambda_i^k + \gamma(x_i^{k+1} - y_i^{k+1}) \quad \forall i \in \mathcal{V} \quad (19)$$

$$\mu_i^{k+1} = \mu_i^k + \gamma(A_i x_i^{k+1} - b_i) \quad \forall i \in \mathcal{V}. \quad (20)$$

Define $\bar{\lambda}_i^{k+1}$ based on the solution $\bar{\mathbf{y}}^{k+1}$

$$\bar{\lambda}_i^{k+1} = \lambda_i^k + \gamma(x_i^{k+1} - \bar{y}_i^{k+1}) \quad \forall i \in \mathcal{V}. \quad (21)$$

Next, the ε -consensus protocol is described.

A. Finite-Time ε -Consensus Protocol

Here, a finite-time “approximate” consensus protocol called the ε -consensus protocol is presented. The protocol was first proposed in an earlier work [50] by the authors. Consider, a set of n agents connected via a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Every agent i has a vector $u_i^0 \in \mathbb{R}^p$. Let $\tilde{u} = \frac{1}{n} \sum_{i=1}^n u_i^0$ denote the average of the vectors. The objective is to design a distributed protocol such that the agents are able to compute an approximate estimate of $\tilde{u}$ in finite time. This approximate estimate is parametrized by a tolerance ε that can be chosen arbitrarily small to make the estimate as precise as needed. To this end, the agents maintain state variables $u_i^k \in \mathbb{R}^p, v_i^k \in \mathbb{R}$ that undergo the following update: for $k \geq 0$

$$u_i^{k+1} = p_{ii} u_i^k + \sum_{j \in \mathcal{N}_i^-} p_{ij} u_j^k \quad (22)$$

$$v_i^{k+1} = p_{ii} v_i^k + \sum_{j \in \mathcal{N}_i^-} p_{ij} v_j^k \quad (23)$$

$$w_i^{k+1} = \frac{1}{v_i^{k+1}} u_i^{k+1} \quad (24)$$

where $w_i^0 = u_i^0, v_i^0 = 1$ for all $i \in \mathcal{V}$ and $\mathcal{N}_i^-$ denotes the set of in-neighbors of agent i . The updates (22)–(24) are based on the push-sum (or ratio consensus) updates (see [51] or [52]). The following assumption on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and weight matrix $\mathcal{P}$ is made.

Assumption 1: The directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is strongly-connected. Let the weighted adjacency matrix $\mathcal{P} = [p_{ij}]$, associated with the digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, be column-stochastic.

Note that $\mathcal{P}$ being a column stochastic matrix allows for a distributed synthesis of the consensus protocol. The variable $w_i^k \in \mathbb{R}^p$ is an estimate of $\tilde{u}$ with each agent i at any iteration k . It is established in prior work that the vector estimates w_i^k converge to the average $\tilde{u}$ asymptotically.

Theorem 1: Let Assumption 1 hold. Let $\{w_i^k\}_{k \geq 0}$ be the sequence generated by (24) at each agent $i \in \mathcal{V}$. Then w_i^k asymptotically converges to $\tilde{u} = \frac{1}{n} \sum_{i=1}^n u_i^0$ for all $i \in \mathcal{V}$, i.e.,

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

Proof: Refer [50], [51] for the proof. $\square$

The ε -consensus protocol is a distributed algorithm to determine when the agent states w_i^k , for all $i \in \mathcal{V}$ are ε -close to each other and, hence, from Theorem 1, ε -close to $\tilde{u}$. To this end, each agent maintains a scalar value R_i termed as the radius of agent i . The motivation behind maintaining such a radius variable is as follows: consider an open ball at iteration k that encloses all the agent states, w_i^k , for all $i \in \mathcal{V}$, with a minimal radius (the existence of such a ball can be shown due to bounded nature of the updates (22)–(24), see [53, Lemma 4.2]). Theorem 1 guarantees that all the agent states w_i^k converge to a single vector, which implies that the minimal ball enclosing all the states will also shrink with iteration k and eventually the radius will become zero. The radius variable R_i is designed to track the radius of this minimal ball. Starting at an iteration s , the radius $R_i^k(s)$ is updated to $R_i^{k+1}(s)$ as follows: for all $k = 0, 1, 2, \dots$, with $R_i^0(s) := 0$

$$R_i^{k+1}(s) = \max_{j \in \mathcal{N}_i^-} \{ \|w_i^{s+k+1} - w_j^{s+k}\| + R_j^k(s) \} \quad (25)$$

for all $i \in \mathcal{V}$. Denote, $\mathcal{B}(w_i^{s+k}, R_i^k(s))$ as the ball of radius $R_i^k(s)$ centered at w_i^{s+k} . It is established in [50] that, after $\mathcal{D}$ number of iterations of the update (25) the ball $\mathcal{B}(w_i^{s+\mathcal{D}}, R_i^{\mathcal{D}}(s))$ encloses the states w_i^s of all the agents $i \in \mathcal{V}$.

Lemma 1: Let $\{w_i^k\}_{k \geq 0}$ be the sequence generated by (24) at each agent $i \in \mathcal{V}$. Given, $s \geq 0$, let $R_i^k(s)$ be updated as in (25) for all $k \geq 0$ and $i \in \mathcal{V}$. Under Assumption 1

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

Proof: See [49], Appendix C for proof. $\square$

$$\text{Let, } \hat{R}_i^m := R_i^{\mathcal{D}}(m\mathcal{D}), m = 0, 1, 2, \dots \quad (26)$$

where $R_i^k(s)$ follows the update rule (25) for any $s \geq 0$. The next result establishes the fact that the sequence of radii $\{\hat{R}_i^{\mathcal{D}}(m\mathcal{D})\}_{m \geq 0}$, for $i \in \mathcal{V}$ converges to zero as $m \rightarrow \infty$.

Theorem 2: Let $\{w_i^k\}_{k \geq 0}, \{\hat{R}_i^m\}_{m \geq 0}$ be the sequences generated by (24) and (26), respectively. Under Assumption 1

$$\lim_{m \rightarrow \infty} \hat{R}_i^m = 0, \text{ for all } i \in \mathcal{V}.$$

Furthermore, $\lim_{m \rightarrow \infty} R_i^m = 0$ if and only if $\lim_{m \rightarrow \infty} \max_{i, j \in \mathcal{V}} \|w_i^{m\mathcal{D}} - w_j^{m\mathcal{D}}\| = 0$.

Proof: See [49], Appendix D for proof. $\square$

Theorem 2 gives a criterion for termination of the consensus iterations (22)–(24) by utilizing the radius updates at each agent $i \in \mathcal{V}$ given by (25). In particular, by tracking the radii $\hat{R}_i^m, m = 0, 1, 2, \dots$ each agent can determine an estimate of the radius of the minimal ball enclosing all the agent states (Lemma 1) distributively. Furthermore, monitoring the value of the $\hat{R}_i^m, m = 0, 1, 2, \dots$ each agent can have the knowledge of the state of consensus among all the agents. A protocol to determine ε -consensus among the agents is given in Algorithm 1. Algorithm 1 is initialized with $w_i^0 = u_i^0, v_i^0 = 1$ and $R_i^0 = 0$ respectively for all $i \in \mathcal{V}$. Each agent follows rules (22)–(24) and (25) to update its state and radius variables, respectively.

Algorithm 1: Finite-time ε -consensus protocol at each node $i \in \mathcal{V}$ [50].

Input:
 Pre-specified tolerance $\varepsilon > 0$;
 Diameter upper bound $\mathcal{D}$;
Initialize:
 $w_i^0 = u_i^0$; $v_i^0 = 1$; $R_i^0 = 0$;
 $m := 1$
Repeat for $k = 0, 1, 2, \dots$
 /* consensus updates (22)–(24) */
 $u_i^{k+1} = p_{ii}u_i^k + \sum_{j \in \mathcal{N}_i^-} p_{ji}u_j^k$
 $v_i^{k+1} = p_{ii}v_i^k + \sum_{j \in \mathcal{N}_i^-} p_{ji}v_j^k$
 $w_i^{k+1} = \frac{1}{v_i^{k+1}}u_i^{k+1}$
 /* radius update (25) */
 $R_i^{k+1} = \max_{j \in \mathcal{N}_i^-} \{ \|w_i^{k+1} - w_j^k\| + R_j^k \}$
if $k = m\mathcal{D} - 1$ **then**
 $\hat{R}_i^{m-1} = R_i^{k+1}$;
if $\hat{R}_i^{m-1} < \varepsilon$ **then**
break ; // ε -consensus achieved
else
 $R_i^{k+1} = 0$;
 $m = m + 1$;
end

R_i^k is reset at each iteration of the form $k = m\mathcal{D}$, $m = 1, 2, \dots$ to have a value equal to 0. The sequence of radii $\{\hat{R}_i^m\}_{m \geq 0}$ is determined by setting the value $\hat{R}_i^{m-1}$ equal to R_i^k for all iterations of the form $k = m\mathcal{D}$, $m = 1, 2, \dots$, i.e., $\hat{R}_i^{m-1} = R_i^{m\mathcal{D}}$ for all $m \geq 1$. From Lemma 1, the ball $\mathcal{B}(w_i^{m\mathcal{D}}, \hat{R}_i^{m-1})$ will contain all estimates $w_i^{(m-1)\mathcal{D}}$. This ball is the estimate of the minimal ball enclosing all agent states $w_i^{(m-1)\mathcal{D}}$. Therefore, as a method to detect ε -consensus, at every iteration of the form $m\mathcal{D}$, for $m = 1, 2, \dots$, $\hat{R}_i^{m-1}$ is compared to the parameter ε , if $\hat{R}_i^{m-1} < \varepsilon$ then all the agent states $w_i^{(m-1)\mathcal{D}}$ were ε -close to $\tilde{u}$ (from Theorem 1) and the iterations (22)–(24) are terminated. Proposition 1 establishes that the ε -consensus converges in finite number of iterations.

Proposition 1: Under the Assumption 1, ε -consensus is achieved in finite number of iterations at each agent $i \in \mathcal{V}$.

Proof: Note, $\hat{R}_i^m \rightarrow 0$ as $m \rightarrow \infty$. Thus, given $\varepsilon > 0$, $i \in \mathcal{V}$ there exists finite $\bar{k}_i^\varepsilon$ such that for $m \geq \bar{k}_i^\varepsilon$, $\hat{R}_i^m < \varepsilon$. $\square$

Note, that the radius estimates $\hat{R}_i^m$ can be different for some of the agents. Thus, the detection of ε -consensus can happen at different iterations for some nodes. In order to, have a global detection, each agent can generate “converged flag” indicating its own detection. Such a flag signal can then be combined by means of a distributed one-bit consensus updates (see [50]), thus allowing the agents to achieve global ε -consensus.

B. y Variable Updates in DC-DistADMM

The finite-time ε -consensus protocol discussed above is utilized to determine an inexact solution to the update (17). At any iteration $k \geq 0$ of the DC-DistADMM algorithm, each agent $i \in \mathcal{V}$ runs an ε -consensus protocol with the tolerance $\varepsilon = \eta_{k+1}$ and the following initialization:

$$u_i^0 = x_i^{k+1} + \frac{1}{\gamma}\lambda_i^k, \quad v_i^0 = 1, \quad \text{and} \quad w_i^0 = u_i^0. \quad (27)$$

Algorithm 2: DC-DistADMM Algorithm.

