Delft University of Technology

Delft Center for Systems and Control

Technical report 23-010

Distributed adaptive resource allocation: An uncertain saddle-point dynamics viewpoint*

D. Yue, S. Baldi, J. Cao, Q. Li, and B. De Schutter

If you want to cite this report, please use the following reference instead:
D. Yue, S. Baldi, J. Cao, Q. Li, and B. De Schutter, "Distributed adaptive resource allocation: An uncertain saddle-point dynamics viewpoint," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 12, pp. 2209–2221, Dec. 2023. doi:10.1109/JAS.2023.123402

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/23_010.html

Distributed Adaptive Resource Allocation: an Uncertain Saddle-point Dynamics Viewpoint

Dongdong Yue, *Member, IEEE/CAA,* Simone Baldi, *Senior Member, IEEE,*
Jinde Cao, *Fellow, IEEE,* Qi Li, and Bart De Schutter, *Fellow, IEEE*

***Abstract*—This paper addresses distributed adaptive optimal resource allocation problems over weight-balanced digraphs. By leveraging state-of-the-art adaptive coupling designs for multiagent systems, two adaptive algorithms are proposed, namely a directed-spanning-tree-based algorithm and a node-based algorithm. The benefits of these algorithms are that they require neither sufficiently small or unitary step sizes, nor global knowledge of Laplacian eigenvalues, which are widely required in the literature. It is shown that both algorithms belong to a class of uncertain saddle-point dynamics, which can be tackled by repeatedly adopting the Peter-Paul inequality in the framework of Lyapunov theory. Thanks to this new viewpoint, global asymptotic convergence of both algorithms can be proven in a unified way. The effectiveness of the proposed algorithms is validated through numerical simulations and case studies in IEEE 30- and 118-bus power systems.**

***Index Terms*—Resource allocation, directed graphs, saddle-point dynamics, adaptive systems**

I. INTRODUCTION

The resource allocation problem, also known as the economic dispatch problem, has recently aroused multi-disciplinary interest. Applications of resource allocation include various engineering fields such as cloud computing, sensor networks, and power systems. While early works studied optimal resource allocation based on a central node collecting and processing all data from every node in the network [1], this architecture is not effective in large-scale networks. Therefore, distributed resource allocation algorithms are highly desirable, i.e., to solve an allocation problem by making each node collect and process the data from only a few neighboring nodes, according to the topology of the network.

This work was supported in part by the China Postdoctoral Science Foundation under Grant BX2021064, in part by the Fundamental Research Funds for the Central Universities under Grant 2242022R20030, in part by the Key Intergovernmental Special Fund of National Key Research and Development Program under Grant 2021YFE0198700, in part by the Research Fund for International Scientists under Grant 62150610499, and in part by the Natural Science Foundation of China under Grant 62073074 and Grant 61833005. *(Corresponding authors: Simone Baldi and Jinde Cao).*

D. Yue is with the School of Mathematics, Southeast University, Nanjing, China (e-mail: yued@seu.edu.cn).

S. Baldi is with the School of Mathematics, Southeast University, Nanjing, China, and with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands (e-mail: S.Baldi@tudelft.nl).

J. Cao is with the School of Mathematics, Southeast University, Nanjing, China, and with the Yonsei Frontier Lab, Yonsei University, Seoul, South Korea (e-mail: jdcao@seu.edu.cn).

Qi Li is with the School of Automation, Southeast University, Nanjing, China (e-mail: liqikj@hotmail.com).

B. De Schutter is with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands (e-mail: B.DeSchutter@tudelft.nl).

Different assumptions can be made on the graph describing the large-scale network: acyclic (tree) graph [2], undirected connected graph [3]–[12], strongly connected weight-balanced digraph [13]–[18], or weight-unbalanced digraph [19]–[21]. In most of these works, the algorithms used to solve the distributed resource allocation problem require unitary step sizes, or sufficiently small step sizes to implement local gradient descent, see e.g. [4]–[6], [17]–[20]. Meanwhile, many algorithms rely on homogeneous and static coupling gains, selected based on the global knowledge of Laplacian eigenvalues, e.g., [7], [10], [14]–[17], [21]. Such a strategy may lead to high-gain instability when the network is large and sparse (with a Laplacian eigenvalue being extremely close to the imaginary axis). Besides, for an effective *distributed* methodology, eliminating the global knowledge of the Laplacian matrix is crucial, which goes under the name of distributed adaptive implementation.

In fact, distributed adaptive algorithms incorporate adaptive (in place of static) coupling gains, which have the superiority of adapting to different network configurations. The reason is that these adaptive gains do not need to be selected based on global knowledge of Laplacian eigenvalues. Distributed adaptive designs with adaptive coupling gains are available in the literature for consensus or tracking [22]–[26], containment or formation [27]–[29], and optimization [30], [31].

Distributed resource allocation solutions with adaptive coupling gains, to our best knowledge, are not available in the literature, even for the simplest case of undirected graphs. The main reason for this gap lies in the following difficulty: in order to obtain an optimal resource allocation solution, the agents are supposed to seek a consensus over the Lagrangian multipliers based on a class of nested primal-dual dynamics [4]. This strategy brings the challenge of individual seeking of optimal allocation decisions and consensus seeking of the Lagrangian multipliers at the same time, without any knowledge of Laplacian eigenvalues. A possible approach to address this challenge is to solve the consensus optimization problem for the Lagrangian multipliers via distributed adaptive optimization of [30], [31]. Such an approach of focusing on the dual problem instead of the primal problem was indeed adopted in [11], [19], [20], but it may bring the so-called “two-time-scale” problem, as each agent needs to solve an auxiliary optimization problem at each time instant towards optimal resource allocation [4]. The “two-time-scale” issue also exists in other approaches, see e.g. the alternating direction method of multiplies [32].

