Jiang, W.; Doostmohammadian, M.; Charalambous, T.

Distributed Resource Allocation via ADMM over Digraphs

Published in:
2022 IEEE 61st Conference on Decision and Control (CDC)

DOI:
10.1109/CDC51059.2022.9993326

Published: 10/01/2023

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Jiang, W., Doostmohammadian, M., & Charalambous, T. (2023). Distributed Resource Allocation via ADMM over Digraphs. In *2022 IEEE 61st Conference on Decision and Control (CDC)* (pp. 5645-5651). (Proceedings of the IEEE Conference on Decision & Control). IEEE. https://doi.org/10.1109/CDC51059.2022.9993326

Distributed Resource Allocation via ADMM over Digraphs

Wei Jiang, Mohammadreza Doostmohammadian and Themistoklis Charalambous

Abstract— **In this paper, we solve the resource allocation problem over a network of agents, with edges as communication links that can be unidirectional. The goal is to minimize the sum of allocation cost functions subject to a coupling constraint in a distributed way by using the finite-time consensus-based alternating direction method of multipliers (ADMM) technique. In contrast to the existing gradient descent (GD) based distributed algorithms, our approach can be applied to non-differentiable cost functions. Also, the proposed algorithm is initialization-free and converges at a rate of $\mathcal{O}(1/k)$, where k is the optimization iteration counter. The fast convergence performance related to iteration counter k compared to state-of-the-art GD based algorithms is shown via a simulation example.**

Index Terms— **Distributed optimization, ADMM, resource allocation, finite-time consensus, digraphs**

I. Introduction

Distributed resource allocation (DRA) optimally assigns a portion of existing resources to a group of cooperating agents (or nodes) based on local information exchange over a network. This is inspired by the recent development of distributed data-processing and parallel computing methods for large-scale data-mining and complex machine learning solutions. Applications include optimal energy management and economic dispatch over smart grid and energy networks [1], cost-optimal CPU scheduling over a network of servers [2], and optimal scheduling of the plug-in electric vehicles charging [3] among others. In terms of mathematical modeling, the problem is to optimize the sum of some local cost functions in a distributed fashion subject to a global linear coupling-constraint (referred to as the feasibility-constraint) and the local (convex) box constraints.

There are mainly two research lines in the literature to solve the DRA problem: *(i)* primal based solutions, e.g., gradient descent (GD) (also known as gradient-Laplacian) and *(ii)* dual based solutions, e.g., alternating direction method of multipliers (ADMM).

For primal based solutions, the GD algorithms typically ensure all-time feasibility given that the initial states are feasible [1], [4]–[6]. This primal feasibility requires the algorithm to be accompanied by specific initializations to satisfy the local box constraints and the global feasibility constraint simultaneously [1]. After the proper initialization, most GD based algorithms can guarantee feasibility at all optimization iterations (referred to as *anytime-feasibility*), which also means the output of the algorithm is feasible at any termination point while the cost is reduced at every iteration. Authors in [7] proposed an algorithm that converges to a point arbitrarily close to the optimal resource allocation Note that the most GD based algorithms (see, e.g., [1], [4]–[7]) require undirected or balanced graphs.

Dual based methods are mostly initialization-free, i.e., from any initial primal state, they can gain feasibility over time (asymptotically). There are some ADMM algorithms proposed for undirected graphs, e.g., the dual consensus ADMM algorithm in [8], which was extended in [9] for DRA problems considering a general nonempty, closed and convex cone rather than a convex equality constraint in [8]. Inspired by the dynamic average consensus [10] technique for gradient tracking in [11]–[15], authors in [3] proposed the tracking-ADMM algorithm by updating both the Lagrangian dual variable and the average of equality constraint in a distributed manner. It is worth noting that the existing ADMM based distributed algorithms in [3], [8], [9] are only applied over undirected communication graphs. Recently, distributed ADMM algorithms for unconstrained optimization problems over directed graphs (digraphs) are proposed in [16], [17].

In this work, we first reformulate the primal DRA problem by its Lagrangian dual and then, by assuming the local cost function of each node is convex, the strong duality between the Lagrangian dual and primal DRA problems holds, i.e., the duality gap is zero. After that, the distributed ADMM using finite-time exact ratio consensus (D-ADMM-FTERC) algorithm developed in [17] is adopted and modified to solve the Lagrangian dual problem. Therefore, we name the modified algorithm as DRA-ADMM-FTERC. Inside DRA-ADMM-FTERC, an additional min-max optimization problem emerges. We solve it by transforming the min-max into a max-min and then, solving the max and min optimization problems separately. In this way, the min-max optimization update is simplified (i.e., DRA-ADMM-FTERC is simplified) to be applied for wider cost functions. As D-ADMM-FTERC is developed for digraphs, inherently, so is DRA-ADMM-FTERC, advancing the recent results for DRA over undirected graphs [3], [7]–[9]. Unlike algorithms in [1], [7] which need specific initialization conditions, DRA-ADMM-FTERC is initialization-free (i.e., no need for the initial feasibility condition of primal states) and can have non-differential cost functions in the DRA primal problem. Based on authors' current knowledge, for solving DRA problems, this work is the first one to propose distributed ADMM based algorithms over digraphs.

W. Jiang and M. Doostmohammadian are with the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland. Email: wei.jiang@aalto.fi, mohammadreza.doostmohammadian@aalto.fi

T. Charalambous is with the Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus. He is also a Visiting Professor at the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland. Email: charalambous.themistoklis@ucy.ac.cy

II. Notation and Preliminaries

A. Notation

$\mathbb{R}, \mathbb{Z}, \mathbb{Z}_+$ and $\mathbb{R}^n$ denote the set of real, integer, positive integer numbers and the n-dimensional real space, respectively. A^{T} is the transpose of matrix A. $\mathbb{1}$ and I denote the all-ones vector and the identity matrix (of appropriate dimensions). We also denote by $e_j^{\mathrm{T}} = [0, \ldots, 0, 1_{j^{th}}, 0, \ldots, \ldots, 0] \in \mathbb{R}^{1\times n}$, where the single "1" entry is at the j^{th} position. $\|\cdot\|$ denotes the 2-norm.