Input:
 choose $\gamma > 0$, $\{\eta_k\}_{k \geq 0}$
Initialize:
 $x_i^0 \in \mathbb{R}^{(m+p)}$, $y_i^0 \in \mathbb{R}^{(m+p)}$, $\lambda_i^0 = \mathbf{0}_{m+p} \in \mathbb{R}^{(m+p)}$, $\mu_i^0 \in \mathbb{R}^m$, $\forall i \in \mathcal{V}$
Repeat for $k = 0, 1, 2, \dots$
for $i = 1, 2, \dots, n$, (*In parallel*) **do**
 $x_i^{k+1} = \underset{x_i}{\operatorname{argmin}} \left\{ f_i(x_i) + \lambda_i^{k\top} (x_i - y_i^k) + \frac{\gamma}{2} \|x_i - y_i^k\|^2 + \mu_i^{k\top} (A_i x_i - b_i) + \frac{\gamma}{2} \|A_i x_i - b_i\|^2 \right\}$,
 $y_i^{k+1} \xleftarrow{\eta_{k+1} \text{-consensus protocol}} (x_i^{k+1} + \frac{1}{\gamma}\lambda_i^k)$
 $\lambda_i^{k+1} = \lambda_i^k + \gamma(x_i^{k+1} - y_i^{k+1})$
 $\mu_i^{k+1} = \mu_i^k + \gamma(A_i x_i^{k+1} - b_i)$
end
until a stopping criterion is met

From Proposition 1, there exists finite t_k such that $\forall i, j \in \mathcal{V}$

$$\|w_i^{t_k} - w_j^{t_k}\| \leq \eta_{k+1}, \quad \text{and} \quad \|w_i^{t_k} - \bar{y}^{k+1}\| \leq \eta_{k+1} \quad (28)$$

where $\bar{y}^{k+1} = \frac{1}{n} \sum_{i=1}^n [x_i^{k+1} + \frac{1}{\gamma}\lambda_i^k]$. The variable y_i^{k+1} of agent i is then set equal to the consensus variable $w_i^{t_k}$, i.e., $y_i^{k+1} = w_i^{t_k}$. Therefore, the obtained vector $\mathbf{y}^{k+1} = [y_1^{k+1}, \dots, y_n^{k+1}]^\top \in \mathbb{R}^{n(m+p)}$ is an *inexact* solution to the update (17) (see [49], Appendix E). Furthermore, $\mathbf{y}^{k+1}$ is $\sqrt{n}\eta_{k+1}$ -close to the exact solution $\bar{\mathbf{y}}^{k+1}$, i.e.,

$$\mathbf{y}^{k+1} = \bar{\mathbf{y}}^{k+1} + \mathbf{e}^{k+1}, \quad \text{with} \quad \|\mathbf{e}^{k+1}\| \leq \sqrt{n}\eta_{k+1}. \quad (29)$$

The DC-DistADMM algorithm is presented in Algorithm 2.

IV. CONVERGENCE RESULTS FOR DC-DistADMM

In this section, convergence result for the proposed DC-DistADMM algorithm is presented. Let $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$, ${}^n\mathbb{R}_{\geq 0}^m := \underbrace{\mathbb{R}_{\geq 0}^m \times \dots \times \mathbb{R}_{\geq 0}^m}_{n \text{ times}}$ and $x^* \in \mathbb{R}^{(m+p)}$ denote an optimal solution of problem (1). Throughout the rest of this article, the following assumptions hold.

Assumption 2: Each $\mathcal{X}_i$ is a closed bounded convex set with diameter $\operatorname{diam}_{\|\cdot\|}(\mathcal{X}_i) = \mathcal{M}_i$.

Note that Assumption 2 can be relaxed if the set of optimal solutions of the original distributed problem (1) is bounded (see [54]); in this case, the existence of a bound on the local variables can be inferred from the bound on optimal solutions. Here, this route is not taken, as in many multiagent distributed optimization applications the variables are required to remain within specified bounds. Assumption 2 is motivated by the constraint set requirement in many practical applications.

Assumption 3: The function $f_i, i \in \{1, \dots, n\}$, is a proper closed convex function, which is not necessarily differentiable.

Assumption 4: There exists a saddle point $(\mathbf{x}^*, \mathbf{y}^*, \lambda^*, \mu^*)$ for the Lagrangian function $\mathcal{L}$ defined in (13), i.e., for all $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n(m+p)} \times \mathbb{R}^{n(m+p)}$, $(\lambda, \mu) \in \mathbb{R}^{n(m+p)} \times \mathbb{R}^{nm}$, where

$$\mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \lambda, \mu) \leq \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \lambda^*, \mu^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}, \lambda^*, \mu^*).$$

Note that since (7) is a convex optimization problem, the existence of dual optimal solutions is guaranteed if a constraint qualification condition like Slater's CQ holds for problem (7) [47]. Note that the indicator function of a set is a convex but not differentiable function. For further analysis of

the DC-DistADMM algorithm consider the following relation: for $x_1, x_2, x_3, x_4 \in \mathbb{R}^p$

$$(x_1 - x_2)^\top (x_3 - x_4) = \frac{1}{2} (\|x_1 - x_4\|^2 - \|x_1 - x_3\|^2) + \frac{1}{2} (\|x_2 - x_3\|^2 - \|x_2 - x_4\|^2). \quad (30)$$

Under Assumption 3, the first-order optimality and primal feasibility conditions for (7) are

$$-\lambda_i^* - A_i^\top \mu_i^* \in \partial f_i(x_i^*), \quad i = 1, 2, \dots, n \quad (31)$$

$$\lambda^* \in \partial \mathcal{I}_{C_0}(\mathbf{y}^*) \quad (32)$$

$$x_i^* = y_i^*, \quad i = 1, 2, \dots, n \quad (33)$$

$$A_i x_i^* - b_i = 0, \quad i = 1, 2, \dots, n \quad (34)$$

where $\partial f_i(x_i^*)$ and $\partial \mathcal{I}_{C_0}(\mathbf{y}^*)$ are the set of all the subgradients of f_i at x_i^* , and $\mathcal{I}_{C_0}$ at $\mathbf{y}^*$, respectively. Similarly, for the DC-DistADMM updates (16), and (17) by the first-order optimality condition it follows that, for all $i \in \{1, \dots, n\}$

$$-(\lambda_i^k + A_i^\top \mu_i^k + \gamma(x_i^{k+1} - y_i^k) + \gamma A_i^\top (A_i x_i^{k+1} - b_i)) \in \partial f_i(x_i^{k+1}) \quad (35)$$

$$-(\lambda_i^{k+1} + \gamma(y_i^{k+1} - y_i^k) + A_i^\top \mu_i^{k+1}) \in \partial f_i(x_i^{k+1}) \quad (36)$$

$$\gamma(\mathbf{x}^{k+1} - \bar{\mathbf{y}}^{k+1}) + \lambda^k \in \partial \mathcal{I}_{C_0}(\bar{\mathbf{y}}^{k+1}) \Rightarrow \bar{\lambda}^{k+1} \in \partial \mathcal{I}_{C_0}(\bar{\mathbf{y}}^{k+1}) \quad (37)$$

where $\partial \mathcal{I}_{C_0}(\bar{\mathbf{y}}^{k+1})$ is the set of all subgradients of $\mathcal{I}_{C_0}$ at $\bar{\mathbf{y}}^{k+1}$. The relation (36) results from combining (35) with the update (19), and in (37) (29) is employed.

Lemma 2: Let $\{\mathbf{x}^k\}_{k \geq 1}$, $\{\mathbf{y}^k\}_{k \geq 1}$, $\{\lambda^k\}_{k \geq 1}$, and $\{\mu^k\}_{k \geq 1}$, be the sequences generated by Algorithm 2. Let $\{\eta_k\}_{k \geq 1}$ be the sequence of tolerances in Algorithm 2. Under Assumptions 1 and 2, there exists $\mathbf{Q}$, independent of k , such that

$$\|\lambda^k\| \leq k\mathbf{Q}, \quad \text{for all } k.$$

Proof: See [49], Appendix F for proof. $\square$

Lemma 3 (Iteration complexity of the $\mathbf{y}^{k+1}$ update (28)): Under Assumptions 1 and 2 at iteration k of Algorithm 2,

after $\bar{t}_k := \left\lceil \frac{\log(\frac{1}{\eta_{k+1}})}{-\log \alpha} + \frac{\log(\frac{8((\mathcal{M} + \|\mathbf{x}^0\|)n + k\sqrt{n}\mathbf{Q}/\gamma))}{\beta}}{-\log \alpha} \right\rceil$ iterations of the consensus protocol [updates (22)–(24)] with the initial condition, $u_i^0 = x_i^{k+1} + \frac{1}{\gamma}\lambda_i^k$, $v_i^0 = 1$, for all $i \in \mathcal{V}$ and $\varepsilon = \eta_{k+1}$ it follows that

$$\|y_i^{k+1} - \bar{y}_i^{k+1}\| \leq \eta_{k+1} \quad \forall i \in \mathcal{V}.$$

Here, $\mathcal{M} := \max_{1 \leq i \leq n} \mathcal{M}_i$, η_{k+1} denotes the inaccuracy introduced in solving relaxed version of problem (17), $\mathbf{Q}$ is a finite constant as given in Lemma 2, and $\beta > 0$, $\alpha \in (0, 1)$ are parameters of the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ satisfying $\beta \geq \frac{1}{n^n}$, $\alpha \leq (1 - \frac{1}{n^n})$.

Proof: See [49], Appendix G for proof. $\square$

Let the sequence of accuracy tolerances $\{\eta_k\}_{k \geq 1}$ in Algorithm 2 be such that

$$\text{for all } k \geq 0, \eta_{k+1} \leq \eta_k, \sum_{k=1}^{\infty} k\eta_k < \infty. \quad (38)$$

Some examples of sequences that satisfy (38) include

$$\text{i) } \eta_k = \frac{1}{k^{2+q}}, q > 0, \text{ ii) } \eta_k = \rho^k, \rho \in (0, 1), \text{ iii) } \eta_k = \frac{\rho^k}{k!}.$$

In the following, the global convergence of the proposed DC-DistADMM algorithm to a solution of problem (7) is established.

Theorem 3 (Convergence of iterates generated by DC-DistADMM algorithm to an optimal solution): Let $\{\mathbf{x}^k\}_{k \geq 1}$, $\{\mathbf{y}^k\}_{k \geq 1}$, $\{\lambda^k\}_{k \geq 1}$, $\{\mu^k\}_{k \geq 1}$, be the sequences generated by Algorithm 2. Let the consensus tolerance sequence $\{\eta_k\}_{k \geq 1}$ satisfy (38). Under Assumptions 1–4, $(\mathbf{x}^k, \mathbf{y}^k, \lambda^k, \mu^k)$ converges to a solution $(\mathbf{x}^\infty, \mathbf{y}^\infty, \lambda^\infty, \mu^\infty)$ of (7), i.e., $\mathbf{F}(\mathbf{x}^\infty) = \mathbf{F}(\mathbf{x}^*)$, $\mathbf{y}^\infty \in \mathcal{C}_0$, $\mathbf{x}^\infty = \mathbf{y}^\infty$, and $\mathbf{A}\mathbf{x}^\infty = \mathbf{b}$.

Proof: See [49], Appendix H for proof. $\square$

Next two estimates of rate of convergence for the proposed DC-DistADMM algorithm are provided.

A. Sublinear Rate of Convergence

Here, the convergence of the proposed DC-DistADMM algorithm for the case when individual functions f_i are convex but not necessarily differentiable is analyzed.

Theorem 4 (Sublinear rate of convergence): Let $\{\mathbf{x}^k\}_{k \geq 1}$, $\{\mathbf{y}^k\}_{k \geq 1}$, $\{\lambda^k\}_{k \geq 1}$, and $\{\mu^k\}_{k \geq 1}$, be the sequences generated by Algorithm 2. Let $\eta_k = \frac{1}{k^{2+q}}$, $q \in (0, 1)$. Let Assumptions 1–4 hold, then for all $k \geq 1$

$$\mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) = O(1/k), \quad \|\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k\| = O(1/k)$$

$$\text{and, } \|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\| = O(1/k)$$

where $\hat{\mathbf{x}}^k := \frac{1}{k} \sum_{s=0}^{k-1} \mathbf{x}^{s+1}$, $\hat{\mathbf{y}}^k := \frac{1}{k} \sum_{s=0}^{k-1} \mathbf{y}^{s+1}$.

