

Distributed Optimization with Gradient Tracking over Heterogeneous Delay-Prone Directed Networks

Evagoras Makridis¹, Gabriele Oliva², Kasagatta Ramesh Narahari³, Mohammadreza Doostmohammadian⁴, Usman A. Khan⁵, and Themistoklis Charalambous^{1,6,7}

Abstract—In this paper, we address the distributed optimization problem over unidirectional networks with possibly time-invariant heterogeneous bounded transmission delays. In particular, we propose a modified version of the Accelerated Distributed Directed OPTimization (ADD-OPT) algorithm, herein called *Robustified ADD-OPT* (R-ADD-OPT), which is able to solve the distributed optimization problem, even when the communication links suffer from heterogeneous but bounded transmission delays. We show that if the gradient step-size of the R-ADD-OPT algorithm is within a certain range, which also depends on the maximum time delay in the network, then the nodes are guaranteed to converge to the optimal solution of the distributed optimization problem. The range of the gradient step-size that guarantees convergence can be computed *a priori* based on the maximum time delay in the network.

Index Terms—distributed optimization, bounded delays, gradient tracking, unidirectional networks, directed graphs.

I. INTRODUCTION

Recently, there has been an increasing number of multi-agent optimization problems where a group of agents aims at optimizing the sum of their objective functions, by allowing local information exchange over an underlying communication network. Such formulation arises in cases where each agent has a local objective and the objective is to reach a compromise without resorting to central coordination. Such a distributed formulation is extensively used to solve multi-node optimization problems for several large-scale applications, including distributed energy resource management [1], [2], resource allocation in data centers [3], large-scale machine learning [4], [5], and distributed localization in sensor networks [6], [7].

In particular, in a multi-agent system comprised of a set of n agents communicating over a network, each agent j , represented by a node in a graph, has access only to

its own local objective function $f_j : \mathbb{R} \rightarrow \mathbb{R}$, and aims at minimizing the sum of objectives $\sum_{j=1}^n f_j(z)$, where the common decision variable¹ $z \in \mathbb{R}$ is computed via information exchange and agreement with its neighbors. More specifically, the agents aim at solving cooperatively the following optimization problem:

$$\min_z \frac{1}{n} \sum_{j=1}^n f_j(z). \quad (1)$$

Over the past decade, considerable research efforts have been dedicated in consensus-based distributed optimization problems over networks with bidirectional links, forming undirected graphs. Such contributions have been extensively studied and presented in [8]–[10]. These works exploit the symmetry of bidirectional communication structure to enable the construction of doubly-stochastic consensus weight matrices, so that local decision variables of networked agents collaboratively converge towards the solution of the distributed optimization problem in (1).

While initial efforts yielded valuable insights and algorithms, attention has shifted to more realistic scenarios with unidirectional network links, where the information might flow in only one direction. This attribute is prevalent in wireless communication due to varying transmission power and interference levels. Such (possibly asymmetric) networks are represented as directed graphs (digraphs), where directed edges represent unidirectional communication links. To address the distributed optimization problem over unidirectional links, there have been different approaches that are based on the following distributed average consensus methods: (a) the distributed weight balancing [11] where agents adjust the weights on their outgoing links to asymptotically reach weights that form a doubly-stochastic matrix, (b) the surplus-based approach [12] where the asymmetry of the digraph is mitigated by augmenting the state of each agent with an additional variable (often called *surplus*) that locally tracks individual state updates through the assignment of weights on both the incoming and outgoing links that form both row-stochastic and column-stochastic matrices, respectively, and (c) ratio consensus [13] where agents run an extra concurrent iteration that asymptotically learns the right eigenvector of the column-stochastic matrix that is formed by having agents assign weights on their outgoing links. For more

¹Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus. E-mails: {makridis.evagoras, charalambous.themistoklis}@ucy.ac.cy.

²Department of Engineering, University Campus Bio-Medico of Rome, Roma, Italy. E-mail: g.oliva@unicampus.it.

³Nokia, Espoo, Finland. E-mail: naraharikasagattaramesh@gmail.com.

⁴School of Engineering, Semnan University, Semnan, Iran. E-mail: doost@semnan.ac.ir.

⁵Department of Electrical and Computer Engineering, Tufts University, Medford, Massachusetts, United States. E-mail: khan@ece.tufts.edu.

⁶Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland.

⁷FinEst Centre for Smart Cities, Tallinn, Estonia.

The work of E. Makridis and T. Charalambous was partly supported by the European Research Council (ERC) Consolidator Grant MINERVA (Grant agreement No. 1011044629).

¹To simplify the presentation and analysis, in this work, we use a scalar decision variable. However, the results presented herein can be readily extended to vector variables.

information and comparison between these distributed average consensus methods over digraphs, the interested reader is referred to [14].

In the context of distributed optimization over reliable and time-invariant digraphs, Xi *et. al.* introduced two algorithms based on the ratio consensus protocol: the Directed EXact first-order Algorithm (DEXTRA) [15] and ADD-OPT [16] that incorporates *gradient tracking* to record and correct the locally aggregated gradients at each agent, and accelerate the convergence to the optimal solution. Moreover, Nedich *et. al.* in [17], [18] proposed the Subgradient-push and Push-DIGing algorithms for time-varying digraphs, based on the push-sum consensus algorithm. Using the surplus-consensus method, Xi *et. al.* introduced the Directed-Distributed Projected Subgradient (D-DPS) and Directed-Distributed Gradient Descent (D-DGD) algorithm, respectively, to solve distributed optimization problems over digraphs, in [19] and [20], respectively.

Although the aforementioned distributed optimization algorithms can successfully overcome the asymmetry of unidirectional networks and potentially find the optimal solution in (1), none of them considered unreliable networks where the information flow between agents might be delayed. This factor arises inherently as the number of agents in the network grows, and hence a corresponding increase in the quantity of communication links can occur. This surge of links often results in information congestion within the network, giving rise to transmission delays that exert a significant degradation on the performance and efficacy of distributed optimization algorithms. Extensive work has been done, towards the development of mitigation strategies for addressing the challenges posed by congested networks, including the transmission delays encountered over communication links. These challenges have been analysed in the works presented in [21], [22], in the context of distributed average consensus. Interestingly, in the context of distributed optimization, an early work in [23] by Tsianos *et. al.* analyzed a push-sum consensus-based distributed optimization algorithm for digraphs that can handle fixed bounded communication delays. However, this work lacks the analysis of convergence, as well as the incorporation of gradient tracking, a crucial feature known for expediting the convergence to the optimal solution [24]–[26].

The aforementioned works illustrate the promising potential in the study and development of communication-aware strategies tailored to overcome the challenges posed by communication delays in the context of consensus-based distributed optimization algorithms. To the best of our knowledge, our work is the first to propose and analyze a distributed optimization algorithm with gradient tracking over directed graphs in the presence of time-invariant delays, as highlighted in Table I.

In this work, we tackle the problem of distributed optimization over directed graphs, where the transmission links are prone to heterogeneous communication delays.

TABLE I: Gradient-based distributed optimization algorithms over directed graphs (k denotes the number of algorithm iterations, and $0 < \mu < 1$ the convexity constant, introduced in the subsequent section).