The information exchange between nodes is captured by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order n with $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ being the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of edges. A directed edge from node v_i to node v_j is denoted by $\varepsilon_{ji} = (v_j, v_i) \in \mathcal{E}$ representing that node v_j to receive information from node v_i. A graph is said to be undirected if and only if $\varepsilon_{ji} \in \mathcal{E}$ implies $\varepsilon_{ij} \in \mathcal{E}$. A digraph is said to be *strongly* connected if there exists a path from each node v_i to each node v_j ($v_j \neq v_i$) in the graph. The graph diameter D is the longest shortest path between any two nodes in the network. We denote n' as an upper bound on n. All nodes that can transmit information to node v_j directly are said to be in-neighbors of node v_j and belong to the set $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid \varepsilon_{ji} \in \mathcal{E}\}$. The nodes that receive information from node v_j belong to the set of out-neighbors of node v_j, denoted by $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid \varepsilon_{lj} \in \mathcal{E}\}$.

B. Finite-Time Exact Ratio Consensus (FTERC)

Before we present FTERC algorithm, the well known average consensus and ratio consensus algorithms are needed. Consider a strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order n. Let w_j^t (for all $v_j \in \mathcal{V}$ and $t = 0, 1, 2, \ldots$) be the result of the average consensus iteration:

$$w_j^{t+1} = p_{jj} w_j^t + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} w_i^t\,, \quad t \geq 0, \tag{1}$$

where $w_j^0 \in \mathbb{R}$ is the initial state of node v_j. Denote $w^t := [w_1^t \;\; w_2^t \;\; \ldots \;\; w_n^t]^{\mathrm{T}}, P := [p_{ji}] \in \mathbb{R}^{n\times n}$ such that we can have the compact form of (1) as $w^{t+1} = Pw^t$, where $w^0 = [w_1^0 \;\; w_2^0 \;\; \ldots \;\; w_n^0]^{\mathrm{T}} \triangleq w_0$. If P is a primitive *doubly* stochastic matrix[1], then $\lim_{t\to\infty} w_j^t = \frac{1}{n}\sum_{v_i\in\mathcal{V}} w_i^0, \forall v_j \in \mathcal{V}$.

Asking matrix P to be primitive doubly stochastic is quite stringent and balancing algorithms [18] are needed to be implemented a priori. To alleviate this, the ratio consensus [19] algorithm is proposed as follows:

$$y_j^{t+1} = p_{jj} y_j^t + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} y_i^t, \tag{2a}$$

$$x_j^{t+1} = p_{jj} x_j^t + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} x_i^t, \tag{2b}$$

where the initial conditions are $y^0 = y_0$ and $x^0 = \mathbb{1}$, and $P = [p_{ji}] \in \mathbb{R}^{n\times n}$ is a primitive *column* stochastic matrix, then, we get $\lim_{t\to\infty} \mu_j^t = \lim_{t\to\infty} \frac{y_j^t}{x_j^t} = \frac{\sum_{v_i\in\mathcal{V}} y_i^0}{n}, \forall v_j \in \mathcal{V}$.

In what follows, we present the FTERC [20], [21] algorithm in which every node can compute $\mu_j \triangleq \lim_{t\to\infty} \mu_j^t$ in a *minimum* number of iteration steps.

Definition 1: (Minimal polynomial of a matrix pair) The minimal polynomial associated with the matrix pair $[P, e_j^{\mathrm{T}}]$ denoted by $q_j(s) = s^{M_j+1} + \sum_{i=0}^{M_j} \alpha_i^{(j)} s^i$ is the monic polynomial of minimum degree M_j+1 that satisfies $e_j^{\mathrm{T}} q_j(P) = 0$. Considering iteration (1) with weight matrix P, it is not difficult to have (e.g., using the techniques in [22])

$$\sum_{i=0}^{M_j+1} \alpha_i^{(j)} w_j^{t+i} = 0, \quad \forall t \in \mathbb{Z}_+, \tag{3}$$

where $\alpha_{M_j+1}^{(j)} = 1$. Denote $W_j(z) \triangleq \mathcal{Z}(w_j^t)$ as the z-transform of w_j^t. Based on the time-shift property of the $z-$transform, from (3) we obtain (see [22], [23])

$$W_j(z) = \frac{\sum_{i=1}^{M_j+1} \alpha_i^{(j)} \sum_{\ell=0}^{i-1} w_j^{\ell} z^{i-\ell}}{q_j(z)}, \tag{4}$$

where $q_j(z)$ is the minimal polynomial of $[P, e_j^{\mathrm{T}}]$. Under the strongly connected communication network, $q_j(z)$ does not have any unstable poles apart from one at 1; then, define the following polynomial:

$$p_j(z) \triangleq \frac{q_j(z)}{z-1} \triangleq \sum_{i=0}^{M_j} \beta_i^{(j)} z^i. \tag{5}$$

By applying the final value theorem [22], [23], we get

$$\phi_y(j) = \lim_{t\to\infty} y_j^t = \lim_{z\to 1}(z-1)Y_j(z) = \frac{y_{M_j}^{\mathrm{T}}\boldsymbol{\beta}_j}{\mathbb{1}^{\mathrm{T}}\boldsymbol{\beta}_j}, \tag{6a}$$

$$\phi_x(j) = \lim_{t\to\infty} x_j^t = \lim_{z\to 1}(z-1)X_j(z) = \frac{x_{M_j}^{\mathrm{T}}\boldsymbol{\beta}_j}{\mathbb{1}^{\mathrm{T}}\boldsymbol{\beta}_j}, \tag{6b}$$

where $y_{M_j}^{\mathrm{T}} = [y_j^0, y_j^1, \ldots, y_j^{M_j}]$, $x_{M_j}^{\mathrm{T}} = [x_j^0, x_j^1, \ldots, x_j^{M_j}]$ with $\boldsymbol{\beta}_j$ being the vector of coefficients of the polynomial $p_j(z)$.