Proof: See Appendix A. $\square$

Theorem 4 establishes that the objective function evaluated at the ergodic average of the optimization variables obtained by the proposed DC-DistADMM algorithm converges to the optimal value. In particular, the objective function value evaluated at the ergodic average converges to the optimal value at a rate of $O(1/k)$.

Remark 1: Let, $\hat{x}_i^k = \frac{1}{k} \sum_{s=0}^{k-1} x_i^s$ and $\hat{y}_i^k = \frac{1}{k} \sum_{s=0}^{k-1} y_i^s$ denote the ergodic averages of the variables x_i^k and y_i^k . Since, $\|\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k\| = O(1/k)$ it implies that for all $i \in \mathcal{V}$, $\|\hat{x}_i^k - \hat{y}_i^k\| = O(1/k)$. Furthermore, since, for all $k \geq 0$ and $i, j \in \mathcal{V}$, $\|y_i^k - y_j^k\| \leq \eta_k$ [see (28)] it implies that $\|\hat{y}_i^k - \hat{y}_j^k\| \leq \frac{1}{k} \sum_{s=0}^{k-1} \|y_i^s - y_j^s\| \leq \frac{1}{k} \sum_{s=0}^{k-1} \eta_s \leq \frac{\zeta(2+q)}{k}$. Thus, $\|\hat{y}_i^k - \hat{y}_j^k\| = O(1/k)$. Therefore, for all $i, j \in \mathcal{V}$

$$\|\hat{x}_i^k - \hat{x}_j^k\| = O(1/k) \text{ and } \lim_{k \rightarrow \infty} \|\hat{x}_i^k - \hat{x}_j^k\| = 0.$$

Thus, in practice, the DC-DistADMM algorithm can be implemented with an additional variable $\hat{x}_i^k$ tracking the ergodic average of x_i^k at each agent $i \in \mathcal{V}$ to achieve a consensual solution at the $O(1/k)$ rate of convergence given in Theorem 4 in a fully distributed manner. Note, that the variable $\hat{x}_i^k$ is computed locally by each agent without any additional cost of communication and a minor addition in cost of computation.

Remark 2: By Lemma 3 the total number of communication iterations performed until iteration k of the DC-DistADMM algorithm in Theorem 4, is upper bounded by $\mathcal{K} := \sum_{s=1}^k \bar{t}_s := O(k \log k)$. The communication complexity is within a factor of $\log k$ of the optimal lower bound $O(k)$.

B. Geometric rate of Convergence

Here, a geometric rate of convergence for the proposed DC-DistADMM algorithm under the following assumption is established.

Assumption 5: Each function $\tilde{f}_i$ is Lipschitz differentiable with constant L_{f_i} and restricted strongly convex with respect to the optimal solution $\mathbf{x}^*$ on $\mathcal{X}_i$ with parameter $\sigma_i > 0$.

Remark 3: Under Assumption 5 problem (1) has a unique optimal solution. However, Assumption 5 is less restrictive than the standard global strong convexity assumption and makes the analysis presented here applicable to a bigger class of functions [55]. For example, the widely used logistic regression objective function is restricted strongly convex but not globally strongly convex [56].

Theorem 5 (Geometric rate of convergence): Let $\{\mathbf{x}^k\}_{k \geq 1}$, $\{\mathbf{y}^k\}_{k \geq 1}$, $\{\lambda^k\}_{k \geq 1}$, and $\{\mu^k\}_{k \geq 1}$, be the sequences generated by Algorithm 2. Let Assumptions 1, 2, 4, and 5 hold. Let $\Delta := (1 - \frac{1}{\delta}) \min\{1, \nu_{\min}(\mathbf{A}\mathbf{A}^\top)\}$ for any $\delta \in (1, 1 + \frac{2\sigma\gamma}{L^2 + \gamma^2}]$, where $L := \max_{1 \leq i \leq n} L_{f_i}$, $\sigma := \min_{1 \leq i \leq n} \sigma_i$, and $\nu_{\min}(\mathbf{A}\mathbf{A}^\top)$ is the minimum eigenvalue of $\mathbf{A}\mathbf{A}^\top$. Let $\eta_k = \rho^k$, where $\rho \in [\frac{1}{1+\Delta}, 1)$. Let $x_i^k \in \text{relint}(\mathcal{X}_i \times \mathbb{R}_{\geq 0}^m)$, $\forall i \in \mathcal{V}$, $k \geq 0$. Then, the agent solution residual $s_k := \frac{\gamma}{2} \|\mathbf{x}^k - \mathbf{x}^*\|^2 + \frac{1}{2\gamma} \|\lambda^k - \lambda^*\|^2 + \frac{1}{2\gamma} \|\mu^k - \mu^*\|^2$ has the following relation: for any $K \geq 0$ and $\epsilon > 0$

$$s_K \leq \Upsilon \rho^K + O(\epsilon)$$

where Υ is a finite constant defined in (66).

Proof: See Appendix B. $\square$

Remark 4: Although, Theorem 5 provides a tight bound on the requirement of ρ , in practice the tolerance sequence parameter ρ can be chosen from the interval $[\frac{2}{3}, 1)$ (see [49], Remark 4 for more details), which does not require the knowledge of the restricted strong convexity parameter σ_i of the individual functions and the optimal solution $\mathbf{x}^*$.

Remark 5: By Lemma 3 the total number of communication iterations performed until iteration k of the DC-DistADMM algorithm in Theorem 5, is upper bounded by $\mathcal{K} := \sum_{s=1}^k \bar{t}_s := O(k^2)$. Thus, compared to Theorem 4 the improved rate of convergence leads to an increase in the number of communication iterations (Remark 2). Moreover, compared to algorithms utilizing multiple communication steps in the literature [12], [57], [58], [59], [60], [61], [62], [63] the DC-DistADMM algorithm has the same communication complexity. In particular, methods in [57], [58], [59] have communication complexity, $O(k^2)$, in getting a geometric rate. Schemes proposed in [12], [60], [61], [62], [63] have the same, $O(k \log k)$, communication complexity in getting a $O(1/k)$ rate of convergence.

Remark 6: The geometric rate of convergence for DC-DistADMM algorithm does not follow from the existing centralized results (see [64], [65]) as problem (7) does not satisfy the assumptions used in these works. In particular, the function $\mathcal{I}_{c_0}(\mathbf{y})$ is not differentiable and does not meet the requirements of being global Lipschitz differentiable, twice continuously differentiable and strongly convex used in [64] and [65]. Furthermore, unlike [64], [65], the $\mathbf{y}$ variable update in the ADMM scheme of DC-DistADMM algorithm in an inexact manner is solved via the ϵ -consensus protocol that adds additional complexities in the analysis. These reasons have motivated us to present the analysis of the DC-DistADMM given in Theorem 5.

V. NUMERICAL SIMULATIONS

In this section, three simulation studies are presented for the proposed DC-DistADMM algorithm. First, a performance comparison of the DC-DistADMM algorithm with two existing algorithms in the literature for solving constrained distributed optimization problems, [30] and [24], in solving a distributed ℓ_1

TABLE I
ALGORITHM PARAMETERS

ALGORITHM	Parameter
DC-DistADMM	tolerance $\eta_k = 0.01, 1/k^{2.1}, 0.75^k, \gamma = 10$
CDA1 [30]	step-size $\alpha = 0.01$
CDA2 [24]	step-size $\alpha(k) = 1/k^{1.2}$
EXTRAPush [10]	step-size $\alpha = 0.0009$
PushPull [11]	step-size $\alpha = 0.05$
PushDIGing [66]	step-size $\alpha = 0.001$
subgradientPush [8]	step-size $\alpha = 0.005$
FDGD [62]	step-size $\alpha = 0.01, \beta(k) = k/(k+3)$
nearDGD [57]	step-size $\alpha = 0.01$

regularized logistic regression with a local linear equality and set (norm-ball) constraints is presented.

Second, a performance comparison of the DC-DistADMM algorithm is provided for solving an unconstrained ℓ_1 regularized Huber loss minimization problem with the existing state-of-the-art unconstrained distributed optimization algorithms on directed graphs. The algorithms used for comparison with the proposed DC-DistADMM algorithm are the following:

- EXTRAPush [10];
- PushPull [11];
- PushDIGing [66];
- the subgradientPush [8] algorithm.

Third, a comparison between DC-DistADMM algorithm and two existing algorithms utilizing multiple communication steps, [57] and [62] on solving the unconstrained distributed least squares problem is presented.

A network of 100 agents connected via: (i) an undirected graph generated using an Erdos-Renyi model [67] with a connectivity probability of 0.3 in simulation studies one and three and (ii) a directed graph generated using an Erdos-Renyi model with a connectivity probability of 0.2 in the simulation study two is considered. The weight matrices for the various algorithms are chosen using the equal neighbor model [68]. To provide a comparison, solution residual plots against the total (computation + communication) iteration counts for all the algorithms unless stated otherwise are presented. Furthermore, a comparison based on the CPU time (the amount of time required by a computer (processor) to execute the instructions) between DC-DistADMM and the other algorithms is provided.

All the numerical examples in this section are implemented in MATLAB, and run on a desktop computer with 16 GB RAM and an Intel Core i7 processor running at 1.90 GHz. The parameters used in the simulations for all the algorithms are reported in Table I. In choosing the step-sizes of the algorithms in Table I, we followed the approach of hand-optimizing the hyperparameters that produce a good performance for these algorithms while maintaining the stability of the algorithmic estimates for the class of problems under consideration.

A. Performance on Constrained Optimization Problem

Consider a ℓ_1 regularized distributed logistic regression with linear equality constraint and local inequality constraints

$$\begin{aligned} & \text{minimize} \sum_{i=1}^n \sum_{j=1}^{n_i} \log(1 + \exp(-y_{ij}(a_{ij}^\top x))) + \theta \|\mathbf{x}\|_1 \\ & \text{subject to } H_i \mathbf{x} = h_i, \mathbf{x}^\top \mathbf{x} \leq r_i, \text{ for all } i \in \mathcal{V} \end{aligned} \quad (39)$$

where each agent $i \in \mathcal{V}$ has n_i data samples $\{(a_{ij}, y_{ij})\}_{j=1}^{n_i}$, with, $a_{ij} \in \mathbb{R}^{50}$ being the feature vector and $y_{ij} \in \{-1, +1\}$ is the binary outcome (or class label) of the feature vector a_{ij} . The objective is to learn the weight vector $x \in \mathbb{R}^{50}$ based on the available data $\{(a_{ij}, y_{ij})\}_{j=1}^{n_i}, i \in \mathcal{V}$ such that x is sparse. The parameter θ enforces the sparsity in x . Furthermore, the solution satisfy the linear equality constraints $H_i x = h_i$, where $H_i \in \mathbb{R}^{500 \times 50}$. Denote $\mathbf{H} := \mathbf{I} \otimes H_i$, $\mathbf{h} = [h_1^\top, \dots, h_n^\top]^\top$. Each agent $i \in \mathcal{V}$ also has a local set constraint $x^\top x \leq r_i$. For our simulation, we generate an instance of problem (39) where each agent has $n_i = 10^6$ data samples. Each feature vector $a_{ij} \in \mathbb{R}^{50}$ and the “true” weight vector $x_{\text{true}} \in \mathbb{R}^{50}$ are generated to have approximately 40% zero entries. The nonzero entries of a_{ij} and x_{true} are sampled independently from the standard normal distribution. The class labels y_{ij} are generated using the equation: $y_{ij} = \text{sign}(a_{ij}^\top x_{\text{true}} + \delta_{ij})$, where δ_{ij} is a normal random variable with zero mean and variance 0.1. The parameter θ is set to $0.1\theta_{\max}$, where $\theta_{\max}$ is the critical value above which the solution $x^* = 0$ (see [13] section 11.2 for the calculation of $\theta_{\max}$). The value of r_i in the constraints is chosen as follows: the unconstrained version of (39) with a centralized solver and denote the solution as x_c^* . The constraint r_i is set as $r_i = (1 + \xi_i)x_c^*$, where ξ_i is drawn from a uniform distribution on $[0, 1]$. A value of $\gamma = 10$ is used for the Augmented Lagrangian. The entries of each H_i are sampled independently from the standard normal distribution. The vector $h_i, i \in \mathcal{V}$ is calculated as $h_i = H_i x_{\text{true}}$. To solve (39) using the DC-DistADMM algorithm, the fast iterative shrinkage thresholding algorithm (FISTA) [69] is used for the updates (16) at each agent i . A proximal residue (difference between the output of the proximal minimization step and the base-point of the proximal term) [69] is utilized as a stopping criterion for FISTA. The FISTA iterations are terminated when the proximal residue becomes less than 10^{-4} . A fixed accuracy for FISTA is used at all iterations of DC-DistADMM algorithm. For the updates (17), the ε -consensus protocol with the tolerance ε set equal to a desired level of inaccuracy η in the solution of (17) is employed.