Algorithm	Delays	Converg. Rate	Grad. Tracking
D-DGD [20]	–	$\mathcal{O}\left(\frac{\ln k}{\sqrt{k}}\right)$	–
D-DPS [19]	–	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	–
S-Push [17]	–	$\mathcal{O}\left(\frac{\ln k}{\sqrt{k}}\right)$	–
PS-DDA [23]	fixed	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	–
DEXTRA [15]	–	$\mathcal{O}(\mu^k)$	✓
ADD-OPT [16]	–	$\mathcal{O}(\mu^k)$	✓
Push-DIGing [18]	–	$\mathcal{O}(\mu^k)$	✓
R-ADD-OPT	fixed	$\mathcal{O}(\mu^k)$	✓

Specifically, we consider how ADD-OPT algorithm [16] can be modified in order to operate even in the presence of heterogeneous communication delays. Towards this end, we develop a framework that seamlessly integrates handling mechanisms of heterogeneous delays induced in the transmission links, by embedding the robustified Ratio Consensus protocol in the classical ADD-OPT algorithm. The proposed algorithm, herein called the Robustified ADD-OPT (R-ADD-OPT) algorithm, enables nodes that communicate over strongly connected digraphs, to converge to the optimal solution of the distributed optimization problem, even in the presence of heterogeneous delays. In our theoretical analysis, we demonstrate that the convergence of nodes’ decision variables to the optimal solution remains assured, given that the gradient step-size is within a specific range. The range for which the algorithm is guaranteed to converge to the optimal solution is dependent on the maximum delay (provided it remains bounded) existing in the network. The proof of convergence is based on augmenting the network’s corresponding weighted adjacency matrix, to handle bounded time-invariant delays. It is conjectured via simulations that the R-ADD-OPT algorithm works also for bounded time-varying delays, for a gradient step-size within a specific range, upper bounded by the step-size for the maximum time-invariant delay. However, the proof for this (or a tighter) range remains an open problem.

II. PRELIMINARIES

A. Notation

In this paper, we denote vectors by lowercase bold letters, and matrices by uppercase italic letters. The $n \times n$ identity matrix is denoted by I_n , and the n dimensional column vectors of all ones and zeros are represented by $\mathbf{1}_n$ and $\mathbf{0}_n$, respectively. The Kronecker product of two matrices A and B is denoted by $A \otimes B$. For any $f(\mathbf{x})$, $\nabla f(\mathbf{x})$ denotes the gradient of f at \mathbf{x} . The spectral radius of a matrix A is represented by $\rho(A)$. The right and left eigenvectors of an irreducible column-stochastic matrix A , corresponding to the eigenvalue of 1, are denoted by $\boldsymbol{\pi}$ and $\mathbf{1}_n^\top$, respectively, such that $\mathbf{1}_n^\top \boldsymbol{\pi} = 1$. By $\|\cdot\|$, and depending

on its argument, we denote a particular matrix norm, or a vector norm that is compatible with this particular matrix norm, *i.e.*, $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$ for all matrices, A , and all vectors, \mathbf{x} . The Euclidean norm of vectors and matrices is denoted by $\|\cdot\|_2$.

B. Network Model

Consider n agents (represented by graph's nodes) communicating over a *strongly connected network*², $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (representing the communication links between agents). The total number of edges in the network is denoted by $m = |\mathcal{E}|$. A directed edge $\varepsilon^{ji} \triangleq (v_j, v_i) \in \mathcal{E}$, where $v_j, v_i \in \mathcal{V}$, represent that node v_j can receive information from node v_i , *i.e.*, $v_i \rightarrow v_j$. The nodes that transmit information to node v_j directly are called in-neighbors of node v_j , and belong to the set $\mathcal{N}_j^{\text{in}} = \{v_i \in \mathcal{V} | \varepsilon^{ji} \in \mathcal{E}\}$, with its cardinality denoted by $d_j^{\text{in}} = |\mathcal{N}_j^{\text{in}}|$ called the in-degree. The nodes that receive information from node v_j directly are called out-neighbors of node v_j , and belong to the set $\mathcal{N}_j^{\text{out}} = \{v_l \in \mathcal{V} | \varepsilon_{lj} \in \mathcal{E}\}$, with its cardinality $d_j^{\text{out}} = |\mathcal{N}_j^{\text{out}}|$ called the out-degree. Note that self-loops are included in digraph \mathcal{G} and this implies that the number of in-going links of node v_j are $(d_j^{\text{in}} + 1)$ and similarly the number of its out-going links is $(d_j^{\text{out}} + 1)$.

C. Robustified Ratio Consensus over Directed Graphs

To reach an agreement on the common decision variable z in (1) in a distributed way and over digraphs, one can employ the *Ratio Consensus* protocol [27] such that all nodes in the network converge to the network-wide average of their initial values. A protocol that handles time-varying (yet bounded) communication delays to ensure asymptotic average consensus was proposed in [21]. Consider that node v_j undergoes an *a priori* unknown delay τ^{ji} bounded by a positive integer $\bar{\tau}^{ji}$, *i.e.*, $\tau^{ji} \leq \bar{\tau}^{ji} < \infty$. The maximum delay in the network is denoted by $\bar{\tau} = \max\{\bar{\tau}^{ji}\}$. The own value of node v_j is always instantly available without delay, *i.e.*, $\tau^{jj} = 0$. At each time step k , each node v_j maintains a state variable $x_k^j \in \mathbb{R}$ (initialized at $x_0^j = V^j$, where V^j is an arbitrary value of node v_j), an auxiliary scalar variable, $y_k^j \in \mathbb{R}^+$ (initialized at $y_0^j = 1$), and $z_k^j \in \mathbb{R}$ set to $z_k^j = x_k^j / y_k^j$. Based on this notation, the strategy proposed in [21] involves each node iteratively updating its states according to:

$$x_{k+1}^j = p_{jj}x_k^j + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} p_{ji}x_{k-\tau^{ji}}^i, \quad (2a)$$

$$y_{k+1}^j = p_{jj}y_k^j + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} p_{ji}y_{k-\tau^{ji}}^i, \quad (2b)$$

where $P = \{p_{ji}\} \in \mathbb{R}_+^{n \times n}$ forms a nonnegative column-stochastic matrix, since the weights p_{lj} are assigned using:

$$p_{lj} = \begin{cases} \frac{1}{1+d_j^{\text{out}}}, & v_l \in \mathcal{N}_j^{\text{out}} \cup \{v_j\}, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

²A digraph \mathcal{G} is strongly connected if for every pair of vertices v_i and $v_j \in \mathcal{V}$, v_j is reachable by a directed walk from v_i

Since each node v_j assigns the weights according to (3), it is required that the nodes have the knowledge of their out-degree. Clearly, in the absence of delays, this strategy reduces to the Ratio Consensus algorithm in [27]. By construction, matrix P is primitive column-stochastic, and the variables y_k^j are set to 1. Then, the limit of the ratio x_k^j over y_k^j , is the average of the initial values and is given by [27], [28]:

$$\lim_{k \rightarrow \infty} z_k^j = \lim_{k \rightarrow \infty} \frac{x_k^j}{y_k^j} = \frac{(\sum_{i=1}^n x_0^i) \pi^j}{n \pi^j} = \frac{1}{n} \sum_{i=1}^n x_0^i. \quad (4)$$

D. Assumptions

Prior introducing the R-ADD-OPT algorithm, we make the following assumptions:

Assumption 1: (Strong connectivity)

The communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a strongly connected digraph.

Assumption 2: (Out-neighborhood knowledge)

Each node $v_j \in \mathcal{V}$ knows its out-degree, *i.e.*, the number of out-neighboring nodes.

Assumption 3: (Lipschitz-continuous gradients and strong convexity)

Each local function f_i is differentiable, strongly convex, and its gradient is globally Lipschitz-continuous, *i.e.*, for any i and $z_1, z_2 \in \mathbb{R}$:

a) there exists a positive constant L such that

$$\|\nabla f_i(z_1) - \nabla f_i(z_2)\| \leq L \|z_1 - z_2\|,$$

b) there exists a positive constant μ such that

$$f_i(z_1) - f_i(z_2) \leq \nabla f_i(z_1)^\top (z_1 - z_2) - \frac{\mu}{2} \|z_1 - z_2\|^2.$$

Notably strong convexity of the functions f_i imply that (1) has a unique and bounded global optimal solution z^* .