Consider the following vectors of $2t+1$ successive discrete-time values at node v_j as

$$y_{2t}^{\mathrm{T}} = [y_j^0, y_j^1, \ldots, y_j^{2t}],$$
$$x_{2t}^{\mathrm{T}} = [x_j^0, x_j^1, \ldots, x_j^{2t}],$$

where y_j^t and x_j^t are two iterations in (2a) and (2b)), respectively. Define the corresponding Hankel matrices as:

$$\Gamma\{y_{2t}^{\mathrm{T}}\} \triangleq \begin{bmatrix} y_j^0 & y_j^1 & \ldots & y_j^t \\ y_j^1 & y_j^2 & \ldots & y_j^{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_j^t & y_j^{t+1} & \ldots & y_j^{2t} \end{bmatrix},$$

$$\Gamma\{x_{2t}^{\mathrm{T}}\} \triangleq \begin{bmatrix} x_j^0 & x_j^1 & \ldots & x_j^t \\ x_j^1 & x_j^2 & \ldots & x_j^{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_j^t & x_j^{t+1} & \ldots & x_j^{2t} \end{bmatrix}.$$

[1] A nonnegative matrix is such that all of its elements are nonnegative. A column (row) stochastic matrix is a real square nonnegative matrix, with each column (row) summing to 1. A doubly stochastic matrix is both row and column stochastic.

Denote the vector differences between successive values of y_j^t and x_j^t as: $\overline{y}_{2t}^{\mathrm{T}} = [y_j^1 - y_j^0, \ldots, y_j^{2t+1} - y_j^{2t}]$, $\overline{x}_{2t}^{\mathrm{T}} = [x_j^1 - x_j^0, \ldots, x_j^{2t+1} - x_j^{2t}]$.

It is shown in [23] that for arbitrary initial conditions y_0 and x_0, $\boldsymbol{\beta}_j$ can be computed as the kernel of the first defective Hankel matrices $\Gamma\{\overline{y}_{2t}^{\mathrm{T}}\}$ and $\Gamma\{\overline{x}_{2t}^{\mathrm{T}}\}$ (i.e., $\boldsymbol{\beta}_j$ can be the normalized kernel $\boldsymbol{\beta}_j = \begin{bmatrix}\boldsymbol{\beta}_j^0 & \boldsymbol{\beta}_j^1 & \ldots & \boldsymbol{\beta}_j^{x_{M_j}-1} & 1\end{bmatrix}^{\mathrm{T}}$ of the first defective Hankel matrix $\Gamma\{\overline{y}_{2t}^{\mathrm{T}}\}$), except a set of initial conditions with Lebesgue measure zero.

Next, with the strongly connected digraphs, the following Lemma 1 shows the exact average μ can be distributively obtained in finite-time.

Lemma 1 ([20], [21]): For all $v_j \in \mathcal{V}$ under a strongly connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and $t = 0, 1, 2, \ldots$, y_j^t and x_j^t are the iterations (2a) and (2b), where $P = [p_{ji}] \in \mathbb{R}^{n \times n}$ is formed as a primitive column stochastic weight matrix associated with the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Then, the average consensus problem can be distributively solved at each node v_j in finite-time, by computing

$$\mu_j \triangleq \lim_{t \to \infty} \frac{y_j^t}{x_j^t} = \frac{\phi_y(j)}{\phi_x(j)} = \frac{y_{M_j}^{\mathrm{T}} \boldsymbol{\beta}_j}{x_{M_j}^{\mathrm{T}} \boldsymbol{\beta}_j}, \tag{7}$$

where $\phi_y(j)$ and $\phi_x(j)$ are from (6a) and (6b), respectively and $\boldsymbol{\beta}_j$, as defined in (5), is the vector of coefficients.

C. Distributed FTERC in Networked Systems

From (7), we know $\boldsymbol{\beta}_j$ and M_j can be different for each node v_j. To implement FTERC for networked systems in a distributed way, all nodes need to know when to terminate ratio consensus (2). Before we describe Algorithm 1 that we proposed in [17] for distributed FTERC termination, we need the following assumptions.

Assumption 1: An upper bound on the number of nodes in the network n', i.e., $n' \geq n$, is known by each node $v_j \in \mathcal{V}$.

Assumption 2: The directed communication graph is strongly connected.

Though Assumption 1 is limiting, there exist distributed approaches to compute the network size; see, for example, [24]. Algorithm 1 guarantees that the ratio consensus iteration number at every step $k \geq 2$ is the *minimum* (see FTERC properties in Section II-B) and that the solution at every step is optimal. Fig. 1 describes the iteration number at each step. The details of Algorithm 1 are as follows:

1) When $k = 0$, node v_j computes y_j^1 via FTERC for $2n'$ steps. By that time, y_j^1 is guaranteed to be computed by each node, which requires computing $\boldsymbol{\beta}_j$ and as a consequence M_j is determined.
2) When $k = 1$, node v_j runs ratio consensus (2) for n' steps and computes y_j^2 with the same $\boldsymbol{\beta}_j$ computed at $k = 0$ (i.e., there is no need to compute the defective Hankel matrices again). It runs a $\max-$consensus[2] algorithm simultaneously with the initial condition $u_j^0 = M_j + 1$. Note that the $\max-$consensus algorithm converges in s iterations ($s \leq D < n'$). Hence, at this step node v_j, not only computes y_j^2, but also the maximum number ($t_{\max} := M_{\max} + 1$) of ratio consensus steps needed by each node v_j to compute their $y_j^{k+1}, k \geq 2$.
3) When $k \geq 2$, each node v_j computes y_j^{k+1} via ratio consensus (2) with the same $\boldsymbol{\beta}_j$ which runs for $t_{\max}$ iterations.

[2] The $\max-$consensus algorithm is used to compute the maximum value in a distributed fashion [25]. $\forall v_j \in \mathcal{V}$, the update is $u_j^{t+1} = \max_{v_i \in \mathcal{N}_j^- \cup \{v_j\}} \{u_i^t\}$ which converges to the maximum value among all nodes in a finite number of steps s, $s \leq D$ (see, e.g., [26]).