The results for the following three choices of the sequence $\{\eta_k\}_{k \geq 1}$:

- i) $\eta_k = 0.01$;
- ii) $\eta_k = 1/k^{2.1}$;
- iii) $\eta_k = 0.75^k$, where k is the iteration counter are presented for Algorithm 2.

Note, that sequences (ii) and (iii) satisfy condition (38) and are utilized to derive explicit rate of convergence for the DC-DistADMM algorithm in Theorems 4 and 5. Two existing algorithms [30] and [24] [termed as constrained distributed algorithm 1 (CDA1) and CDA2, respectively, for reference] for solving constrained distributed optimization problems are compared with the DC-DistADMM algorithm. Fig. 1 shows the plots of consensus constraint residual ($\|\hat{x}^k - \hat{y}^k\|$ for DC-DistADMM algorithm and $\|x^k - \sum_{i=1}^n x_i^k/n\|$ for CDA1 and CDA2) and the equality constraint residual for the three algorithms with respect to the total iterations. The DC-DistADMM performs well as shown in Fig. 1(a). The consensus residual under DC-DistADMM decreases to a value less than 10^{-4} within 200 iterations for the choice $\eta_k = 0.75^k$. Note, that the constant η_k sequence also have a good performance with a value less than 10^{-2} within 200 iterations. The plots in Fig. 1(b) demonstrate the decrease in the equality constraint residual $\|\mathbf{H}\hat{x}^k - \mathbf{h}\|$. Fig. 2 presents the trajectory of solution residuals $\frac{\|x^k - x^*\|^2}{\|x^0 - x^*\|^2}$ with respect to the total iterations and the CPU time (secs) for

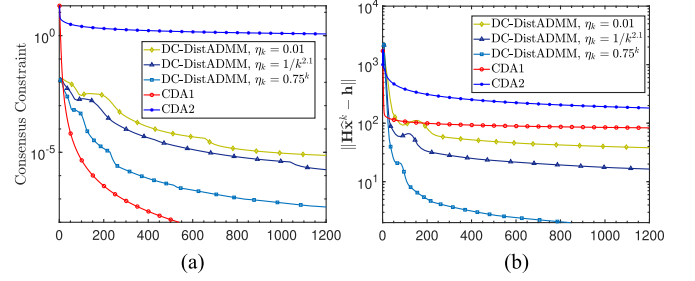


Fig. 1. (a) Comparison of consensus constraint residuals against total iterations (including communication steps) between DC-DistADMM and constrained optimization algorithms. (b) Comparison of equality constraint residuals against total iterations between DC-DistADMM and constrained optimization algorithms. $\eta^k = 0.75^k$ gives the best performance for the DC-DistADMM algorithm in terms of the equality constraint residual. The DC-DistADMM algorithm is second best in terms of the consensus constraints.

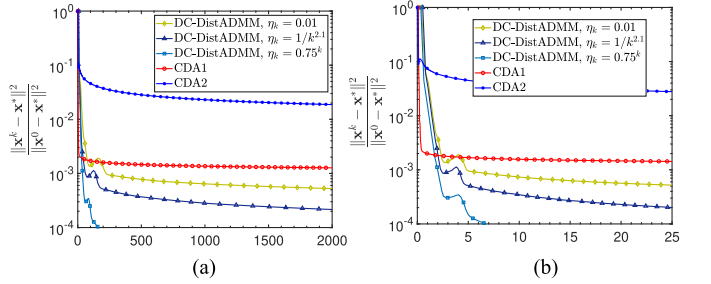


Fig. 2. (a) Comparison of solution residuals against total iterations between DC-DistADMM and constrained optimization algorithms. (b) Comparison of solution residuals against CPU time between DC-DistADMM and constrained optimization algorithms. DC-DistADMM algorithm with $\eta^k = 0.75^k$ has the best decrease in the solution residual.

each algorithm. The proposed DC-DistADMM algorithm outperforms the other two algorithms. The solution residual for DC-DistADMM is lesser than CDA1 and CDA2 for all the three choices of the tolerance sequence η_k with significant improvement with the choice $\eta_k = 0.75^k$. Note that both CDA1 and CDA2 cannot handle directed communication topologies and need centralized synthesis for problem (39) whereas DC-DistADMM does not.

B. Performance on Unconstrained Optimization Problem

In the second part of the numerical simulation study, a comparison of the proposed DC-DistADMM algorithm is presented with some state-of-the art distributed optimization algorithms for solving unconstrained optimization problems. Consider the following ℓ_1 regularized Huber loss minimization problem:

$$\text{minimize } \sum_{i=1}^n \Phi_i(x) + \theta \|x\|_1 \quad (40)$$

where $\Phi_i(x)$ for all i is the standard Huber loss given by

$$\Phi_i(x) = \begin{cases} \frac{1}{2} \|D_i x - d_i\|^2, & \text{if } \|D_i x - d_i\| \leq 1 \\ \|D_i x - d_i\| - \frac{1}{2}, & \text{if } \|D_i x - d_i\| > 1. \end{cases}$$

Each agent $i \in \mathcal{V}$ has a measured data vector $d_i \in \mathbb{R}^{100}$ and a scaling matrix $D_i \in \mathbb{R}^{100 \times 25}$. The objective is to estimate the unknown signal $x \in \mathbb{R}^{25}$. The entries of the matrices $D_i, i \in \mathcal{V}$ and the observed data d_i are sampled independently from the standard normal distribution. The true solution vector x^* has 70% nonzero entries, which are sampled from a standard normal

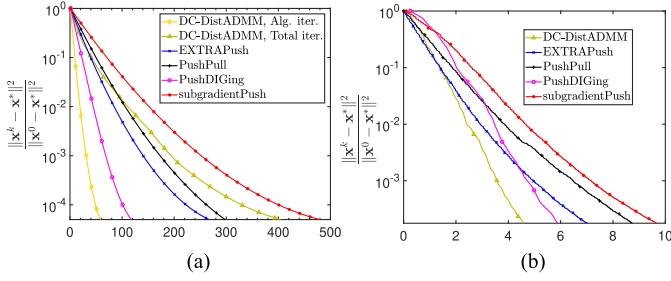


Fig. 3. (a) Comparison of solution residuals against total iterations between DC-DistADMM and unconstrained optimization algorithms. (b) Comparison of solution residuals against CPU time between DC-DistADMM and unconstrained optimization algorithms. DC-DistADMM algorithm performs the best in terms of the CPU time while has acceptable performance with respect to total iterations. DC-DistADMM has the best decrease in solution residual with respect to algorithm iterations.

distribution. A value of $\theta = 3$ is chosen for the regularization parameter. Here, FISTA is employed to solve the subproblem (16) with the same stopping criterion (proximal residue $< 10^{-4}$) as in the logistic regression problem. The update (17) is solved using ε -consensus protocol with the tolerance $\eta_k = 1/k^{2.1}$. The progression of the solution residual with respect to the total iterations and the CPU time is presented in Fig. 3. In Fig. 3(a), plot of the solution residual with respect to the algorithm iteration k (first curve) for the DC-DistADMM algorithm is also provided. The rationale for this is to provide a comparison between DC-DistADMM and other algorithm both with respect to algorithm iterations and total iterations. It can be seen that the proposed DC-DistADMM algorithm has a significantly better performance compared to other algorithms with respect to the algorithm iterations. In particular, solution residual decreases to a value of 10^{-4} in less than 50 algorithm iterations of DC-DistADMM algorithm whereas the second best method PushDIGing takes around 100 iterations to reach the same solution residual. In terms of total iterations the DC-DistADMM algorithm has acceptable performance. The ε -consensus protocol utilized at each iteration of the DC-DistADMM algorithm incurs additional consensus related steps, however, the information mixing steps (22) and (23) have a low computational footprint and do not significantly increase the overall run time of the DC-DistADMM algorithm. This is illustrated by the comparison between the algorithms based on the required CPU time shown in Fig. 3(b). The residual plots illustrate that DC-DistADMM algorithm performs better in terms of the CPU time requirement compared to the other methods to reach the same level of residual value.

C. Comparison With Algorithms Utilizing Multiple Communication Steps

Here, a comparison with two unconstrained distributed optimization algorithms that utilize multiple communication steps, nearDGD [57] and the algorithm in [62] [referred here as FastDGD (FDGD)] is provided; the reader is directed to [57], [62] for the motivations for multiple consensus steps. We consider an unconstrained distributed least squares problem

$$\text{minimize } \frac{1}{2} \sum_{i=1}^n \|D_i x - d_i\|^2. \quad (41)$$

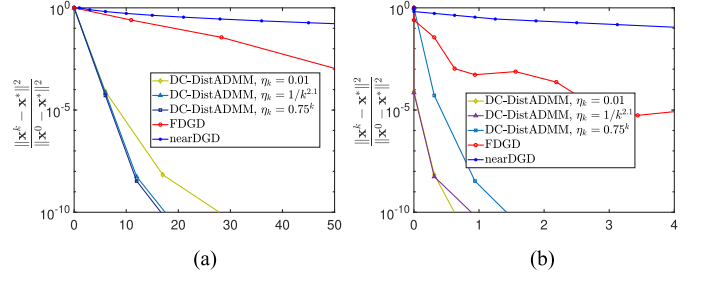


Fig. 4. (a) Comparison of solution residuals against total iterations between DC-DistADMM and algorithms utilizing multiple communication steps. (b) Comparison of solution residuals against CPU time between DC-DistADMM and algorithms utilizing multiple communication steps. $\eta_k = 0.75^k$ has the best decrease in solution residual with respect to total iterations. DC-DistADMM with a constant tolerance performs the best with respect to CPU time.

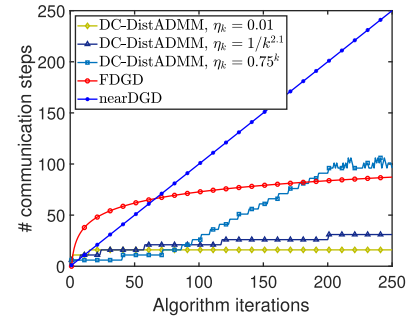


Fig. 5. Comparison of the number of communication iterations performed by DC-DistADMM algorithm for the three choices of η_k and the FDGD and nearDGD algorithms. DC-DistADMM with $\eta_k = 0.01$ and $\eta_k = 1/k^{2.1}$ has lesser communication cost compared to the FDGD and nearDGD algorithms while the FDGD has better communication complexity than DC-DistADMM with $\eta_k = 0.75^k$ as the number of iterations increase.