Assumption 4: (Time-invariant heterogeneous delays)

The delay on link ε^{ji} is denoted by $0 \leq \tau^{ji} \leq \bar{\tau}^{ji} \leq \bar{\tau}$, where $\bar{\tau}^{ji}$ is the maximum delay on link ε^{ji} , and $\bar{\tau}$ is the maximum delay of all links in the digraph, *i.e.*, $\bar{\tau} = \max\{\bar{\tau}^{ji}\}$.

III. ACCELERATED DISTRIBUTED OPTIMIZATION OVER DELAYED UNIDIRECTIONAL NETWORKS

In this section, we mitigate the impediments of induced communication delays, by embedding delay handling mechanisms into the ADD-OPT algorithm. However, such a strategy does not guarantee the convergence of network nodes' decision variables to the optimal solution of problem (1) when assuming a gradient step-size within the range for the original ADD-OPT algorithm. Hence, to guarantee the convergence of the R-ADD-OPT algorithm to the optimal solution, we provide a step-size analysis by computing the allowable step-size range which depends on the maximum delay in the network, given that it is bounded.

A. Robustified ADD-OPT Algorithm (R-ADD-OPT)

We first introduce a modified version of the ADD-OPT algorithm that handles heterogeneous time-invariant, hereinafter called R-ADD-OPT. In particular, the Robustified

Ratio Consensus algorithm is embedded into the ADD-OPT algorithm to ensure that the local decision variables z^j reach the average consensus value z regardless the transmission delays on the links, while the ADD-OPT algorithm shifts the coordinated value z to the optimal solution of the optimization problem z^* . Similarly to the original ADD-OPT algorithm, each node $v_j \in \mathcal{V}$ maintains three scalar variables, x_k^j, y_k^j, w_k^j all $\in \mathbb{R}$, and sets $z_k^j = x_k^j / y_k^j$. At each iteration k , node v_j assigns the weights to its states, and iteratively updates its variables at each time step $k \geq 0$, according to:

$$\begin{aligned} x_{k+1}^j &= \sum_{v_i \in \tilde{\mathcal{N}}_j^{\text{in}}} (p_{ji} x_{k-\tau}^{ji} - \alpha w_{k-\tau}^{ji}) - \alpha w_k^j, \\ y_{k+1}^j &= \sum_{v_i \in \tilde{\mathcal{N}}_j^{\text{in}}} p_{ji} y_{k-\tau}^{ji}, \\ z_{k+1}^j &= \frac{x_{k+1}^j}{y_{k+1}^j}, \\ w_{k+1}^j &= \sum_{v_i \in \tilde{\mathcal{N}}_j^{\text{in}}} p_{ji} w_{k-\tau}^{ji} + \nabla f_j(z_{k+1}^j) - \nabla f_j(z_k^j), \end{aligned} \quad (5)$$

where $\tilde{\mathcal{N}}_j^{\text{in}} = \{\mathcal{N}_j^{\text{in}} \cup \{v_j\}\}$, and each node is initialized with $y_0^j = 1$, $w_0^j = \nabla f_j(z_0^j)$, and arbitrary scalar values for x_0^j , and z_0^j . Clearly, when $\bar{\tau} = 0$, then the R-ADD-OPT algorithm reduces to the ADD-OPT algorithm in [16].

B. Delayed Digraph Augmentation

To model possibly delayed information exchange between the network nodes, we employ an augmented graph representation (as in [21]) by adding extra virtual nodes that represent the delays on the links. The number of extra virtual nodes representing the delays on the links is bounded by the maximum delay in the network, $\bar{\tau}$. Hence, for each node $v_j \in \mathcal{V}$, we add $\bar{\tau}$ extra virtual nodes, $v_j^{(1)}, v_j^{(2)}, \dots, v_j^{(\bar{\tau})}$, where the virtual node $v_j^{(r)}$ holds the information that is destined to arrive at node v_j after r time steps. Thus, the augmented digraph consists of at most $n(\bar{\tau} + 1)$ nodes and $(1 + 2\bar{\tau})|\mathcal{E}|$ edges. In general, we augment a network of $n = |\mathcal{V}|$ nodes, by introducing $n\bar{\tau}$ nodes which results to a total of $\bar{n} = n(\bar{\tau} + 1)$ nodes. An example of a two-node digraph where the information transmitted over the link ε^{21} is delayed by 2 time steps, and for the link ε^{12} is delayed by 1 time step, is provided in Fig. 1.

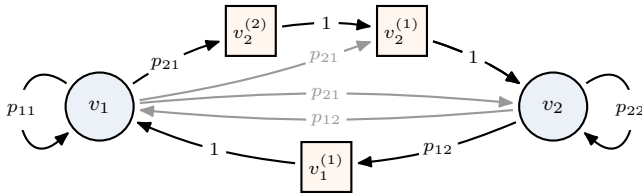


Fig. 1: Example of the augmented digraph with fixed transmission delays.

Based on the graph augmentation, we extend the original vectors of the ADD-OPT algorithm, to keep the information for all nodes and all possible delays, as follows: $\hat{\mathbf{x}}_k =$

$[\mathbf{x}_{k,(0)}, \dots, \mathbf{x}_{k,(\bar{\tau})}]^\top$, $\hat{\mathbf{y}}_k = [\mathbf{y}_{k,(0)}, \dots, \mathbf{y}_{k,(\bar{\tau})}]^\top$, $\hat{\mathbf{z}}_k = [\mathbf{z}_{k,(0)}, \dots, \mathbf{z}_{k,(\bar{\tau})}]^\top$, $\hat{\mathbf{w}}_k = [\mathbf{w}_{k,(0)}, \dots, \mathbf{w}_{k,(\bar{\tau})}]^\top$, and $\nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) = [\nabla \mathbf{f}(\mathbf{z}_{k,(0)}), \mathbf{0}_{n\bar{\tau}}]^\top$, where $\nabla \mathbf{f}(\mathbf{z}_{k,(0)}) = [\nabla f_1(z_{1,(0)}), \dots, \nabla f_n(z_{n,(0)})]^\top$. Note that, the gradients of the virtual nodes $\nabla \mathbf{f}(\mathbf{z}_{k,(r)})$, for $r \neq 0$, are always zero. The vector $\mathbf{x}_{k,(0)}$ contains all scalar states of all actual nodes, while $\mathbf{x}_{k,(r)}$ contains all scalar states of the virtual nodes generated for delay r . Hence, the R-ADD-OPT algorithm can be written in matrix-vector form (for the analysis presented in the subsequent sections) as follows:

$$\hat{\mathbf{x}}_{k+1} = \Xi \hat{\mathbf{x}}_k - \alpha \hat{\mathbf{w}}_k, \quad (6a)$$

$$\hat{\mathbf{y}}_{k+1} = \Xi \hat{\mathbf{y}}_k, \quad (6b)$$

$$\hat{\mathbf{z}}_{k+1} = Y_{k+1}^{-1} \hat{\mathbf{x}}_{k+1}, \quad (6c)$$

$$\hat{\mathbf{w}}_{k+1} = \Xi \hat{\mathbf{w}}_k + \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_{k+1}) - \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k), \quad (6d)$$