Algorithm 1 Distributed FTERC

Input: n' (upper bound on n)
Initialization: $k = 0$ and y_j^0 for node $v_j \in \mathcal{V}$
if $k = 0$ **then**
  Run FTERC for $2n'$ steps to compute y_j^1 and determine M_j and $\boldsymbol{\beta}_j$
else if $k = 1$ **then**
  Run $\max-$consensus to determine $M_{\max}$ from $M_j + 1, v_j \in \mathcal{V}$; run ratio consensus (2) for n' steps to compute y_j^2 with the same $\boldsymbol{\beta}_j$
else
  Run ratio consensus (2) with the same $\boldsymbol{\beta}_j$ for $t_{\max} := M_{\max} + 1$ steps to compute y_j^{k+1}
end if
Output: Node v_j obtains: $y_j^{k+1} = \frac{1}{n} \sum_{v_i \in \mathcal{V}} y_i^k$

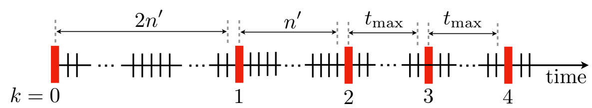

Fig. 1. Algorithm 1 diagram: FTERC is terminated after $2n'$ steps when $k = 1$ during which each node v_j computes M_j. When $k = 2$, it runs a $\max-$consensus algorithm and $M_{\max}$ is thus determined; at this stage the ratio consensus is terminated after n' steps. Thereafter, each ratio consensus algorithm running step is changed to $t_{\max} := M_{\max} + 1$.

III. Main Results

Recall the economic dispatch problem among n agents as [6], [7]

$$\begin{aligned} \min_{y_i \in \mathbb{R}} & \sum_{i=1}^n \phi_i(y_i), \\ \text{s.t.} & \sum_{i=1}^n y_i = b, \end{aligned} \tag{8}$$

where $\phi_i(y_i) : \mathbb{R} \to \mathbb{R}$ is the generation cost or the disutility related to shifting power demands; y_i represents the power injection of agent i (where negative injection $y_i < 0$ means that agent i consumes power from the grid); $b \in \mathbb{R}$ is the power injection not accounted for by the agents.

Motivated by this type of problem setting, we generalize the DRA problem studied in the paper as follows:

$$\begin{aligned} \min_{y_i \in \mathbb{R}^{p_i}} & \sum_{i=1}^n \phi_i(y_i), \\ \text{s.t.} & \sum_{i=1}^n (A_i y_i - b_i) = 0, \end{aligned} \tag{9}$$

where $\phi_i(y_i) : \mathbb{R}^{p_i} \to \mathbb{R}$ is a local cost function associated with agent i; $y_i \in \mathbb{R}^{p_i}$ represents the allocated resource to the agent i; the overall resources $\sum_{i=1}^{n} b_i$, $b_i \in \mathbb{R}^p$ is constant and the parameter matrices $A_i \in \mathbb{R}^{p\times p_i}$.

Assumption 3: The cost function $\phi_i : \mathbb{R}^{p_i} \to \mathbb{R}$ is closed, proper and convex. ϕ_i has at least one subgradient at y_i and $\phi_i(y_i) < \infty$. The minimum of DRA problem (9) is attained.

Assumption 3 allows ϕ_i to be non-differentiable [27].

A. Lagrangian Dual

Define the Lagrangian associated with DRA (9) as

$$L(y_i, x) = \sum_{i=1}^{n} \phi_i(y_i) + x^{\mathrm{T}} \sum_{i=1}^{n} (A_i y_i - b_i), \qquad (10)$$

where $x \in \mathbb{R}^p$ is the Lagrangian multiplier vector (or dual variable) associated with (9). We derive the Lagrangian dual function $f(x) = \inf_{y_i\in\mathbb{R}^{p_i}} L(y_i, x)$ as follows:

$$\begin{aligned} f(x) =& - x^{\mathrm{T}} \sum_{i=1}^{n} b_i + \inf_{y_i\in\mathbb{R}^{p_i}} \sum_{i=1}^{n} (\phi_i(y_i) + x^{\mathrm{T}} A_i y_i) \\ =& - x^{\mathrm{T}} \sum_{i=1}^{n} b_i - \sup_{y_i\in\mathbb{R}^{p_i}} \sum_{i=1}^{n} ((-A_i^{\mathrm{T}} x)^{\mathrm{T}} y_i - \phi_i(y_i)). \end{aligned}$$

Then, by using the *conjugate*[3], we have

$$f(x) = - x^{\mathrm{T}} \sum_{i=1}^{n} b_i - \sum_{i=1}^{n} \phi_i^*(-A_i^{\mathrm{T}} x). \qquad (11)$$

Since ϕ_i is convex from Assumption 3 and there is no inequality constraint in DRA problem (9), *Slater's condition* holds [28, Section 5.2.3] which means strong duality holds from Slater's theory, i.e., the duality gap is zero. As a result, the Lagrangian dual of DRA problem (9) is to maximize $f(x)$, which is

$$\max_{x\in\mathbb{R}^p} f(x) = -x^{\mathrm{T}} \sum_{i=1}^{n} b_i - \sum_{i=1}^{n} \phi_i^*(-A_i^{\mathrm{T}} x). \qquad (12)$$

Denote

$$f_i(x) = x^{\mathrm{T}} b_i + \phi_i^*(-A_i^{\mathrm{T}} x). \qquad (13)$$

We know $f_i(x)$ is convex based on Assumption 3. Then, the Lagrangian dual (12) can be changed to

$$\min_{x\in\mathbb{R}^p} \sum_{i=1}^{n} f_i(x). \qquad (14)$$

Remark 1: There are some existing GD based algorithms, e.g., [29] and [30], to solve problem (14) over a digraph satisfying Assumption 2. However, these algorithms require $f_i(x)$ in (13) to be differentiable. In other words, if $f_i(x)$ in (13) is non-differentiable, these algorithms are no longer applicable. Considering this point, we propose an ADMM based algorithm satisfying both Assumptions 2 and 3 (allowing $f_i(x)$ in (13) to be non-differentiable).

B. D-ADMM-FTERC Solution

In this subsection, we will use and modify D-ADMM-FTERC algorithm developed in [17] to solve the Lagrangian dual problem (14). Furthermore, since the duality gap is zero between Lagrangian dual problem (14) and DRA problem (9), we propose D-ADMM-FTERC based algorithm to address the DRA problem (9).