Here, each agent $i \in \mathcal{V}$ has a measured data vector $d_i \in \mathbb{R}^{100}$ and a scaling matrix $D_i \in \mathbb{R}^{100 \times 25}$. The objective is to estimate the unknown signal $x \in \mathbb{R}^{25}$. The entries of the matrices $D_i, i \in \mathcal{V}$ and the observed data d_i are sampled independently from a standard normal distribution $\mathcal{N}(0, 1)$. The true signal x^* also have entries that are sampled from an independent identically distributed standard normal distribution. In this case, the subproblem (16) takes a closed form solution, which can be readily computed by the first-order optimality condition of the unconstrained problem. The solution residuals obtained while solving (41) using the three algorithms are presented with respect to the total iterations and CPU time in Fig. 4. It can be seen that the DC-DistADMM algorithm has the fastest decrease in the solution residual. Fig. 5 presents the number of communication iterations utilized by the DC-DistADMM algorithm while solving (41) for the three error sequences and the FDGD and nearDGD algorithms. The trend suggested by Lemma 3 can be seen in the plots in Fig. 5, where the number of communication iterations for the DC-DistADMM algorithm increase as we tighten the accuracy. Note, the sequence $\eta_k = 1/k^{2.1}$ with convergence rate guarantees provided by Theorem 4 results in logarithmic increase in the number of communication iterations and is a suitable choice for getting good performance. The DC-DistADMM algorithm with tolerance $\eta_k = 0.01$ and $\eta_k = 1/k^{2.1}$ has lesser communication cost compared to the

FDGD and nearDGD algorithms while the FDGD has better communication complexity than the DC-DistADMM algorithm with the tolerance $\eta_k = 0.75^k$ as the number of iterations increase.

Summarizing, the simulation results indicate that with respect to the state-of-the-art *constrained* optimization problems DC-DistADMM provides a better performance while encompassing a large class of distributed optimization scenarios with linear equality, inequality, and set constraints. Even with respect to the state-of-the-art *unconstrained* optimization frameworks, DC-DistADMM algorithm provides a better performance with respect to computations (CPU time) and remains acceptable with respect to total communication steps.

VI. CONCLUSION

In this article, a novel DC-DistADMM algorithm is presented to solve constrained multiagent optimization problems with local linear equality, inequality and set constraints over general directed graphs. Moreover, the algorithm is suited for distributed synthesis. The proposed algorithm, to the best of authors' knowledge, is the first ADMM-based algorithm to solve (un)constrained distributed optimization problems over directed graphs. The DC-DistADMM algorithm combines techniques used in Lagrangian dual-based optimization methods along with the ideas in the average consensus literature. In the DC-DistADMM algorithm, each agent solves a local constrained convex optimization problem and utilizes a finite-time ε -consensus algorithm to update its estimate of the optimal solution. The proposed DC-DistADMM algorithm enjoys provable rate of convergence guarantees: (i) a $O(1/k)$ rate of convergence when the individual functions are convex but not-necessarily differentiable, (ii) a geometric decrease to arbitrary small neighborhood of the optimal solution when the objective functions are smooth and restricted strongly convex at the optimal solution. The proposed DC-DistADMM algorithm eliminates the optimization step involving a primal variable in the standard ADMM setup and replaces it with a less computation intensive ε -consensus protocol which makes it more suitable for distributed multiagent systems. To show the efficacy of the DC-DistADMM algorithm numerical simulation results comparing the performance of the DC-DistADMM algorithm with the existing state-of-the-art algorithms in the literature of solving constrained and unconstrained distributed optimization problems are presented. A comparison of the DC-DistADMM algorithm and two existing algorithms in the literature utilizing multiple consensus steps is also provided. Extension of the DC-DistADMM algorithmic framework to networks with time-delays in communication [70] between the agents and time-varying connectivity among the agents [71] is a future work of this article.

APPENDIX A: PROOF OF THEOREM 4

From (36) we have, for $i = 1, \dots, n$

$$df_i(x_i^{k+1}) + \lambda_i^{k+1} + A_i^\top \mu_i^{k+1} + \gamma(y_i^{k+1} - y_i^k) = 0$$

where $df_i(x_i^{k+1})$ is a subgradient of f_i at x_i^{k+1} . Writing the abovementioned n inequalities compactly

$$d\mathbf{F}(\mathbf{x}^{k+1}) + \lambda^{k+1} + \mathbf{A}^\top \mu^{k+1} + \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k) = 0$$

where $d\mathbf{F}(\mathbf{x}^{k+1})$ is a vector with subgradients $df_i(x_i^{k+1})$ stacked together, respectively. Furthermore, using (37) $d(\bar{\mathbf{y}}^{k+1}) + \bar{\lambda}^{k+1} = 0$, where $d(\bar{\mathbf{y}}^{k+1})$ is a subgradient of $\mathcal{I}_{C_0}$ at $\bar{\mathbf{y}}^{k+1}$. Noticing that $\mathbf{F}$ and $\mathcal{I}_{C_0}$ are convex functions and using (21) we have

$$\begin{aligned} & \mathbf{F}(\mathbf{x}^{k+1}) - \mathbf{F}(\mathbf{x}^*) + \mathcal{I}_{C_0}(\bar{\mathbf{y}}^{k+1}) - \mathcal{I}_{C_0}(\mathbf{y}^*) \\ & \leq -[\mathbf{x}^* - \mathbf{x}^{k+1}]^\top d\mathbf{F}(\mathbf{x}^{k+1}) - [\mathbf{y}^* - \bar{\mathbf{y}}^{k+1}]^\top d(\bar{\mathbf{y}}^{k+1}) \\ & \leq [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\lambda^{k+1} + \mathbf{A}^\top \mu^{k+1} + \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k)] \\ & \quad - [\mathbf{y}^* - \bar{\mathbf{y}}^{k+1}]^\top \bar{\lambda}^{k+1}. \end{aligned}$$

Note that $\mathcal{I}_{C_0}(\bar{\mathbf{y}}^{k+1}) = \mathcal{I}_{C_0}(\mathbf{y}^*) = 0$. Using (19) and (21)

$$\begin{aligned} & \mathbf{F}(\mathbf{x}^{k+1}) - \mathbf{F}(\mathbf{x}^*) \\ & \leq [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\lambda^{k+1} + \mathbf{A}^\top \mu^{k+1} + \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k)] \\ & \quad - [\mathbf{y}^* - \bar{\mathbf{y}}^{k+1}]^\top [\lambda^{k+1} + \gamma(\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1})] \\ & \leq [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\lambda^{k+1} + \mathbf{A}^\top \mu^{k+1} + \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k)] \\ & \quad - [\mathbf{y}^* - \bar{\mathbf{y}}^{k+1}]^\top \lambda^{k+1} + \gamma e^{k+1\top} (\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*) \\ & = [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\lambda^{k+1} + \mathbf{A}^\top \mu^{k+1} + \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k)] \\ & \quad - [\mathbf{y}^* - \bar{\mathbf{y}}^{k+1}]^\top \lambda^{k+1} + e^{k+1\top} (-\lambda^{k+1} + \gamma(\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*)) \\ & = [\mathbf{y}^{k+1} - \mathbf{x}^{k+1}]^\top \lambda^{k+1} + \gamma[\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{y}^{k+1} - \mathbf{y}^k] \\ & \quad + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top \mathbf{A}^\top \mu^{k+1} + e^{k+1\top} (-\lambda^{k+1} + \gamma(\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*)). \end{aligned}$$

Therefore, for any λ and μ , using (34), we have

$$\begin{aligned} & \mathbf{F}(\mathbf{x}^{k+1}) - \mathbf{F}(\mathbf{x}^*) + \lambda^\top (\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) + \mu^\top (\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \\ & \leq [\lambda^{k+1} - \lambda]^\top [\mathbf{y}^{k+1} - \mathbf{x}^{k+1}] + \gamma[\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{y}^{k+1} - \mathbf{y}^k] \\ & \quad + [\mu - \mu^{k+1}]^\top \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{b}) \\ & \quad + e^{k+1\top} (-\lambda^{k+1} + \gamma(\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*)). \end{aligned}$$

Using (19) and (20), we get

$$\begin{aligned} & \mathbf{F}(\mathbf{x}^{k+1}) - \mathbf{F}(\mathbf{x}^*) + \lambda^\top (\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) + \mu^\top (\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \\ & \leq \frac{1}{\gamma} [\lambda - \lambda^{k+1}]^\top [\lambda^{k+1} - \lambda^k] + \gamma[\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{y}^{k+1} - \mathbf{y}^k] \\ & \quad + \frac{1}{\gamma} [\mu - \mu^{k+1}]^\top [\mu^{k+1} - \mu^k] \\ & \quad + e^{k+1\top} (-\lambda^{k+1} + \gamma(\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*)). \end{aligned} \quad (42)$$

Using the identity (30), for any $s \geq 0$ we get

$$\begin{aligned} & \frac{1}{\gamma} [\lambda - \lambda^{s+1}]^\top [\lambda^{s+1} - \lambda^s] = \frac{1}{2\gamma} (\|\lambda - \lambda^s\|^2 \\ & \quad - \|\lambda - \lambda^{s+1}\|^2 - \|\lambda^{s+1} - \lambda^s\|^2) \end{aligned} \quad (43)$$

$$\begin{aligned} & \gamma[\mathbf{x}^* - \mathbf{x}^{s+1}]^\top [\mathbf{y}^{s+1} - \mathbf{y}^s] = \frac{\gamma}{2} (\|\mathbf{x}^{s+1} - \mathbf{y}^{s+1}\|^2 \\ & \quad - \|\mathbf{x}^{s+1} - \mathbf{y}^s\|^2 + \|\mathbf{x}^* - \mathbf{y}^s\|^2 - \|\mathbf{x}^* - \mathbf{y}^{s+1}\|^2) \end{aligned} \quad (44)$$

$$\begin{aligned} & \frac{1}{\gamma} [\mu - \mu^{k+1}]^\top [\mu^{k+1} - \mu^k] = \frac{1}{2\gamma} \\ & \quad \times (\|\mu - \mu^k\|^2 - \|\mu - \mu^{k+1}\|^2 - \|\mu^{k+1} - \mu^k\|^2). \end{aligned} \quad (45)$$

Note that from Lemma 2, (19) and (29), under Assumption 2, $e^{k+1\top}(-\lambda^{k+1} + \gamma(\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*)) = -e^{k+1\top}\lambda^{k+1} + \gamma e^{k+1\top}\mathbf{y}^{k+1} - \gamma\|e^{k+1}\|^2 - e^{k+1\top}\mathbf{y}^* \leq \|e^{k+1}\|\|\lambda^{k+1}\| + \gamma\|e^{k+1}\|\|\mathbf{y}^{k+1}\| + \|e^{k+1}\|\|\mathbf{y}^*\| \leq (k+1)\mathbf{Q}\sqrt{n}\eta_{k+1} + (\gamma n(\mathcal{M} + \|\mathbf{x}^*\|) + 2(k+1)\sqrt{n}\mathbf{Q})\eta_{k+1} + \sqrt{n}\|\mathbf{y}^*\|\eta_{k+1}$. Let $\mathbf{M} := \gamma n(\mathcal{M} + \|\mathbf{x}^*\|) + \sqrt{n}\|\mathbf{y}^*\|$. From (54), (55), (56), and (57) for any λ, μ , and $s \geq 0$ we get

$$\begin{aligned} & \mathbf{F}(\mathbf{x}^{s+1}) - \mathbf{F}(\mathbf{x}^*) + \lambda^\top(\mathbf{x}^{s+1} - \mathbf{y}^{s+1}) + \mu^\top(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \\ & \leq \frac{1}{2\gamma}(\|\lambda - \lambda^s\|^2 - \|\lambda - \lambda^{s+1}\|^2) + \frac{\gamma}{2}(\|\mathbf{x}^* - \mathbf{y}^s\|^2 \\ & \quad - \|\mathbf{x}^* - \mathbf{y}^{s+1}\|^2) + \frac{1}{2\gamma}(\|\mu - \mu^s\|^2 - \|\mu - \mu^{s+1}\|^2) \\ & \quad + \frac{\mathbf{M}}{s^{2+q}} + 3\sqrt{n}\mathbf{Q}(s+1)\frac{1}{(s+1)^{2+q}}. \end{aligned}$$