where $Y_k = \text{diag}(\hat{\mathbf{y}}_k)$. Additionally, we have the initial conditions, $\hat{\mathbf{x}}_0 = [V^\top \mathbf{0}_{n\bar{\tau}}^\top]^\top$, $\hat{\mathbf{y}}_0 = [\mathbf{1}_n^\top \mathbf{0}_{n\bar{\tau}}^\top]^\top$, $\hat{\mathbf{w}}_0 = \nabla \hat{\mathbf{f}}_0(\hat{\mathbf{z}}_0)$. Matrix $\Xi_k \in \mathbb{R}_+^{\bar{n} \times \bar{n}}$ is a nonnegative matrix associated with the augmented digraph:

$$\Xi \triangleq \begin{pmatrix} P^{(0)} & I_{n \times n} & 0 & \dots & 0 \\ P^{(1)} & 0 & I_{n \times n} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P^{(\bar{\tau}-1)} & 0 & 0 & \dots & I_{n \times n} \\ P^{(\bar{\tau})} & 0 & 0 & \dots & 0 \end{pmatrix} \quad (7)$$

where each element of $P^{(r)}$ is determined by:

$$P^{(r)}(j, i) = \begin{cases} P(j, i), & \text{if } \tau^{ji} = r, (j, i) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Clearly, matrix Ξ maintains column-stochasticity, although the information sent over the links might be delayed.

IV. CONVERGENCE ANALYSIS

To analyze the R-ADD-OPT in the presence of transmission delays, we further define the following notation:

$$\bar{\mathbf{x}}_k = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top \hat{\mathbf{x}}_k \quad (9a) \quad \hat{\mathbf{z}}^* = [\mathbf{1}_n^\top z^* \mathbf{0}_{\bar{n}-n}^\top]^\top \quad (9b)$$

$$\bar{\mathbf{w}}_k = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top \hat{\mathbf{w}}_k \quad (9c) \quad \xi = \|\Xi - I_{\bar{n}}\|_2 \quad (9d)$$

$$\bar{\mathbf{g}}_k^{\hat{\mathbf{z}}} = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) \quad (9e) \quad \epsilon = \|I_{\bar{n}} - \Xi_\infty\|_2 \quad (9f)$$

$$\bar{\mathbf{g}}_k^{\bar{\mathbf{x}}} = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top \nabla \hat{\mathbf{f}}(\bar{\mathbf{x}}_k) \quad (9g) \quad \eta = \max(\chi, \zeta) \quad (9h)$$

where z^* comes from assumption A3, $\zeta = |1 - \bar{n}\alpha L|$, $\chi = |1 - \bar{n}\alpha\mu|$, $\Xi_\infty = \lim_{k \rightarrow \infty} \Xi^k$, and $Y_\infty = \lim_{k \rightarrow \infty} Y_k$. The convergence of Ξ and Y_∞ is shown through the Lemma 1 and Lemma 2.

Lemma 1 (Xi *et. al.* [16]). *Let assumption A1 and A2 hold. Consider $Y_\infty = \lim_{k \rightarrow \infty} Y_k$ and $\Xi_k = \Xi$ being the column-stochastic matrix as defined in (7), with its non- $\mathbf{1}_{\bar{n}}$ Perron vectors denoted by $\boldsymbol{\pi}$. Then, for any vector $\mathbf{a} \in \mathbb{R}^{\bar{n}}$, and for*

³Note, that the ratio $\mathbf{z}_{k,(1)}, \dots, \mathbf{z}_{k,(\bar{\tau})}$ of the virtual nodes could involve a division by zero since their corresponding $\mathbf{y}_{k,(1)}, \dots, \mathbf{y}_{k,(\bar{\tau})}$ initiates at 0. However, the values of the virtual nodes are never used to evaluate gradients but they are introduced for analysis purposes instead.

$$\bar{\mathbf{a}} = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^T \mathbf{a}, \text{ there exists } 0 < \sigma < 1 \text{ such that } \forall k$$

$$\|\Xi \mathbf{a} - Y_\infty \bar{\mathbf{a}}\|_\pi \leq \sigma \|\mathbf{a} - Y_\infty \bar{\mathbf{a}}\|_\pi. \quad (10)$$

Lemma 2 (Nedic et. al. [17]). *Let Assumptions A1 and A2 hold, and consider Y_k and its limit Y_∞ as generated from the weight matrix Ξ . Then, there exist $0 < \gamma_1 < 1$ (the contraction factor defined in Lemma 1 and $0 < \psi < \infty$ such that $\forall k$*

$$\|Y_k - Y_\infty\| \leq \psi \gamma_1^k. \quad (11)$$

Given the notation for the delayed case in (9), we can further denote $\mathbf{t}_k, \mathbf{s}_k \in \mathbb{R}^3$, and $G, H_k \in \mathbb{R}^{3 \times 3}$, for all k as:

$$\mathbf{t}_k = \begin{bmatrix} \|\hat{\mathbf{x}}_k - Y_\infty \bar{\mathbf{x}}_k\| \\ \|\bar{\mathbf{x}}_k - \hat{\mathbf{z}}^*\|_2 \\ \|\hat{\mathbf{w}}_k - Y_\infty \bar{\mathbf{g}}_k\| \end{bmatrix}, \quad \mathbf{s}_k = \begin{bmatrix} \|\hat{\mathbf{x}}_k\|_2 \\ 0 \\ 0 \end{bmatrix},$$

$$G = \begin{bmatrix} \sigma & 0 & \alpha \\ \alpha c L \tilde{y} & \eta & 0 \\ c d \epsilon L \tilde{y} (\xi + \alpha L y \tilde{y}) & \alpha d \epsilon L^2 y \tilde{y} & \sigma + \alpha c d \epsilon L \tilde{y} \end{bmatrix},$$

$$H_k = \begin{bmatrix} 0 & 0 & 0 \\ \alpha L \tilde{y} \psi \gamma_1^{k-1} & 0 & 0 \\ (\alpha L y + 2) d \epsilon L \tilde{y}^2 \psi \gamma_1^{k-1} & 0 & 0 \end{bmatrix}, \quad (12)$$

where σ is given in Lemma 3, c and d are positive constants from the equivalence of $\|\cdot\|$ and $\|\cdot\|_2$, L is the Lipschitz-continuity constant from assumption A3, $y = \sup_k \|Y_k\|_2$, $\tilde{y} = \sup_k \|Y_k^{-1}\|_2$, and ξ, ϵ, η are defined in (9).

Lemma 3 (Xi et. al. [29]). *Let assumption A1 hold and consider an irreducible and primitive nonnegative matrix $W \in \mathbb{R}_+^{n \times n}$ and its non- $\mathbf{1}_n$ Perron vector denoted by \mathbf{v} . Then $\forall \mathbf{a} \in \mathbb{R}^n$ we have:*

$$\|W \mathbf{a} - W_\infty \mathbf{a}\|_{\mathbf{v}} \leq \sigma_W \|\mathbf{a} - W_\infty \mathbf{a}\|_{\mathbf{v}} \quad (13)$$

where $\sigma_W \triangleq \|W - W_\infty\|_{\mathbf{v}} < 1$. One can also verify that

$$\sigma_W = \sigma_2 \left(\text{diag}(\sqrt{\mathbf{v}})^{-1} W \text{diag}(\sqrt{\mathbf{v}}) \right) \quad (14)$$

where $\sigma_2(\cdot)$ is the second largest singular value of a matrix. ■

In order to prove convergence of the R-ADD-OPT algorithm, we first need to show that the evolution of $\|\hat{\mathbf{x}}_k - Y_\infty \bar{\mathbf{x}}_k\|$, $\|\bar{\mathbf{x}}_k - \hat{\mathbf{z}}^*\|_2$, and $\|\hat{\mathbf{w}}_k - Y_\infty \bar{\mathbf{g}}_k\|$ are bounded. In other words, these elements should be bounded with respect to their predecessor state. Given the notation in (12) we introduce Lemma 4, by which we show that these elements are kept bounded (the proof is reported in the Appendix).