[3]Let $\phi(y) : \mathbb{R}^p \to \mathbb{R}$. The function $\phi^* : \mathbb{R}^p \to \mathbb{R}$ defined as $\phi^*(x) = \sup_{y\in\mathbb{R}^p} (x^{\mathrm{T}} y - \phi(y))$ is called the conjugate of the function $\phi(y)$.

In order to distributively solve (14) and to use the ADMM structure simultaneously, a separate decision variable x_i for node v_i is introduced and the constraint $x_i = x_j$ is imposed to guarantee that the node decision variables are equal. In this way, problem (14) is reformulated as

$$\begin{aligned} &\min_{X\in\mathbb{R}^{np}} \sum_{i=1}^{n} f_i(x_i), \\ &\text{s.t. } x_i = x_j, \forall v_i, v_j \in \mathcal{V}, \end{aligned} \qquad (15)$$

Where $X := [x_1^{\mathrm{T}}, x_1^{\mathrm{T}}, \ldots, x_n^{\mathrm{T}}]^{\mathrm{T}}$. Denote a closed nonempty convex set: $\mathcal{C} = \{[x_1^{\mathrm{T}}, x_1^{\mathrm{T}}, \ldots, x_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{np} \,|\, x_i = x_j\}$. By making variable $z \in \mathbb{R}^{np}$ as a copy of X, the problem (15) changes to

$$\begin{aligned} &\min_{X\in\mathbb{R}^{np}} \sum_{i=1}^{n} f_i(x_i), \\ &\text{s.t. } X = z,\, z \in \mathcal{C}. \end{aligned} \qquad (16)$$

Thus, by defining $g(z)$ as the indicator function of the set $\mathcal{C}$ in the following

$$g(z) = \begin{cases} 0, & \text{if } z \in \mathcal{C}, \\ \infty, & \text{otherwise}, \end{cases} \qquad (17)$$

the problem (16) becomes

$$\begin{aligned} &\min_{X\in\mathbb{R}^{np}, z\in\mathbb{R}^{np}} \sum_{i=1}^{n} f_i(x_i) + g(z), \\ &f_i(x_i) = \max_{y_i\in\mathbb{R}^{p_i}} (-x_i^{\mathrm{T}} y_i - \phi_i(y_i)) + x_i^{\mathrm{T}} b_i, \\ &\text{s.t. } X - z = 0, \end{aligned} \qquad (18)$$

where $f_i(x_i)$ comes from $f(x)$ in (13). For the convenience of notation, denote $F(X) = \sum_{i=1}^{n} f_i(x_i)$. Thus, denote the Lagrangian associated with (18) as

$$L(X, z, \lambda) = F(X) + g(z) + \lambda^T (X - z), \qquad (19)$$

with $\lambda \in \mathbb{R}^{np}$ being the Lagrange multiplier. By Denoting $z_i \in \mathbb{R}^p$ as the i-th element of vector z, at iteration k, the corresponding augmented Lagrangian of problem (18) is

$$\begin{aligned} &L_\rho(X^k, z^k, \lambda^k) \qquad (20) \\ =& \sum_{i=1}^{n} f_i(x_i^k) + g(z^k) + \lambda^{k^{\mathrm{T}}} (X^k - z^k) + \frac{\rho}{2} \|X^k - z^k\|^2 \\ =& \sum_{i=1}^{n} \left(f_i(x_i^k) + \lambda_i^{k^{\mathrm{T}}} (x_i^k - z_i^k) + \frac{\rho}{2} \|x_i^k - z_i^k\|^2 \right) + g(z^k), \\ &f_i(x_i^k) = \max_{y_i\in\mathbb{R}^{p_i}} (-\phi_i(y_i) - x_i^{k^{\mathrm{T}}} A_i y_i) + x_i^{k^{\mathrm{T}}} b_i. \end{aligned}$$

By ignoring terms that are unrelated to the decision variable of the minimization iteration (i.e., x_i, z), $\forall v_i \in \mathcal{V}$, the distributed ADMM updates become:

$$x_i^{k+1} = \arg\min_{x_i\in\mathbb{R}^p} f_i(x_i) + \lambda_i^{k^{\mathrm{T}}} x_i + \frac{\rho}{2} \|x_i - z_i^k\|^2, \qquad (21)$$

$$\begin{aligned} z^{k+1} =& \arg\min_{z\in\mathbb{R}^{np}} g(z) + \lambda^{k^{\mathrm{T}}} (X^{k+1} - z) + \frac{\rho}{2} \|X^{k+1} - z\|^2 \\ =& \arg\min_{z\in\mathbb{R}^{np}} g(z) + \frac{\rho}{2} \|X^{k+1} - z + \frac{1}{\rho} \lambda^k\|^2, \end{aligned} \qquad (22)$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1}), \qquad (23)$$

where $f_i(x_i)$ is defined in (18) and the last term in (22) is calculated from $2a^{\mathrm{T}}b + b^2 = (a+b)^2 - a^2$ with $a = \lambda^k/\rho$ and $b = X^{k+1} - z$.

For z^{k+1} update (22), based on the works in [16], [17], it is an average consensus problem and at each iteration k, we set $z_i^k = x_i^{k+1} + \lambda_i^k/\rho, \forall i$ as the input for distributed FTERC Algorithm 1 to get the output $z_i^{k+1} = \frac{1}{n}\sum_{j=1}^{n} z_j^k$ for each node i.

As we can see, λ_i^{k+1} update (23) is trivial. However, x_i^{k+1} update (21) is a min-max optimization problem which is not obvious to solve. Therefore, we show a solution in the following subsection.