Summing from $s = 0$ to $k-1$ and dividing by $\frac{1}{k}$ gives

$$\begin{aligned} & \frac{1}{k} \sum_{s=0}^{k-1} [\mathbf{F}(\mathbf{x}^{s+1}) - \mathbf{F}(\mathbf{x}^*) + \lambda^\top(\mathbf{x}^{s+1} - \mathbf{y}^{s+1}) \\ & \quad \mu^\top(\mathbf{A}\mathbf{x}^{s+1} - \mathbf{b})] \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} + \frac{\|\lambda - \lambda^0\|^2}{2\gamma k} + \frac{\|\mu - \mu^0\|^2}{2\gamma k} \\ & \quad + \frac{\mathbf{M} \sum_{s=0}^{k-1} \frac{1}{s^{2+q}}}{k} + \frac{3\sqrt{n}\mathbf{Q} \sum_{s=0}^{k-1} (s+1) \frac{1}{(s+1)^{2+q}}}{k}. \end{aligned}$$

Let $\hat{\mathbf{x}}^k := \frac{1}{k} \sum_{s=0}^{k-1} \mathbf{x}^{s+1}$, and $\hat{\mathbf{y}}^k := \frac{1}{k} \sum_{s=0}^{k-1} \mathbf{y}^{s+1}$. Since, $\mathbf{F}$ is a convex function we have, $\frac{1}{k} \sum_{s=0}^{k-1} \mathbf{F}(\mathbf{x}^{s+1}) \geq \mathbf{F}(\hat{\mathbf{x}}^k)$. Therefore

$$\begin{aligned} & \mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) + \lambda^\top(\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k) + \mu^\top(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}) \\ & \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} + \frac{\|\lambda - \lambda^0\|^2}{2\gamma k} + \frac{\|\mu - \mu^0\|^2}{2\gamma k} \\ & \quad + \frac{\mathbf{M}\zeta(2+q) + 3\sqrt{n}\mathbf{Q}\zeta(1+q)}{k} \end{aligned} \quad (46)$$

where $\zeta(\cdot)$ is the Riemann zeta function. Let $\mu = 0, \lambda = 0$, therefore, we get

$$\begin{aligned} & \mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} + \frac{\|\lambda^0\|^2 + \|\mu^0\|^2}{2\gamma k} \\ & \quad + \frac{\mathbf{M}\zeta(2+q) + 3\sqrt{n}\mathbf{Q}\zeta(1+q)}{k} = O(1/k). \end{aligned}$$

Define, $\hat{\bar{\mathbf{y}}}^k := \frac{1}{k} \sum_{s=0}^{k-1} \bar{\mathbf{y}}^{s+1}$, where $\bar{\mathbf{y}}^{s+1}$ is as defined in (18). Since, $(\mathbf{x}^*, \mathbf{y}^*, \lambda^*, \mu^*)$ is a saddle point we have, $\mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \lambda, \mu) \leq \mathcal{L}(\hat{\mathbf{x}}^k, \hat{\bar{\mathbf{y}}}^k, \lambda^*, \mu^*)$. Therefore, $\mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) + \lambda^{*\top}(\hat{\mathbf{x}}^k - \hat{\bar{\mathbf{y}}}^k) + \lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k) + \mu^{*\top}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}) \geq 0$. Let $\lambda = \lambda^* + \frac{\hat{\mathbf{x}}^k - \hat{\bar{\mathbf{y}}}^k}{\|\hat{\mathbf{x}}^k - \hat{\bar{\mathbf{y}}}^k\|}$ and $\mu = \mu^*$. Therefore

$$\begin{aligned} & \|\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k\| \leq \mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) + \lambda^{*\top}(\hat{\mathbf{x}}^k - \hat{\bar{\mathbf{y}}}^k) \\ & \quad + \mu^{*\top}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}) + \lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k) + \|\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k\| \\ & \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} \\ & \quad + \frac{1 + \|\lambda^0 - \lambda^*\|^2}{\gamma k} + \frac{\|\mu^* - \mu^0\|^2}{2\gamma k} \end{aligned}$$

$$\begin{aligned} & + \frac{\mathbf{M}\zeta(2+q) + 3\sqrt{n}\mathbf{Q}\zeta(1+q)}{k} \\ & + \lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k). \end{aligned}$$

Note that $\|\lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k)\| \leq \|\lambda^*\| \|\frac{1}{k} \sum_{s=0}^{k-1} [\mathbf{y}^{s+1} - \bar{\mathbf{y}}^{s+1}]\| = \frac{\|\lambda^*\|}{k} \sum_{s=0}^{k-1} \|\mathbf{y}^{s+1} - \bar{\mathbf{y}}^{s+1}\| = \frac{\|\lambda^*\|}{k} \sum_{s=0}^{k-1} \|e^{s+1}\| \leq \frac{\|\lambda^*\|}{k} \sum_{s=0}^{k-1} \sqrt{n}\eta_{s+1} \leq \frac{\sqrt{n}\|\lambda^*\|}{k} \sum_{s=0}^{\infty} \eta_{s+1} = \frac{\sqrt{n}\|\lambda^*\|}{k} \zeta(2+q)$. Therefore

$$\begin{aligned} & \|\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k\| \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} + \frac{1 + \|\lambda^0 - \lambda^*\|^2}{\gamma k} + \frac{\|\mu^* - \mu^0\|^2}{2\gamma k} \\ & \quad + \frac{(\mathbf{M} + \sqrt{n}\|\lambda^*\|)\zeta(2+q) + 3\sqrt{n}\mathbf{Q}\zeta(1+q)}{k} \\ & = O(1/k). \end{aligned}$$

Alternatively, let, $\mu = \mu^* + \frac{\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}}{\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|}$ and $\lambda = \lambda^*$, we get

$$\begin{aligned} & \|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\| \leq \mathbf{F}(\hat{\mathbf{x}}^k) - \mathbf{F}(\mathbf{x}^*) + \lambda^{*\top}(\hat{\mathbf{x}}^k - \hat{\mathbf{y}}^k) \\ & \quad + \mu^{*\top}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}) + \lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k) + \|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\| \\ & \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} \\ & \quad + \frac{\|\lambda^0 - \lambda^*\|^2}{2\gamma k} + \frac{1 + \|\mu - \mu^0\|^2}{\gamma k} + \frac{\mathbf{M}\zeta(1+q)}{k} \\ & \quad + \lambda^{*\top}(\hat{\bar{\mathbf{y}}}^k - \hat{\mathbf{y}}^k) \\ & \leq \frac{\gamma\|\mathbf{x}^* - \mathbf{y}^0\|^2}{2k} + \frac{\|\lambda^0 - \lambda^*\|^2}{2\gamma k} + \frac{1 + \|\mu - \mu^0\|^2}{\gamma k} \\ & \quad + \frac{(\mathbf{M} + \sqrt{n}\|\lambda^*\|)\zeta(2+q) + 3\sqrt{n}\mathbf{Q}\zeta(1+q)}{k} = O(1/k). \end{aligned} \quad \square$$

APPENDIX B: PROOF OF THEOREM 5

From the definition of $\mathbf{F}$ and Assumption 5, it follows that $\mathbf{F}$ is Lipschitz differentiable with constant $L := \max_{1 \leq i \leq n} L_{f_i}$ for all $\bar{\mathbf{x}} \in \text{relint}(\mathcal{X})$, $\hat{\mathbf{x}}_s \in \text{relint}({}^n\mathbb{R}_{\geq 0}^m)$ and restricted strongly convex with respect to the optimal solution $\mathbf{x}^*$ on the set $\mathcal{X}$ with parameter $\sigma := \min_{1 \leq i \leq n} \sigma_i$. For subgradients $d\mathbf{F}(\mathbf{x}^{k+1}) \in \partial\mathbf{F}(\mathbf{x}^{k+1})$ and $d\mathbf{F}(\mathbf{x}^*) \in \partial\mathbf{F}(\mathbf{x}^*)$, using (31) and (36) under Assumption 5

$$\begin{aligned} & \sigma\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \leq [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [d\mathbf{F}(\mathbf{x}^{k+1}) - d\mathbf{F}(\mathbf{x}^*)] \\ & = [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [-\lambda^{k+1} - \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k) - \mathbf{A}^\top \mu^{k+1} \\ & \quad + \lambda^* + \mathbf{A}^\top \mu^*] \\ & = [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [\lambda^* - \lambda^{k+1}] - \gamma[\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [\mathbf{y}^{k+1} - \mathbf{y}^k] \\ & \quad + [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top \mathbf{A}^\top [\mu^* - \mu^{k+1}] \\ & = [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [\lambda^* - \lambda^{k+1}] + [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top \mathbf{A}^\top [\mu^* - \mu^{k+1}] \\ & \quad + \gamma[\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{x}^k] \\ & \quad + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [2\lambda^k - \lambda^{k+1} - \lambda^{k-1}]. \end{aligned}$$

Using, (43) and (44) we get

$$\begin{aligned}
& \leq [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top [\lambda^* - \lambda^{k+1}] + \gamma [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{x}^k] \\
& + [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top \mathbf{A}^\top [\mu^* - \mu^{k+1}] + [\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*]^\top [\lambda^{k+1} - \lambda^*] \\
& - [\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*]^\top \lambda^{k+1} + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [2\lambda^k - \lambda^{k+1} - \lambda^{k-1}] \\
& = [\lambda^* - \lambda^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{y}^{k+1}] + \gamma [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{x}^k] \\
& + [\mathbf{x}^{k+1} - \mathbf{x}^*]^\top \mathbf{A}^\top [\mu^* - \mu^{k+1}] - [\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*]^\top \lambda^{k+1} \\
& + [\lambda^* - \lambda^{k+1}]^\top e^{k+1} + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [2\lambda^k - \lambda^{k+1} - \lambda^{k-1}] \\
& = \frac{1}{\gamma} [\lambda^* - \lambda^{k+1}]^\top [\lambda^{k+1} - \lambda^k] + \gamma [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{x}^k] \\
& + \frac{1}{\gamma} [\mu^* - \mu^{k+1}]^\top [\mu^{k+1} - \mu^k] - [\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*]^\top \lambda^{k+1} \\
& + [\lambda^* - \lambda^{k+1}]^\top e^{k+1} + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [2\lambda^k - \lambda^{k+1} - \lambda^{k-1}]
\end{aligned}$$

where in the second equality we used (29), and in the last equality we used (19) and (20). Let $t_k := 2\lambda^k - \lambda^{k+1} - \lambda^{k-1}$. Using (30) for $\frac{1}{\gamma} [\lambda^* - \lambda^{k+1}]^\top [\lambda^{k+1} - \lambda^k]$, $\frac{1}{\gamma} [\mu^* - \mu^{k+1}]^\top [\mu^{k+1} - \mu^k]$, and $\gamma [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top [\mathbf{x}^{k+1} - \mathbf{x}^k]$

$$\begin{aligned}
& \sigma \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + \frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{1}{2\gamma} \|\lambda^{k+1} - \lambda^k\|^2 \\
& \leq \left[\frac{\gamma}{2} \|\mathbf{x}^k - \mathbf{x}^*\|^2 + \frac{1}{2\gamma} \|\lambda^k - \lambda^*\|^2 + \frac{1}{2\gamma} \|\mu^k - \mu^*\|^2 \right] \\
& - \left[\frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + \frac{1}{2\gamma} \|\lambda^{k+1} - \lambda^*\|^2 + \frac{1}{2\gamma} \|\mu^{k+1} - \mu^*\|^2 \right] \\
& - [\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*]^\top \lambda^{k+1} + [\lambda^* - \lambda^{k+1}]^\top e^{k+1} + [\mathbf{x}^* - \mathbf{x}^{k+1}]^\top t_k. \tag{47}
\end{aligned}$$