Lemma 4. *Consider assumptions A1-A4 for the case of time-invariant delays, and let $\mathbf{t}_k, \mathbf{s}_k, G$, and H_k be defined as in (12), then the following linear relation is satisfied:*

$$\mathbf{t}_k \leq G \mathbf{t}_{k-1} + H_{k-1} \mathbf{s}_{k-1}. \quad (15)$$

Next, we show that given an appropriate selection of the gradient step-size, the R-ADD-OPT algorithm is guaranteed to converge to the optimal solution of the problem in (1), regardless the length of time-invariant delays on the transmission links. It is worth mentioning that, the range of the gradient step-size that can guarantee convergence, can be computed *a priori* based on the maximum fixed transmission delay in the network, given

that it is bounded. The step-size range that guarantees the convergence of the R-ADD-OPT algorithm to the optimal solution, is given by Lemma 5, and its proof is available in the Appendix.

Lemma 5 (Gradient step-size for convergence, Xi et. al. [16]). *Consider the matrix G_α being a function of the step-size α , as defined in (12). It is worth mentioning that every element in G_α , other than the step-size α , is constant. Then, it follows that $\rho(G_\alpha) < 1$ if the step-size $\alpha \in (0, \bar{\alpha})$, where*

$$\bar{\alpha} = \min \left\{ \frac{\sqrt{\delta^2 + 4\bar{n}\mu(1-\sigma)^2\theta} - \delta}{2\theta}, \frac{1}{\bar{n}L} \right\}, \quad (16)$$

and $\delta = \bar{n}\mu c d \epsilon L \tilde{y} (1 - \sigma + \xi)$, $\theta = c d \epsilon L^2 y \tilde{y}^2 (L + \bar{n}\mu)$, while c and d are the constants from the equivalence of $\|\cdot\|$ defined in Lemma 1 and $\|\cdot\|_2$.

Theorem 1. *Consider the R-ADD-OPT algorithm in (5) and let assumptions A1-A4 hold. The R-ADD-OPT algorithm converges exponentially for an appropriate gradient step-size $\alpha \in (0, \bar{\alpha})$, where $\bar{\alpha}$ is given in (16), which depends on the upper bound of the delays in the network.*

Proof. In order to prove the linear convergence of R-ADD-OPT, it is sufficient to show that $\|\mathbf{t}_k\|$ in (15) goes to zero exponentially. To this aim, it is sufficient to show that:

- The spectral radius of matrix G is less than 1, i.e., $\rho(G) < 1$, that is the largest absolute value of the eigenvalues of G is less than 1. Following the results of Lemma 5, we can guarantee that $\rho(G) < 1$ if the step-size is within the range $\alpha \in (0, \bar{\alpha})$, where $\bar{\alpha}$ is given in (16).
- The term H_k decays in linear fashion. This can be shown since $0 < \gamma_1 < 1$, and hence according to [16, Lemma 5], H_k decays with k .

V. NUMERICAL EXAMPLE

Consider a network of 5 agents (nodes), represented by the digraph shown in digraph in Fig. 2. In this example, the main goal of the nodes is to allocate their resources in order to minimize a global cost function of the form of (1), where each node is endowed with a scalar local cost function $f_i(x) : \mathbb{R} \rightarrow \mathbb{R}$. Such scalar local cost functions, within the context of the distributed resource allocation (DRA) problem, are often quadratic of the form:

$$f_i(x) = \frac{1}{2} \beta_i (x - \varphi_i)^2 \quad (17)$$

where $\beta_i > 0, \varphi_i \in \mathbb{R}$ is the demand in node v_i (and in our case is a positive real number), and x is a global optimization parameter that will determine the resource allocation at each node. The optimal allocation for problem (1) with local quadratic cost functions as in (17) can be solved in closed form, and its minimizer x^* is given by

$$x^* = \arg \min_{x \in \mathcal{X}} \sum_{v_i \in \mathcal{V}} f_i(x) = \frac{\sum_{v_i \in \mathcal{V}} \beta_i \varphi_i}{\sum_{v_i \in \mathcal{V}} \beta_i} \quad (18)$$

where \mathcal{X} is the set of feasible values of x . Note that if $\beta_i = 1$ for all $v_i \in \mathcal{V}$, the optimal solution is the average consensus value.

Each node v_j chooses the weights of its out-going links, as defined in (3), and executes the R-ADD-OPT iterations in (5). The nodes initiate their iterations at $\mathbf{x}_0 = [4 \ 1 \ 5 \ 2 \ 3]^\top$, $\mathbf{y}_0 = [1 \ 1 \ 1 \ 1 \ 1]^\top$, and $\beta_0 = [1 \ 5 \ 3 \ 4 \ 1]^\top$. The maximum delay in the network is $\bar{\tau} = \max\{\bar{\tau}^{ji}\}$. In this setting, the range on the step-size that guarantees convergence can be computed by setting $c = d = L = 1$, $y = 1.67$, $\tilde{y} = 3$, $\mu = 0.1$, $\epsilon = 1.1$, and $\xi = 1.13$ to (16).

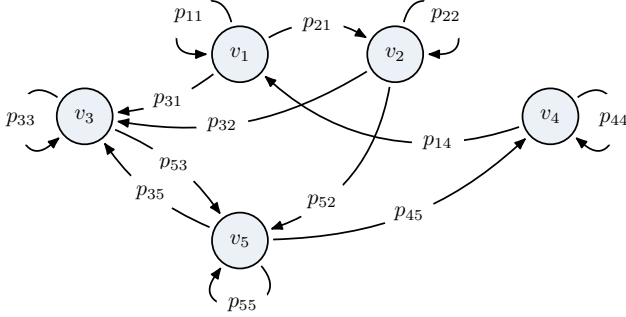


Fig. 2: Digraph consisting of five agents.

To emphasize the influence of delays on the convergence speed of the R-ADD-OPT algorithm, we compute σ as given in Lemma 3 for different upper bound on the delays. As shown in Table II, with longer delays on the links, σ approaches one as the second largest eigenvalue of Ξ approaches one, while for the non-delayed case, the σ parameter is the second largest eigenvalue of P .

TABLE II: Parameter σ for different upper bound on the delays.

$\bar{\tau}$	0	2	5	10
σ	0.599	0.877	0.963	0.987

Fig. 3 depicts the variation of spectral radius of matrix G_a , i.e., $\rho(G_a)$, with respect to step-size, for different lengths of maximum delay in the network. In particular, Fig. 3 indicates the validity of the theoretical bounds on the step-sizes. Additionally, we can observe that we can achieve the best convergence rate by selecting the step-size that minimizes the spectral radius. Intuitively we can infer that for larger delays the gradient step-size should be smaller to guarantee convergence to the optimal value.

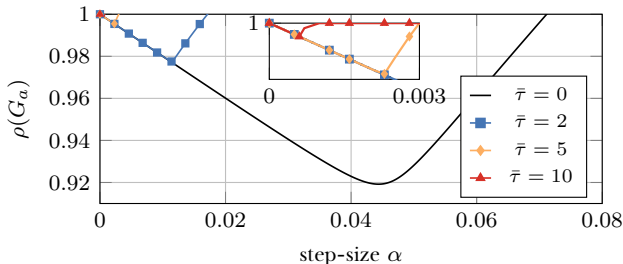


Fig. 3: Spectral radius with alpha step-size value for different delays.