C. Solution to x_i^{k+1} Update

Restructure x_i^{k+1} update (21) as

$$\begin{aligned}
x_i^{k+1} =& \arg\min_{x_i}\max_{y_i}\{-\phi_i(y_i) - x_i^{\mathrm{T}}A_iy_i + x_i^{\mathrm{T}}b_i + \lambda_i^{k\mathrm{T}}x_i \\
&+ \frac{\rho}{2}\|x_i\|^2 - \rho x_i^{\mathrm{T}}z_i^k + \frac{\rho}{2}\|z_i^k\|^2\} \\
=& \arg\min_{x_i}\max_{y_i}\{-\phi_i(y_i) - x_i^{\mathrm{T}}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k) \\
&+ \frac{\rho}{2}\|x_i\|^2\} \\
=& \arg\min_{x_i}\max_{y_i}\{-\phi_i(y_i) + \frac{\rho}{2}[\|x_i\|^2 - \frac{2x_i^{\mathrm{T}}}{\rho}(A_iy_i \\
&- b_i - \lambda_i^k + \rho z_i^k)]\} \\
=& \arg\min_{x_i}\max_{y_i}\{\frac{\rho}{2}[\|x_i - \frac{1}{\rho}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k)\|^2 \\
&- \|\frac{1}{\rho}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k)\|^2] - \phi_i(y_i)\}, \quad (24)
\end{aligned}$$

where $\frac{\rho}{2}\|z_i^k\|^2$ is dropped out from the second equality as it is not related to the min-max problem.

Since the min-max objective function (24) is convex in x_i for any given y_i and concave in y_i for any given x_i, based on the minimax theory in [31, Proposition 2.6.2], the min-max problem (24) can be transformed to the max-min problem as follows:

$$\begin{aligned}
y_i^{k+1} =& \arg\max_{y_i}\min_{x_i}\{\frac{\rho}{2}[\|x_i - \frac{1}{\rho}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k)\|^2 \\
&- \|\frac{1}{\rho}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k)\|^2] - \phi_i(y_i)\}. \quad (25)
\end{aligned}$$

Let y_i^{k+1} be the inner maximizer of min-max objective function (24) such that (x_i^{k+1}, y_i^{k+1}) is a saddle point[4] for (24). Therefore, based on [31, Proposition 2.6.2], (y_i^{k+1}, x_i^{k+1}) is the outer-inner solution to the max-min problem (25). As x_i^{k+1} is the inner solution to (25), max-min problem (25) can be changed to

$$y_i^{k+1} = \arg\max_{y_i}\{-\frac{\rho}{2}\|\frac{1}{\rho}(A_iy_i - b_i - \lambda_i^k + \rho z_i^k)\|^2 - \phi_i(y_i)\}, \quad (26)$$

by designing x_i^{k+1} as

$$x_i^{k+1} = \frac{1}{\rho}(A_iy_i^{k+1} - b_i - \lambda_i^k + \rho z_i^k). \quad (27)$$

[4]For the problem $\min_x \sup_y \phi(x,y)$, a pair of vectors (x^*, y^*) is called a saddle point of ϕ if $\phi(x^*, y) \le \phi(x^*, y^*) \le \phi(x, y^*)$ [31, Definition 2.6.1].

Algorithm 2 DRA-ADMM-FTERC

Input: n' (upper bound on n), $\rho > 0$, $k_{\max}$ (ADMM maximum iterating number)
Initialization: Node $v_i \in \mathcal{V}$ sets $x_i^0, z_i^0, \lambda_i^0$ and $k = 0$
Node $v_i \in \mathcal{V}$ does the following:
while $k \le k_{\max}$ **do**
  Compute y_i^{k+1} by Eq. (28)
  Compute x_i^{k+1} by Eq. (27)
  Set $z_i^k = x_i^{k+1} + \lambda_i^k/\rho$, put k, z_i^k as input to Algorithm 1 and get output $z_i^{k+1} = \frac{1}{n}\sum_{j=1}^{n} z_j^k$
  Compute λ_i^{k+1} by Eq. (23)
  if ADMM stopping criterion is satisfied **then**
    Stop DRA-ADMM-FTERC
  end if
  $k \leftarrow k + 1$
end while
Output: Node $v_i \in \mathcal{V}$ obtains the solution: y_i^*

What is more, y_i^{k+1} in (26) can be transformed equally to

$$y_i^{k+1} = \arg\min_{y_i}\{\frac{1}{2\rho}\|A_iy_i - b_i - \lambda_i^k + \rho z_i^k\|^2 + \phi_i(y_i)\}. \quad (28)$$

Remark 2: x_i^{k+1} update (21), which is a min-max problem, can be solved in the following two steps: (i) solve the min subproblem (28) to get y_i^{k+1} and (ii) update x_i^{k+1} using the closed form (27) with y_i^{k+1} obtained previously.

D. DRA-ADMM-FTERC Algorithm

Now, we are ready to summarize our Algorithm 2 to solve the DRA problem (9). We name the algorithm as DRA-ADMM-FTERC. Note that the ADMM terminates when the stopping criterion[5] is satisfied or the maximum number of optimization iteration, $k_{\max}$ is reached.

In the following theorem, we state that when the optimal x^* to the Lagrangian dual problem (14) is achieved asymptotically, the solution to the DRA problem (9) can also be achieved asymptotically.

Theorem 1: Under Assumptions 1-3, for DRA-ADMM-FTERC Algorithm 2, when $k \to \infty, \forall i, x_i^k \to x^*$ which is the optimal solution to the Lagrangian dual (14). When $k \to \infty, (y_1^k, y_2^k, \ldots, y_n^k)$ is the primal optimal to DRA problem (9) and converges at a rate of $\mathcal{O}(1/k)$.

Proof: To solve the Lagrangian dual problem (14), the D-ADMM-FTERC from [17] is used. Thus, based on [17, Theorem 2], the ergodic average of x_i^k converges to x^*, i.e., $\frac{1}{k}\sum_{s=1}^{k} x_i^s \to x^*$ at a rate of $\mathcal{O}(1/k)$, which also means $x_i^k \to x^*, k \to \infty, \forall i$.

Inspired by [8, Theorem 2], in order to prove $(y_1^k, y_2^k, \ldots, y_n^k), k \to \infty$ is the primal optimal to DRA problem (9), we need to prove that

$$\partial\phi_i(y_i^k) + A_i^{\mathrm{T}}x^* \to 0, k \to \infty, \forall i, \quad (29)$$

$$\sum_{i=1}^{n}(A_iy_i^k - b_i) \to 0, k \to \infty, \quad (30)$$

[5]The stopping criterion can be referred to [27, Section 3.3.1].

where $\partial\phi_i(y_i^k)$ is the subgradient of function ϕ_i at y_i^k and (29) comes from the Lagrangian (10) associated with DRA problem (9). Recall the y_i^{k+1} update (28) which is a minimization problem, than, we get

$$0 = \partial\phi_i(y_i^{k+1}) + \frac{1}{\rho}A_i^{\mathrm{T}}(A_i y_i^{k+1} - b_i - \lambda_i^k + \rho z_i^k), \forall k \geq 0.$$

Based on the x_i^{k+1} update (27), we have

$$\partial\phi_i(y_i^k) + A_i^{\mathrm{T}} x_i^k = 0, \forall k \geq 1. \quad (31)$$

As it is proved previously that $x_i^k \to x^*, k \to \infty$, (29) can be proved.