Observe from (31), (36), $\nabla \mathbf{F}(\mathbf{x}^{k+1}) - \nabla \mathbf{F}(\mathbf{x}^*) = -\lambda^{k+1} - \gamma(\mathbf{y}^{k+1} - \mathbf{y}^k) - \mathbf{A}^\top \mu^{k+1} + \lambda^* + \mathbf{A}^\top \mu^* = -\lambda^{k+1} + \lambda^* - \mathbf{A}^\top \mu^{k+1} + \mathbf{A}^\top \mu^* + \lambda^{k+1} + \lambda^{k-1} - 2\lambda^k - \gamma(\mathbf{x}^{k+1} - \mathbf{x}^k)$, where we used (19) to get a relation between $\mathbf{y}^{k+1}$ and $\mathbf{y}^k$. Therefore, $\gamma(\mathbf{x}^{k+1} - \mathbf{x}^k)$ is the summation of the terms $\mathbf{q}_1 := \nabla \mathbf{F}(\mathbf{x}^*) - \nabla \mathbf{F}(\mathbf{x}^{k+1})$ and $\mathbf{r}_1 := \lambda^* - \lambda^{k+1} + \mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} + \lambda^{k+1} + \lambda^{k-1} - 2\lambda^k$. Therefore, for any $\delta > 1$, we can apply the inequality $\|\mathbf{q}_1 + \mathbf{r}_1\|^2 + (\delta - 1)\|\mathbf{q}_1\|^2 \geq (1 - \frac{1}{\delta})\|\mathbf{r}_1\|^2$

$$\begin{aligned}
& \gamma^2 \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + (\delta - 1) \|\nabla \mathbf{F}(\mathbf{x}^{k+1}) - \nabla \mathbf{F}(\mathbf{x}^*)\|^2 \\
& \geq \left(1 - \frac{1}{\delta}\right) \|\lambda^* - \lambda^{k+1} + \mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\|^2 \\
& \geq \left(1 - \frac{1}{\delta}\right) [\|\lambda^{k+1} - \lambda^*\|^2 + \nu_{\min}(\mathbf{A}\mathbf{A}^\top) \|\mu^{k+1} - \mu^*\|^2 \\
& + \|t_k\|^2 - 2\|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\| \\
& - 2\|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\|]
\end{aligned}$$

where $\nu_{\min}(\mathbf{A}\mathbf{A}^\top) > 0$. By Lipschitz differentiability of $\mathbf{F}$

$$\frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{(\delta - 1)L^2}{2\gamma} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2$$

$$\begin{aligned}
& \geq \frac{(1 - \frac{1}{\delta})}{2\gamma} [\|\lambda^{k+1} - \lambda^*\|^2 + \nu_{\min}(\mathbf{A}\mathbf{A}^\top) \|\mu^{k+1} - \mu^*\|^2 \\
& - 2\|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\| \\
& - 2\|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\|]. \tag{48}
\end{aligned}$$

Similarly, $\lambda^{k+1} - \lambda^k = \gamma(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) = \gamma(\mathbf{x}^{k+1} - \mathbf{x}^*) + \gamma(\mathbf{y}^* - \mathbf{y}^{k+1})$. Therefore, $\lambda^{k+1} - \lambda^k$ is the summation of $\mathbf{q}_2 := \gamma(\mathbf{x}^{k+1} - \mathbf{x}^*)$ and $\mathbf{r}_2 := \gamma(\mathbf{y}^* - \mathbf{y}^{k+1})$. Therefore, for any $\delta > 1$, we can apply the inequality $\|\mathbf{q}_2 + \mathbf{r}_2\|^2 + (\delta - 1)\|\mathbf{q}_2\|^2 \geq (1 - \frac{1}{\delta})\|\mathbf{r}_2\|^2$, and obtain

$$\begin{aligned}
& \|\lambda^{k+1} - \lambda^k\|^2 + \gamma^2(\delta - 1) \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
& \geq \left(1 - \frac{1}{\delta}\right) \|\gamma(\mathbf{y}^{k+1} - \mathbf{y}^*)\|^2 \\
& = \left(1 - \frac{1}{\delta}\right) \|\lambda^{k+1} - \lambda^k - \gamma(\mathbf{x}^{k+1} - \mathbf{x}^*)\|^2 \\
& \geq \left(1 - \frac{1}{\delta}\right) [\|\lambda^{k+1} - \lambda^k\|^2 + \gamma^2 \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
& - 2\gamma \|\lambda^{k+1} - \lambda^k\| \|\mathbf{x}^{k+1} - \mathbf{x}^*\|]. \tag{49}
\end{aligned}$$

Therefore, using (60) and (61) we get

$$\begin{aligned}
& \frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{1}{2\gamma} \|\lambda^{k+1} - \lambda^k\|^2 \\
& + \frac{(\delta - 1)(\frac{L^2}{\gamma} + \gamma)}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
& \geq \left(1 - \frac{1}{\delta}\right) \left[\frac{1}{2\gamma} \|\lambda^{k+1} - \lambda^*\|^2 + \frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \right. \\
& + \frac{\nu_{\min}(\mathbf{A}\mathbf{A}^\top)}{2\gamma} \|\mu^{k+1} - \mu^*\|^2 \left. \right] \\
& - \frac{(1 - 1/\delta)}{\gamma} [\|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\| \\
& + \|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\| + \gamma \|\lambda^{k+1} - \lambda^k\| \|\mathbf{x}^{k+1} - \mathbf{x}^*\|]. \\
& \text{Let } \Delta := (1 - \frac{1}{\delta}) \min\{1, \nu_{\min}(\mathbf{A}\mathbf{A}^\top)\} \text{ and } s_k := \frac{\gamma}{2} \|\mathbf{x}^k - \mathbf{x}^*\|^2 + \frac{1}{2\gamma} \|\lambda^k - \lambda^*\|^2 + \frac{1}{2\gamma} \|\mu^{k+1} - \mu^*\|^2. \text{ Therefore, for } \delta \in (1, 1 + \frac{2\sigma\gamma}{L^2 + \gamma^2}], \text{ we have} \\
& \sigma \|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + \frac{\gamma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{1}{2\gamma} \|\lambda^{k+1} - \lambda^k\|^2 \\
& \geq \Delta s_{k+1} - \frac{(1 - 1/\delta)}{\gamma} [\|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\| \\
& + \|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\| + \gamma \|\lambda^{k+1} - \lambda^k\| \|\mathbf{x}^{k+1} - \mathbf{x}^*\|]. \tag{50}
\end{aligned}$$

Therefore, from (59) and (62), we have

$$\begin{aligned}
s_{k+1} & \leq \left(\frac{1}{1 + \Delta}\right) s_k + \left(\frac{1}{1 + \Delta}\right) (\|\lambda^{k+1} - \lambda^*\| e^{k+1} \\
& + \|\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*\| \|\lambda^{k+1}\| + \|\mathbf{x}^* - \mathbf{x}^{k+1}\| \|t_k\| \\
& + \frac{(1 - 1/\delta)}{\gamma} [\|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\|
\end{aligned}$$

$$+ \|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\| + \gamma \|\lambda^{k+1} - \lambda^k\| \|\mathbf{x}^{k+1} - \mathbf{x}^*\| \Big]. \quad (51)$$

Using (47)

$$\|\lambda^k - \lambda^*\| \leq \sqrt{2\gamma \left(s_0 + \frac{\mathbf{R}}{1-\rho} + \frac{3\sqrt{n}\mathbf{Q}\rho}{(1-\rho)^2} \right)} := \Xi. \quad (52)$$

Let $S_1 := \|\lambda^{k+1} - \lambda^*\| \|e^{k+1}\|$, $S_2 := \|\bar{\mathbf{y}}^{k+1} - \mathbf{y}^*\| \|\lambda^{k+1}\|$, $S_3 := \|\mathbf{x}^* - \mathbf{x}^{k+1}\| \|t_k\|$, $S_4 := \frac{(1-1/\delta)}{\gamma} \|\lambda^{k+1} - \lambda^*\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1} - t_k\|$, $S_5 := \frac{(1-1/\delta)}{\gamma} \|t_k\| \|\mathbf{A}^\top \mu^* - \mathbf{A}^\top \mu^{k+1}\|$ and $S_6 := (1 - 1/\delta) \|\lambda^{k+1} - \lambda^*\| \|\mathbf{x}^{k+1} - \mathbf{x}^*\|$. Note $S_1 = \|\lambda^{k+1} - \lambda^*\| \|e^{k+1}\| \leq \sqrt{n} \Xi \rho^{k+1}$. Furthermore, from Theorem 3, there exist $\mathbf{C} < \infty$ and $\mathbf{S}^k$ with $\lim_{k \rightarrow \infty} \mathbf{S}^k = 0$ such that $S_2 + S_3 + S_4 + S_5 + S_6 \leq \mathbf{C} \mathbf{S}^k$. Therefore, from (63) and using the fact $\rho^{k+1} \leq \rho^k, \forall k$

$$s_{k+1} \leq \left(\frac{1}{1+\Delta}\right) s_k + \sqrt{n}\Xi \left(\frac{1}{1+\Delta}\right) \rho^k + \mathbf{C} \left(\frac{1}{1+\Delta}\right) \mathbf{S}^k. \quad (53)$$

By repeated substitution for $k = 0, 1, \dots, K$ we get

$$\begin{aligned}
s_K &\leq \left(\frac{1}{1+\Delta}\right)^K s_0 + \sqrt{n}\Xi \sum_{t=0}^K \left(\frac{1}{1+\Delta}\right)^t \rho^{K-t} \\
&\quad + \mathbf{C} \sum_{t=0}^K \left(\frac{1}{1+\Delta}\right)^{K-t+1} \mathbf{S}^t \\
&= \left(\frac{1}{1+\Delta}\right)^K s_0 + \rho^K \sqrt{n}\Xi \sum_{t=1}^K \left(\frac{1}{1+\Delta}\right)^t \rho^{-t} \\
&\quad + \mathbf{C} \sum_{t=0}^K \left(\frac{1}{1+\Delta}\right)^{K-t+1} \mathbf{S}^t \\
&\leq \rho^K \left(s_0 + \sqrt{n}\Xi \sum_{t=1}^K \left(\frac{1}{1+\Delta}\right)^t \rho^{-t} \right) \\
&\quad + \mathbf{C} \sum_{t=0}^K \left(\frac{1}{1+\Delta}\right)^{K-t+1} \mathbf{S}^t.
\end{aligned}$$