To get intuition about the convergence rate of the R-ADD-OPT algorithm with respect to different lengths of delays, we plot the residual $\frac{1}{n} \sum_{i=1}^n (z_k^i - x^*)^2$, for different upper bounds on the delays. In this example we set the step-size on the value that minimizes the spectral radius as shown in Fig. 3, and we examine both time-invariant and time-varying delays, assuming that all available (and non-self loops) links are delay-prone. In particular, for time-invariant delays, the information is delayed by exactly $\bar{\tau}$ on all delay-prone links, while for time-varying delays, the delay length is random and within $0 \leq \tau_k^{ji} \leq \bar{\tau}^*$. As shown in Fig. 4, the convergence rate of the R-ADD-OPT algorithm to the optimal solution depends on the length of the delays, and the step-size should be set at lower values for longer delays on links such that the convergence is guaranteed (see Fig. 3). Despite showing a single ensemble for the time-varying delays in Fig. 4, we ran simulations for several different realizations for random integer variable τ_k^{ji} , satisfying $0 \leq \tau_k^{ji} \leq \bar{\tau}^*$. It is conjectured that the R-ADD-OPT algorithm works also for bounded time-varying delays, as long as the step-size is within a range specified by the delay upper bound $\bar{\tau}^*$. However, the proof for this range, while it is possibly a conservative bound, remains an open problem.

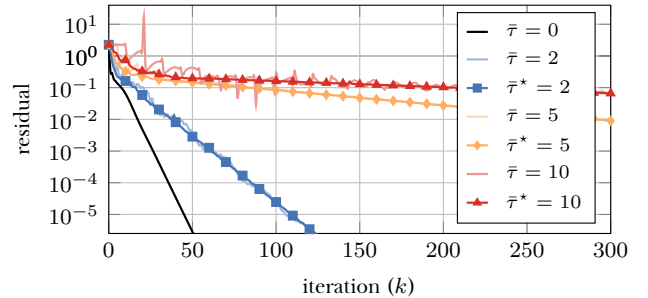


Fig. 4: Residual for different upper bound on the delays in the network $\bar{\tau}$. Time-varying delays are within the bound $0 \leq \tau_k^{ji} \leq \bar{\tau}^*$.

VI. CONCLUSIONS

In this paper, we proposed a modified version of the ADD-OPT algorithm, called R-ADD-OPT, that handles heterogeneous transmission delays over unidirectional networks. We showed analytically that if the gradient step-size of the R-ADD-OPT algorithm is within a certain predefined range, the agents are guaranteed to converge to the optimal solution of the distributed optimization problem, regardless the length of delays on the transmission links. The range of the step-size that can guarantee convergence, can be computed *a priori* based on the maximum transmission delay in the network, given that it is bounded. Finally, we showed with experimental simulations the potential of the R-ADD-OPT algorithm to converge to the global optimal value in the presence of time-varying delays on the communication links.

Proof of Lemma 1. First, from Perron-Frobenius Theorem we know that $\rho(\Xi) = 1$ since Ξ is column-stochastic and irreducible. Every eigenvalue of Ξ other than the largest in magnitude, *i.e.*, $|\lambda_1|$, is strictly less than $\rho(\Xi)$. Moreover, since Ξ is primitive, we know that $\boldsymbol{\pi}$ is a strictly positive (right) eigenvector corresponding to the eigenvalue of 1 such that $\mathbf{1}_{\bar{n}}^\top \boldsymbol{\pi} = 1$, and hence $\Xi_\infty = \lim_{k \rightarrow \infty} \Xi^k = \boldsymbol{\pi} \mathbf{1}_{\bar{n}}^\top$. One can also verify that $\Xi \Xi_\infty = \Xi_\infty \Xi_\infty = \Xi_\infty$. Then it follows that:

$$\begin{aligned} \Xi \mathbf{a} - Y_\infty \bar{\mathbf{a}} &= \Xi \mathbf{a} - \Xi_\infty \mathbf{a} \\ &= \Xi \mathbf{a} - \Xi_\infty \mathbf{a} + \Xi_\infty \Xi_\infty Y_\infty \bar{\mathbf{a}} - \Xi \Xi_\infty Y_\infty \bar{\mathbf{a}} \\ &= (\Xi - \Xi_\infty)(\mathbf{a} - Y_\infty \bar{\mathbf{a}}). \end{aligned}$$

Therefore, considering the right eigenvector $\boldsymbol{\pi}$ of Ξ , we get:

$$\begin{aligned} \|\Xi \mathbf{a} - Y_\infty \bar{\mathbf{a}}\|_{\boldsymbol{\pi}} &= \|(\Xi - \Xi_\infty)(\mathbf{a} - Y_\infty \bar{\mathbf{a}})\|_{\boldsymbol{\pi}} \\ &\stackrel{(i)}{\leq} \|\Xi - \Xi_\infty\|_{\boldsymbol{\pi}} \|\mathbf{a} - Y_\infty \bar{\mathbf{a}}\|_{\boldsymbol{\pi}} \\ &\stackrel{(ii)}{\leq} \sigma \|\mathbf{a} - Y_\infty \bar{\mathbf{a}}\|_{\boldsymbol{\pi}}, \end{aligned}$$

where (i) stems from the sub-multiplicative matrix norm property and (ii) stems from Lemma 3. \blacksquare

Proof of Lemma 4. The proof follows the results of [16] *mutatis mutandis* by considering the augmented matrix-vector form of the R-ADD-OPT algorithm in (6), which we include here for completeness. Let assumptions A1-A4 hold. For the first element $\|\hat{\mathbf{x}}_k - Y_\infty \hat{\mathbf{x}}_k\|$, we first substitute $\hat{\mathbf{x}}_k$ from (6a), and $\bar{\mathbf{x}}_k$ from Lemma 6, and we get:

$$\begin{aligned} \|\hat{\mathbf{x}}_k - Y_\infty \bar{\mathbf{x}}_k\| &= \|\Xi \hat{\mathbf{x}}_{k-1} - \alpha \hat{\mathbf{w}}_{k-1} - Y_\infty (\bar{\mathbf{x}}_{k-1} - \alpha \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}})\| \\ &\leq \|\Xi \hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1}\| \\ &\quad + \alpha \|\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}\| \\ &\leq \sigma \|\hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1}\| \\ &\quad + \alpha \|\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}\|, \end{aligned} \quad (19)$$

where the last inequality holds by Lemma 1.

For the second element, $\|\bar{\mathbf{x}}_k - \hat{\mathbf{z}}^*\|_2$, we first substitute $\bar{\mathbf{x}}_k$ from Lemma 6, and we get:

$$\begin{aligned} \|\bar{\mathbf{x}}_k - \hat{\mathbf{z}}^*\| &= \|\bar{\mathbf{x}}_{k-1} - \alpha \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}} - \hat{\mathbf{z}}^*\|_2 \\ &= \|\mathbf{x}_+ - \hat{\mathbf{z}}^* - \alpha \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}} + \alpha \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}\|_2 \\ &\leq \eta \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 + \alpha \|\hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}} - \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}\|_2 \\ &\leq \eta \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 + \alpha L \|\hat{\mathbf{z}}_{k-1} - \bar{\mathbf{x}}_{k-1}\|_2 \\ &\leq \eta \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 + \alpha L \|Y_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \bar{\mathbf{x}}_{k-1}\|_2 \\ &\leq \eta \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 \\ &\quad + \alpha L \|Y_{k-1}^{-1} (\hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1})\|_2 \\ &\quad + \alpha L \|(Y_{k-1}^{-1} Y_\infty - I_{\bar{n}}) \bar{\mathbf{x}}_{k-1}\|_2 \\ &\quad + \alpha L \tilde{y} \psi \gamma_1^{k-1} \|\bar{\mathbf{x}}_{k-1}\|_2, \end{aligned} \quad (20)$$

where the second equality holds by letting $\mathbf{x}_+ = \bar{\mathbf{x}}_{k-1} - \alpha \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}$, the first inequality holds by applying Lemma 7, and the second inequality by making use of the Lipschitz-continuity assumption from A3. Then, the third inequality

is obtained by substituting $\hat{\mathbf{z}}_{k-1}$ from (6c), and finally, we get the last inequality by applying [16, Lemma 8].