In order to prove (30), from x_i^{k+1} update (27), we obtain

$$\rho \sum_{i=1}^{n} x_i^{k+1} = \sum_{i=1}^{n}(A_i y_i^{k+1} - b_i) + \sum_{i=1}^{n}(\rho z_i^k - \lambda_i^k). \quad (32)$$

From step 7 of DRA-ADMM-FTERC Algorithm 2, we know $z_i^{k+1} = \frac{1}{n}\sum_{v_i \in \mathcal{V}}(x_i^{k+1} + \lambda_i^k/\rho)$, which is also

$$\rho \sum_{i=1}^{n} z_i^{k+1} = \sum_{i=1}^{n}(\rho x_i^{k+1} + \lambda_i^k). \quad (33)$$

Integrating (33) into (32) we get

$$\sum_{i=1}^{n}(A_i y_i^k - b_i) = \rho \sum_{i=1}^{n}(z_i^k - z_i^{k-1}). \quad (34)$$

Similar as $x_i^k \to x^*, k \to \infty, \forall i$, from [17, Theorem 2], the ergodic average of z_i^k converges to $\bar{z}^*$ (we denote $\bar{z}^*$ as the optimal for z_i^{k+1} update of each node i), i.e., $\frac{1}{k}\sum_{s=1}^{k} z_i^s \to \bar{z}^*$ at a rate of $\mathcal{O}(1/k)$, which means $z_i^k \to \bar{z}^*, k \to \infty, \forall i$. As a result, (30) is proved.

In addition, based on the relationship between y_i^k and x_i^k in (27), since the ergodic average of x_i^k converges to x^* at a rate of $\mathcal{O}(1/k)$, the ergodic average of y_i^k also converges to its optimal at a rate of $\mathcal{O}(1/k)$. ■

Remark 3: We need Assumption 1 to run DRA-ADMM-FTERC. Note that n' ($n' \geq n$) in Assumption 1 is a piece of global information, i.e., DRA-ADMM-FTERC is a distributed optimization method, not a decentralized one. Though there exist distributed solutions for computing n', e.g., [24], inside DRA-ADMM-FTERC (Algorithm 2), we can replace FTERC (Algorithm 1) to be finite-time distributed termination (FTDT) algorithm in [32, Section II-F] which does not require any global information. In such a way, DRA-ADMM-FTDT will become a decentralized optimization method.

IV. Examples

Consider the following convex cost function satisfying Assumption 3 borrowed from [6], [33]

$$\min_{y_i \in \mathbb{R}} \Phi(\mathbf{y}) = \sum_{i=1}^{n} \phi_i(y_i),\ \phi_i(y_i) = w_i(y_i - a_i)^4 \quad (35)$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$ with parameters $w_i \in (0, 4]$, $a_i \in [-2, 4]$ randomly. We set $n = 6, A_i = I, \forall i$ and the equality constraint parameter $b_i = i, i = 1, 2, \ldots, n$. As a result, $\sum_{i=1}^{n} b_i = 21$. The centralized CVX is used to find the optimal solution $\mathbf{y}^* = [y_1^*, y_2^*, \ldots, y_n^*]^{\mathrm{T}} = [4.6585, 5.7118, -0.0153, 1.5717, 4.6335, 4.4399]^{\mathrm{T}}$.

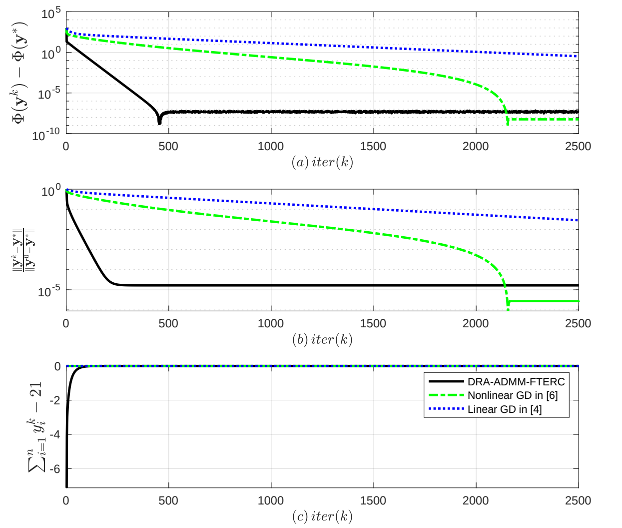

Fig. 2. Algorithm performance comparison: (a) objective function residual; (b) normalized primal variable residual; (c) feasibility check.

We adopt algorithms in [4], [6] for comparison with DRA-ADMM-FTERC. In detail, for the nonlinear GD algorithm in [6], we set $v_1 = 0.4$, $v_2 = 1.4$, $\eta = 0.0002$ (without quantization); for the linear GD algorithm in [4], we set $\eta = 0.0002$. For the sake of comparison, we use the undirected cycle network with symmetric and doubly stochastic weights. In general, our DRA-ADMM-FTERC algorithm can work for directed strongly connected graphs and non-differential objective functions.

Fig. 2 shows the performance comparison among DRA-ADMM-FTERC, nonlinear GD in [6] and linear GD in [4]. One can see DRA-ADMM-FTERC converges faster from Fig. 2 (a) and (b). The feasibility condition check is demonstrated in Fig. 2 (c). The GD based algorithms in [4], [6] need the feasibility condition $\sum_{i=1}^{n} y_i - 21 = 0$ to be satisfied initially and can preserve solution-feasibility over time. DRA-ADMM-FTERC algorithm, on the other hand, does not require initial feasibility (i.e., the initial condition can be random) and can reach feasibility constraint asymptotically, which is one improvement over the GD based algorithms.