Given, $\epsilon > 0$, let K' be such that $\mathbf{S}^k < \epsilon$ for all $k \geq K'$ (Theorem 3 guarantees existence of such K'). Let $\bar{\mathbf{S}} := \max_{0 \leq t \leq K'} \mathbf{S}^t$, therefore

$$\begin{aligned} s_K &\leq \rho^K \left(s_0 + \sqrt{n} \Xi \sum_{t=1}^K \left(\frac{1}{1+\Delta} \right)^t \rho^{-t} + \mathbf{C} \bar{\mathbf{S}} \sum_{t=0}^{K'} \rho^{1-t} \right) \\ &\quad + \epsilon \left(\mathbf{C} \sum_{t=K'}^K \rho^{K-t+1} \right) \\ &\leq \rho^K \left(s_0 + \frac{\sqrt{n} \Xi \rho (1+\Delta)}{\rho(1+\Delta) - 1} + \frac{\mathbf{C} \bar{\mathbf{S}} (\rho^{-K'+1} - \rho^2)}{1 - \rho} \right) \\ &\quad + \epsilon \left(\frac{\mathbf{C}}{1 - \rho} \right) \\ &= \Upsilon \rho^K + O(\epsilon). \end{aligned}$$

$$\text{where } \Upsilon := s_0 + \frac{\mathbf{S}_2 \rho(1 + \Delta)}{\rho(1 + \Delta) - 1}. \quad (54)$$

☐

REFERENCES

- [1] A. Nedić and J. Liu, “Distributed optimization for control,” *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 77–103, 2018.
- [2] A. Nedić, “Distributed gradient methods for convex machine learning problems in networks: Distributed optimization,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 92–101, May 2020.
- [3] X. Wu, H. Wang, and J. Lu, “Distributed optimization with coupling constraints,” *IEEE Trans. Autom. Control*, 2021, doi: [10.1109/TAC.2022.3169955](https://doi.org/10.1109/TAC.2022.3169955).
- [4] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, “Dual decomposition for multi-agent distributed optimization with coupling constraints,” *Automatica*, vol. 84, pp. 149–158, 2017.
- [5] I. Notarnicola and G. Notarstefano, “Constraint-coupled distributed optimization: A relaxation and duality approach,” *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 483–492, Mar. 2020.
- [6] J. N. Tsitsiklis, “*Problems in decentralized decision making and computation*,” Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Cambridge, MA, USA, Tech. Rep., ADA150025, 1984.
- [7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.
- [8] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.
- [9] R. Xin, S. Kar, and U. A. Khan, “Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 102–113, May 2020.
- [10] J. Zeng and W. Yin, “Extrapush for convex smooth decentralized optimization over directed networks,” *J. Comput. Math.*, vol. 35, no. 4, pp. 383–396, 2017.
- [11] S. Pu, W. Shi, J. Xu, and A. Nedić, “Push-pull gradient methods for distributed optimization in networks,” *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Jan. 2021.
- [12] V. Khatana, G. Saraswat, S. Patel, and M. V. Salapaka, “Gradient-consensus method for distributed optimization in directed multi-agent networks,” 2021. [Online]. Available: <https://arxiv.org/pdf/1909.10070v7>
- [13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: <http://dx.doi.org/10.1561/22000000016>
- [14] E. Wei and A. Ozdaglar, “Distributed alternating direction method of multipliers,” in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 5445–5450.
- [15] A. Makhdoumi and A. Ozdaglar, “Convergence rate of distributed ADMM over networks,” *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5082–5095, Oct. 2017.
- [16] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the ADMM in decentralized consensus optimization,” *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
- [17] E. Wei and A. Ozdaglar, “On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers,” in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 551–554.
- [18] T.-H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus ADMM,” *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
- [19] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, “Explicit convergence rate of a distributed alternating direction method of multipliers,” *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 892–904, Apr. 2016.
- [20] F. Mansoori and E. Wei, “A flexible framework of first-order primal-dual algorithms for distributed optimization,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3500–3512, 2021, doi: [10.1109/TSP.2021.3083981](https://doi.org/10.1109/TSP.2021.3083981).
- [21] D. Gabay, “Chapter IX applications of the method of multipliers to variational inequalities,” in *Studies in Mathematics and Its Applications*, vol. 15. Amsterdam, The Netherlands: Elsevier, 1983, pp. 299–331.

- [22] G. Chen, Q. Yang, Y. Song, and F. L. Lewis, "A distributed continuous-time algorithm for nonsmooth constrained optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4914–4921, Nov. 2020.
- [23] G. Chen, Q. Yang, Y. Song, and F. L. Lewis, "Fixed-time projection algorithm for distributed constrained optimization on time-varying digraphs," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 390–397, Jan. 2022.
- [24] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.
- [25] F. Tian, W. Yu, J. Fu, W. Gu, and J. Gu, "Distributed optimization of multiagent systems subject to inequality constraints," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 2232–2241, Apr. 2021.
- [26] S. Yang, Q. Liu, and J. Wang, "A multi-agent system with a proportional-integral protocol for distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3461–3467, Jul. 2017.
- [27] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3310–3315, Dec. 2015.
- [28] H. Zhou, X. Zeng, and Y. Hong, "Adaptive exact penalty design for constrained distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4661–4667, Nov. 2019.
- [29] P. Lin, J. Xu, W. Ren, C. Yang, and W. Gui, "Angle-based analysis approach for distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5569–5576, Nov. 2021.
- [30] Q. Liu, S. Yang, and Y. Hong, "Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4259–4265, Aug. 2017.
- [31] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
- [32] H. Li, Q. Lü, G. Chen, T. Huang, and Z. Dong, "Distributed constrained optimization over unbalanced directed networks using asynchronous broadcast-based algorithm," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1102–1115, Mar. 2021.
- [33] P. Xie, K. You, R. Tempo, S. Song, and C. Wu, "Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4331–4337, Dec. 2018.
- [34] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.
- [35] D. Yuan, D. W. Ho, and S. Xu, "Regularized primal-dual subgradient method for distributed constrained optimization," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2109–2118, Sep. 2016.
- [36] J. Lei, H.-F. Chen, and H.-T. Fang, "Primal-dual algorithm for distributed constrained optimization," *Syst. Control Lett.*, vol. 96, pp. 110–117, 2016.
- [37] S. Patel, S. Attree, S. Talukdar, M. Prakash, and M. V. Salapaka, "Distributed apportioning in a power network for providing demand response services," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, 2017, pp. 38–44.
- [38] S. Patel, V. Khatana, G. Saraswat, and M. V. Salapaka, "Distributed detection of malicious attacks on consensus algorithms with applications in power networks," in *Proc. IEEE 7th Int. Conf. Control, Decis. Inf. Technol.*, 2020, vol. 1, pp. 397–402.
- [39] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
- [40] W. Jiang and T. Charalambous, "Fully distributed alternating direction method of multipliers in digraphs via finite-time termination mechanisms," 2021, *arXiv:2107.02019*.
- [41] W. Jiang and T. Charalambous, "Distributed alternating direction method of multipliers using finite-time exact ratio consensus in digraphs," in *Proc. Eur. Control Conf.*, 2021, pp. 2205–2212, doi: [10.23919/ECC54610.2021.9654976](https://doi.org/10.23919/ECC54610.2021.9654976).
- [42] W. Jiang, A. Grammenos, E. Kalyvianaki, and T. Charalambous, "An asynchronous approximate distributed alternating direction method of multipliers in digraphs," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 3406–3413, doi: [10.1109/CDC45484.2021.9683229](https://doi.org/10.1109/CDC45484.2021.9683229).
- [43] K. Rokade and R. K. Kalaimani, "Distributed ADMM over directed graphs," 2020, *arXiv:2010.10421*.
- [44] V. Khatana and M. V. Salapaka, "D-distADMM: A $O(1/k)$ distributed ADMM for distributed optimization in directed graph topologies," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, pp. 2992–2997.
- [45] R. Diestel, *Graph Theory*. Berlin, Germany: Springer-Verlag, 2006.
- [46] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [47] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2015.
- [48] X. Wu, Z. Qu, and J. Lu, "A second-order proximal algorithm for consensus optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1864–1871, Apr. 2021.
- [49] V. Khatana and M. V. Salapaka, "DC-distADMM: ADMM algorithm for constrained distributed optimization over directed graphs," 2021, *arXiv:2003.13742*.
- [50] 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 *Proc. Int. Federation Autom. Control*, vol. 53, no. 2, 2020, pp. 2951–2956, [Online]. Available: http://box5779.temple.domains/jamesmel/wp-content/uploads/2019/11/Vector_Consensus_IFAC-1.pdf
- [51] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. IEEE 44th Annu. Symp. Foundations Comput. Sci.*, 2003, pp. 482–491.
- [52] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed strategies for average consensus in directed graphs," in *Proc. IEEE 50th Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 2124–2129.
- [53] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, 2010, pp. 1753–1757.
- [54] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, pp. 334–334, 1997.
- [55] H. Zhang and L. Cheng, "Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization," *Optim. Lett.*, vol. 9, no. 5, pp. 961–979, 2015.
- [56] F. Bach, "Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 595–627, 2014.
- [57] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3141–3155, Aug. 2019.
- [58] A. S. Berahas, R. Bollapragada, and E. Wei, "On the convergence of nested decentralized gradient methods with multiple consensus and gradient steps," *IEEE Trans. Signal Process.*, vol. 69, pp. 4192–4203, 2021, doi: [10.1109/TSP.2021.3094906](https://doi.org/10.1109/TSP.2021.3094906).
- [59] A. I.-A. Chen, "Fast distributed first-order methods," S.M. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2012.
- [60] H. Li, C. Fang, W. Yin, and Z. Lin, "A sharp convergence rate analysis for distributed accelerated gradient methods," 2018, *arXiv:1810.01053*.
- [61] H. Ye, L. Luo, Z. Zhou, and T. Zhang, "Multi-consensus decentralized accelerated gradient descent," 2020, *arXiv:2005.00797*.
- [62] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.
- [63] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Sub-gradient methods and consensus algorithms for solving convex optimization problems," in *Proc. IEEE 47th Conf. Decis. Control*, 2008, pp. 4185–4190.
- [64] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the ADMM with multiblock variables," *SIAM J. Optim.*, vol. 25, no. 3, pp. 1478–1497, 2015.
- [65] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Math. Operations Res.*, vol. 1, no. 2, pp. 97–116, 1976.
- [66] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [67] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
- [68] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.
- [69] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [70] M. Prakash, S. Talukdar, S. Attree, V. Yadav, and M. V. Salapaka, "Distributed stopping criterion for consensus in the presence of delays," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 85–95, Mar. 2020.

- [71] G. Saraswat, V. Khatana, S. Patel, and M. V. Salapaka, "Distributed finite-time termination for consensus algorithm in switching topologies," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: [10.1109/TNSE.2022.3216286](https://doi.org/10.1109/TNSE.2022.3216286).
- [72] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *Eur. J. Control*, vol. 18, no. 6, pp. 539–557, 2012.
- [73] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. control Optim.*, vol. 14, no. 5, pp. 877–898, 1976.



Vivek Khatana received the B.Tech degree in electrical engineering from the Indian Institute of Technology, Roorkee, Roorkee, India, in 2018. He is currently working toward the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA.

His research interests include distributed optimization, consensus algorithms, distributed control and stochastic calculus, power system analysis and control.



Murti V. Salapaka (Fellow, IEEE) received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology, Madras, Chennai, India, in 1991 and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1993 and 1997, respectively.

He was a Faculty Member with the Electrical and Computer Engineering Department, Iowa State University, Ames, IA, USA, from 1997 to 2007. He is currently the Director of Graduate Studies and the Vincentine Hermes Luh Chair Professor with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN, USA. His research interests include control and network science, nanoscience, and single molecule physics.

Dr. Salapaka was the recipient of the NSF CAREER Award and the ISU Young Engineering Faculty Research Award for the years 1998 and 2001.