For the third element, $\|\hat{\mathbf{w}}_k - Y_\infty \hat{\mathbf{g}}_k^\hat{\mathbf{z}}\|$, we substitute (6d), and we get:

$$\begin{aligned} \|\hat{\mathbf{w}}_k - Y_\infty \hat{\mathbf{g}}_k^\hat{\mathbf{z}}\| &= \|\Xi \hat{\mathbf{w}}_{k-1} + \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) - \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_{k-1}) - Y_\infty \hat{\mathbf{g}}_k^\hat{\mathbf{z}}\| \\ &\leq \|\Xi \hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}\| \\ &\quad + \left\| \left(\nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) - \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_{k-1}) \right) - (Y_\infty \hat{\mathbf{g}}_k^\hat{\mathbf{z}} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}) \right\| \\ &\leq \sigma \|\hat{\mathbf{w}}_{k-1} - Y_\infty \bar{\mathbf{w}}_{k-1}\| \\ &\quad + \left\| \left(\nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) - \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_{k-1}) \right) - (Y_\infty \hat{\mathbf{g}}_k^\hat{\mathbf{z}} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}) \right\| \\ &\leq \sigma \|\hat{\mathbf{w}}_{k-1} - Y_\infty \bar{\mathbf{w}}_{k-1}\| \\ &\quad + \left\| (I_{\bar{n}} - \Xi_\infty) \left(\nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_k) - \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_{k-1}) \right) \right\|_2 \\ &\leq \sigma \|\hat{\mathbf{w}}_{k-1} - Y_\infty \bar{\mathbf{w}}_{k-1}\| + d\epsilon L \|\hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1}\|_2 \end{aligned} \quad (21)$$

where the second inequality comes from Lemma 6 and Lemma 1. Then, since $\frac{1}{\bar{n}} Y_\infty \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top = \Xi_\infty$, we get the third inequality, while by using the Lipschitz-continuity assumption A3, we obtain the last inequality. Now we need to bound $\|\hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1}\|_2$:

$$\begin{aligned} \|\hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1}\|_2 &\leq \|Y_k^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1})\|_2 \\ &\quad + \|(Y_k^{-1} - Y_{k-1}^{-1}) \hat{\mathbf{x}}_{k-1}\|_2 \\ &\leq \|Y_k^{-1} (\Xi - I_{\bar{n}}) \hat{\mathbf{x}}_{k-1}\|_2 + \alpha \|Y_k^{-1} \hat{\mathbf{w}}_{k-1}\|_2 \\ &\quad + \|Y_k^{-1} - Y_{k-1}^{-1}\|_2 \|\hat{\mathbf{x}}_{k-1}\|_2 \\ &\leq (\tilde{y}\xi + \alpha \tilde{y}^2 y L) \|\hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1}\|_2 \\ &\quad + \alpha \tilde{y} \|\mathbf{w}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}\|_2 \\ &\quad + \alpha \tilde{y} y L \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 \\ &\quad + (\alpha y L + 2) \tilde{y}^2 \psi \gamma_1^{k-1} \|\hat{\mathbf{x}}_{k-1}\|_2, \end{aligned} \quad (22)$$

where the first inequality is obtained by substituting (6c), second inequality by substituting (6a), third inequality holds due to $(\Xi - I_{\bar{n}}) Y_\infty \hat{\mathbf{x}}_{k-1} = \mathbf{0}_{\bar{n}}$, and:

$$\|\hat{\mathbf{g}}_{k-1}^\hat{\mathbf{x}}\| = \left\| \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top \nabla \hat{\mathbf{f}}(\bar{\mathbf{x}}_{k-1}) \right\|_2 \leq L \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|, \quad (23)$$

which bounds the following:

$$\begin{aligned} \|Y_k^{-1} \hat{\mathbf{w}}_{k-1}\| &\leq \|Y_k^{-1} (\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}})\|_2 \\ &\quad + \|Y_k^{-1} Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}\|_2 \\ &\quad + \|Y_k^{-1} Y_\infty (\hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}} - \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}})\|_2 \\ &\leq \tilde{y} \|\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}\|_2 \\ &\quad + \tilde{y} y L \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 \\ &\quad + \tilde{y} y L \|\hat{\mathbf{z}}_{k-1} - \bar{\mathbf{x}}_{k-1}\|_2 \\ &\leq \tilde{y} \|\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^\hat{\mathbf{z}}\|_2 \\ &\quad + \tilde{y} y L \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 \\ &\quad + \tilde{y}^2 y L \|\hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1}\|_2 \\ &\quad + \tilde{y}^2 y L \psi \gamma_1^{k-1} \|\hat{\mathbf{x}}_{k-1}\|_2, \end{aligned} \quad (24)$$

where the last inequality holds due to the bound of $\|\hat{\mathbf{z}}_{k-1} - \bar{\mathbf{x}}_{k-1}\|$ obtained from the last inequality of (20). Finally, by substituting (22) in (21) we get the bound on

the third element, given by:

$$\begin{aligned} \|\hat{\mathbf{w}}_k - Y_\infty \hat{\mathbf{g}}_k^z\| &\leq (cdeL\xi\tilde{y} + \kappa) \|\hat{\mathbf{x}}_{k-1} - Y_\infty \bar{\mathbf{x}}_{k-1}\| \\ &\quad + \alpha d\epsilon L^2 y \tilde{y} \|\bar{\mathbf{x}}_{k-1} - \hat{\mathbf{z}}^*\|_2 \\ &\quad + (\sigma + \alpha cdeL\tilde{y}) \|\hat{\mathbf{w}}_{k-1} - Y_\infty \hat{\mathbf{g}}_{k-1}^z\| \\ &\quad + (\alpha yL + 2)d\epsilon L\tilde{y}^2 \psi \gamma_1^{k-1} \|\hat{\mathbf{x}}_{k-1}\|_2, \end{aligned} \quad (25)$$

where $\kappa = \alpha cdeL^2 y \tilde{y}^2$.