Fig. 3 (a), (b), (c) describe the primal variable y_i^k, dual variable x_i^k and z_i^k (as a copy of x_i^k), respectively. For the convenience of presentation, we choose k from 1 to 300. One can see y_i^k reaches $\mathbf{y}^*$ and we have $x_i^k = z_i^k, \forall i$ asymptotically. It is worth noting that $z_i^k = z_j^k, \forall i, j, k$ from Fig. 3 (c), which demonstrates the effectiveness of Algorithm 1 used in step 7 of DRA-ADMM-FTERC Algorithm 2.

V. Conclusions

A distributed algorithm is proposed to solve the resource allocation problem over directed graphs. Thanks to the nature of ADMM algorithm, our proposed ADMM based solution can be applied to non-differential objective functions and has no requirement for variable initialization compared to some gradient descent ones. Furthermore, the fast convergence

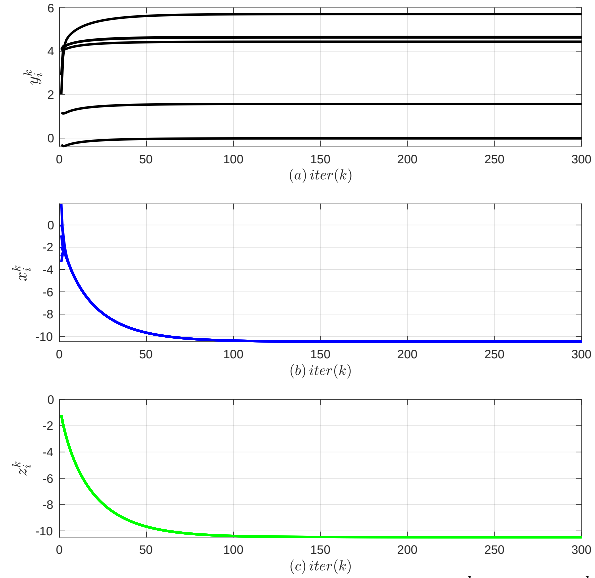

Fig. 3. DRA-ADMM-FTERC variables: (a) primal y_i^k; (b) dual x_i^k; (c) z_i^k update from distributed FTERC Algorithm 1.

performance related to the optimization iteration counter is demonstrated via a comparison example.

Future work will be validating the performance of proposed algorithm on large-scale systems, targeting non-convex objective functions and considering more (possible non-convex) constraints such as additional inequality constraints.

References

[1] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE*

[2] A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, T. Charalambous, and K. H. Johansson, "Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism," in *Proc. 60th IEEE Conf. Dec. Control*, 2021, pp. 6276–6281.

[3] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[4] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Optimiz. Theory App.*, vol. 129, no. 3, pp. 469–488, 2006.

[5] E. Ghadimi, M. Johansson, and I. Shames, "Accelerated gradient methods for networked optimization," in *Proc. Amer. Control Conf.* IEEE, 2011, pp. 1668–1673.

[6] M. Doostmohammadian, A. Aghasi, M. Pirani, E. Nekouei, U. A. Khan, and T. Charalambous, "Fast-convergent anytime-feasible dynamics for distributed allocation of resources over switching sparse networks with quantized communication links," in *Proc. Eur. Control Conf.*, 2022, pp. 84–89.

[7] X. Wu, S. Magnusson, and M. Johansson, "A new family of feasible methods for distributed resource allocation," in *Proc. IEEE Conf. Dec. Control*, 2021, pp. 3355–3360.

[8] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Proces.*, vol. 63, no. 2, pp. 482–497, 2014.

[9] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus ADMM," in *Proc. Amer. Control Conf.* IEEE, 2019, pp. 2789–2794.

[10] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[11] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, 2018.

Trans. Control Netw. Syst., vol. 2, no. 3, pp. 226–237, 2015.

[12] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 994–1009, 2015.

[13] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, 2017.

[14] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1329–1339, 2017.

[15] 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.

[16] V. Khatana and M. V. Salapaka, "D-DistADMM: A $O(1/k)$ distributed ADMM for distributed optimization in directed graph topologies," in *Proc. 59th IEEE Conf. Dec. Control*, Dec. 2020, pp. 2992–2997.

[17] 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.

[18] A. Rikos, T. Charalambous, and C. Hadjicostis, "Distributed weight balancing over digraphs," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 190–201, June 2014.

[19] A. D. Domínguez-García and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 537–542.

[20] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Decentralised minimum-time average consensus in digraphs," in *Proc. 52nd IEEE Conf. Dec. Control*, 2013, pp. 2617–2622.

[21] ——, "Distributed finite-time average consensus in digraphs in the presence of time delays," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 4, pp. 370–381, 2015.

[22] Y. Yuan, G.-B. Stan, L. Shi, and J. Gonçalves, "Decentralised final value theorem for discrete-time LTI systems with application to minimal-time distributed consensus," in *Proc. 48th IEEE Conf. Dec. Control*, Dec. 2009, pp. 2664–2669.

[23] Y. Yuan, G.-B. Stan, M. Barahona, L. Shi, and J. Gonçalves, "Decentralised minimal-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227–1235, May 2013.

[24] I. Shames, T. Charalambous, C. Hadjicostis, and M. Johansson, "Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs," in *50th Ann. Allerton Conf. Commun., Control, and Computing*, Oct. 2012, pp. 1885–1892.

[25] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, pp. 726–737, March 2008.

[26] S. Giannini, D. Di Paola, A. Petitti, and A. Rizzo, "On the convergence of the max-consensus protocol with asynchronous updates," in *Proc. 52nd IEEE Conf. Dec. Control*, 2013, pp. 2605–2610.

[27] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[28] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 315–320, 2018.

[30] S. Pu, W. Shi, J. Xu, and A. Nedic, "Push-pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, 2020.

[31] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[32] W. Jiang and T. Charalambous, "Fully distributed alternating direction method of multipliers in digraphs via finite-time termination mechanisms," *arXiv preprint arXiv:2107.02019*, 2021.

[33] T. T. Doan and A. Olshevsky, "Distributed resource allocation on dynamic networks in quadratic time," *Syst. Control Lett.*, vol. 99, pp. 57–63, 2017.