Proof completed with all elements bounded. \blacksquare

Proof of Lemma 5. To derive the range of the gradient step-size α in (16) that maintains the spectral radius $\rho(G_a) < 1$, we adopt similar analysis as in [16] *mutatis mutandis*, by replacing n , with the total number of nodes in the augmented digraph \bar{n} . \blacksquare

Lemma 6 (Auxiliary relation). Consider the variables $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{w}}_k$ from (9) that incorporate the transmission delays due to the exchange of information between the nodes. The following equations hold for all k :

$$(a) \bar{\mathbf{w}}_k = \hat{\mathbf{g}}_k^z, \text{ and } (b) \bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k - \alpha \hat{\mathbf{g}}_k^z.$$

Proof. From the column-stochasticity property of Ξ , and (6d) we have that $\bar{\mathbf{w}}_k = \bar{\mathbf{w}}_{k-1} + \hat{\mathbf{g}}_k^z - \nabla \hat{\mathbf{g}}_{k-1}^z$. Writing $\bar{\mathbf{w}}_k$ recursively we get $\bar{\mathbf{w}}_k = \bar{\mathbf{w}}_0 + \hat{\mathbf{g}}_k^z - \hat{\mathbf{g}}_0^z$, which gives $\bar{\mathbf{w}}_k = \hat{\mathbf{g}}_k^z$ due to the initial conditions in (6d), and $\bar{\mathbf{w}}_0 = \nabla \hat{\mathbf{f}}(\hat{\mathbf{z}}_0)$. From (6a) we have that $\bar{\mathbf{x}}_{k+1} = \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}} \mathbf{1}_{\bar{n}}^\top (\Xi \bar{\mathbf{x}}_k - \alpha \hat{\mathbf{w}}_k) = \bar{\mathbf{x}}_k - \alpha \hat{\mathbf{g}}_k^z$. \blacksquare

Lemma 7 (Bubeck [30]). Let Assumption A3 hold for the objective functions $f_i(\mathbf{x})$ in (1). For any $\mathbf{x} \in \mathbb{R}^p$ define $\mathbf{x}_+ = \mathbf{x} - \alpha \nabla \mathbf{f}(\mathbf{x})$, where $0 < \alpha < \frac{2}{\bar{n}L}$. Then

$$\|\mathbf{x}_+ - \underline{\mathbf{x}}^*\| \leq \eta \|\mathbf{x} - \underline{\mathbf{x}}^*\| \quad (26)$$

where $\eta = \max(|1 - \alpha \bar{n}L|, |1 - \alpha \bar{n}\mu|)$.

REFERENCES

- [1] Zhu Wang, Dong Wang, Changyun Wen, Fanghong Guo, and Wei Wang. Push-Based Distributed Economic Dispatch in Smart Grids Over Time-Varying Unbalanced Directed Graphs. *IEEE Trans. Smart Grid*, 12(4):3185–3199, 2021.
- [2] Fanghong Guo, Guoqi Li, Changyun Wen, Lei Wang, and Ziyang Meng. An Accelerated Distributed Gradient-Based Algorithm for Constrained Optimization With Application to Economic Dispatch in a Large-Scale Power System. *IEEE Trans. Sys., Man, and Cybernetics: Systems*, 51(4):2041–2053, 2021.
- [3] Andreas Grammenos, Themistoklis Charalambous, and Evangelia Kalyvianaki. CPU Scheduling in Data Centers Using Asynchronous Finite-Time Distributed Coordination Mechanisms. *IEEE Trans. Netw. Sci. Eng.*, 10(4):1880–1894, 2023.
- [4] Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Consensus-based Distributed Optimization: Practical Issues and Applications in Large-scale Machine Learning. In *Annu. Allerton Conf. Commun. Control Comput.*, pages 1543–1550, 2012.
- [5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Machine Learning*, 3(1):1–122, 2011.
- [6] Usman A Khan, Soumya Kar, and José MF Moura. DILAND: An Algorithm for Distributed Sensor Localization with Noisy Distance Measurements. *IEEE Trans. Signal Process.*, 58(3):1940–1947, 2009.
- [7] Sam Safavi, Usman A Khan, Soumya Kar, and José MF Moura. Distributed Localization: A Linear Theory. *Proc. IEEE*, 106(7):1204–1223, 2018.
- [8] Guannan Qu and Na Li. Harnessing Smoothness to Accelerate Distributed Optimization. *IEEE Trans. Control Net. Systems*, 5(3):1245–1260, 2017.
- [9] Kun Yuan, Qing Ling, and Wotao Yin. On the Convergence of Decentralized Gradient Descent. *SIAM J. Optim.*, 26(3):1835–1854, 2016.
- [10] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An Exact First-order Algorithm for Decentralized Consensus Optimization. *SIAM J. Optim.*, 25(2):944–966, 2015.
- [11] Alejandro D. Dominguez-Garcia and Christoforos N. Hadjicostis. Distributed Matrix Scaling and Application to Average Consensus in Directed Graphs. *IEEE Trans. Autom. Control*, 58(3):667–681, 2013.
- [12] K. Cai and H. Ishii. Average Consensus on General Strongly Connected Digraphs. *Automatica*, 48(11):2750–2761, 2012.
- [13] Alejandro D Dominguez-Garcia and Christoforos N Hadjicostis. Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services. In *IEEE Int. Conf. Smart Grid Commun.*, pages 537–542, 2010.
- [14] Christoforos N Hadjicostis, Alejandro D Domínguez-García, Themistoklis Charalambous, et al. Distributed Averaging and Balancing in Network Systems: with Applications to Coordination and Control. *Found. Trends Syst. Control*, 5(2-3):99–292, 2018.
- [15] Chenguang Xi and Usman A Khan. DEXTRA: A Fast Algorithm for Optimization over Directed Graphs. *IEEE Trans. Autom. Control*, 62(10):4980–4993, 2017.
- [16] Chenguang Xi, Ran Xin, and Usman A Khan. ADD-OPT: Accelerated Distributed Directed Optimization. *IEEE Trans. Autom. Control*, 63(5):1329–1339, 2017.
- [17] Angelia Nedić and Alex Olshevsky. Distributed Optimization over Time-varying Directed Graphs. *IEEE Trans. Autom. Control*, 60(3):601–615, 2014.
- [18] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving Geometric Convergence for Distributed Optimization over Time-varying Graphs. *SIAM J. Optim.*, 27(4):2597–2633, 2017.
- [19] Chenguang Xi and Usman A Khan. Distributed Subgradient Projection Algorithm over Directed Graphs. *IEEE Trans. Autom. Control*, 62(8):3986–3992, 2016.
- [20] Chenguang Xi, Qiong Wu, and Usman A Khan. On the Distributed Optimization over Directed Networks. *Neurocomputing*, 267:508–515, 2017.
- [21] Christoforos N Hadjicostis and Themistoklis Charalambous. Average Consensus in the Presence of Delays in Directed Graph Topologies. *IEEE Trans. Autom. Control*, 59(3):763–768, 2013.
- [22] Evagoras Makridis, Themistoklis Charalambous, and Christoforos N. Hadjicostis. Utilizing Feedback Channel Mechanisms for Reaching Average Consensus over Directed Network Topologies. In *American Control Conf.*, pages 1781–1787, 2023.
- [23] Konstantinos I Tsianos and Michael G Rabbat. Distributed Consensus and Optimization under Communication Delays. In *Annu. Allerton Conf. Commun. Control Comput.*, pages 974–982, 2011.
- [24] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented Distributed Gradient Methods for Multi-Agent Optimization under Uncoordinated Constant Stepsizes. In *IEEE Conf. Decision Control*, pages 2055–2060, 2015.
- [25] Zhuoqing Song, Lei Shi, Shi Pu, and Ming Yan. Optimal Gradient Tracking for Decentralized Optimization. *Mathematical Programming*, pages 1–53, 2023.
- [26] Ivano Notarnicola, Michelangelo Bin, Lorenzo Marconi, and Giuseppe Notarstefano. The Gradient Tracking Is a Distributed Integral Action. *IEEE Trans. Autom. Control*, pages 1–8, 2023.
- [27] Alejandro D Domínguez-García and Christoforos N Hadjicostis. Distributed Strategies for Average Consensus in Directed Graphs. In *IEEE Conf. Decision Control*, pages 2124–2129, 2011.
- [28] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based Computation of Aggregate Information. In *IEEE Symp. Found. Comput. Sci.*, pages 482–491, 2003.
- [29] Ran Xin, Anit Kumar Sahu, Usman A Khan, and Soumya Kar. Distributed Stochastic Optimization with Gradient Tracking over Strongly-Connected Networks. In *IEEE Conf. Decision Control*, pages 8353–8358, 2019.
- [30] Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.