Motivated by the above discussions, this work studies distributed adaptive solutions to the resource allocation problem.

We provide a novel perspective into this problem by showing that the optimal solution corresponds to the (generalized) equilibrium of a class of uncertain saddle-point dynamics. The basic idea to guarantee convergence to this equilibrium is to introduce heterogeneous adaptive coupling gains promoting consensus over the Lagrangian multipliers of optimal decisions, and to let the agents self-determine the coupling strengths between each other. To implement this idea, two distributed adaptive strategies are studied, i.e., directed-spanning-tree-based (DST-based) and node-based: in the former, only the gains associated with edges along an DST are made adaptive; in the latter, the gains associated with all incoming edges for each node (so that all edges in the network) are made adaptive. The main contributions of this paper are as follows:

1) We propose a new point of view into the resource allocation problems, which is made possible by framing the problem via a novel class of uncertain saddle-point dynamics. We show that the optimal solution to the resource allocation problem corresponds to a generalized equilibrium point of the uncertain saddle-point dynamics, as discussed in Definition 1 and Lemma 6.
2) Inspired by the uncertain saddle-point dynamics viewpoint, we propose two novel distributed adaptive frameworks for solving optimal resource allocation over digraphs and prove their convergence in a unified way (Theorems 1-2).
3) Two novel classes of convexity conditions named spanning-tree-based strongly convexity and jointly strongly convexity are identified for the proposed algorithms, respectively. We also show a relatively standard class of local cost functions that automatically satisfies the proposed convexity conditions (Corollaries 1-2).
4) The proposed algorithms require neither sufficiently small or unitary step sizes, nor global knowledge of Laplacian eigenvalues, which are widely required in nonadaptive strategies proposed in the literature, see e.g. [4]–[7], [14]–[21]. Besides, the proposed algorithms focus on the primal resource allocation problem directly: thus, the "two-time-scale" issue in the duality-based literature [11], [19], [20] does not arise.

The rest of the paper is organized as follows. In Section II, we give the preliminaries and problem statement, and we introduce uncertain saddle-point dynamics for the problem. In Section III and Section IV, two distributed adaptive resource allocation algorithms are established as DST-based and node-based, respectively. In Section V, simulations are performed to validate the theoretical results. Some discussions are presented in Section VI. Finally, Section VII concludes the paper and discusses some future topics.

II. Preliminaries and Problem Formulation

A. Matrix Algebra

A series of technical lemmas useful for stability analysis is now introduced. The so-called Peter-Paul inequality will be frequently used throughout this paper to bound non-definite terms with positive definite expressions.

Notations:	
$\mathbb{R}$ (resp. $\mathbb{R}^+$)	set of real (resp. positive) scalars;
$\mathbb{R}^n$	set of n-dimensional column vectors;
$\mathbb{R}^n_+$	set of n-dimensional positive (all n entries being positive) column vectors;
$\mathbb{R}^{n\times m}$	set of $n\times m$ matrices;
$\mathbf{I}_n$	$n\times n$ identity matrix;
$\mathbf{1}_n$	column vector with all n elements being one;
0	zero scalar, zero vectors, and zero matrices;
A^s	symmetric part of square matrix A, i.e., $(A+A^T)/2$;
$\bar{\lambda}(A)$ (resp. $\underline{\lambda}(A)$)	maximum (resp. minimum) eigenvalue of real symmetric matrix A;
$\mathcal{M}^n_{\rm r}$	set of $n\times n$ matrices with zero row sums;
$A\succ 0$ (resp. $A\succeq 0$)	A is positive definite (resp. semi-definite);
$\mathcal{I}_N$	set of natural numbers $\{1,2,\cdots,N\}$;
$\mathcal{S}_1\setminus\mathcal{S}_2$	set difference of sets $\mathcal{S}_1$ and $\mathcal{S}_2$;
$\mathrm{col}(x_1,\cdots,x_N)$	column vectorization $({x_1}^T,\cdots,{x_N}^T)^T$;
$\mathrm{diag}(\cdot)$	diagonalization operator;
$\otimes$	Kronecker product;
$\mathcal{O}\vert_{x_0}$	velocity of autonomous dynamical system $\mathcal{O}:\dot{x}=f(x)$ at x_0, i.e., $f(x_0)$;
$\nabla_x f$	partial derivative of f with respect to x;
∇f	gradient of f.

Lemma 1 (Peter-Paul inequality, [33]): For any $a,b\in\mathbb{R}^n$ and $\epsilon\in\mathbb{R}^+$, there holds

$$a^Tb\le\frac{a^Ta}{2\epsilon}+\frac{\epsilon b^Tb}{2}.$$

Proof: The lemma follows directly from the Young inequality with exponents 2 and a positive bias ϵ. ■

The following lemma can be inferred from [22, Lemma 2.3], and will be used (cf. (39)) to analyze the node-based algorithm of Section IV.

Lemma 2: Suppose that $U\in\mathbb{R}^{N\times N}$. Let $S\in\mathbb{R}^{n\times n}$ be an orthogonal matrix and $x=\mathrm{col}(x_1,\cdots,x_N)$ be an aggregated vector with $x_i\in\mathbb{R}^n$, $i\in\mathcal{I}_N$. Then,

$$x^T(U\otimes\mathbf{I}_n)x=\sum_{k=1}^n y_k^TUy_k$$

where $y_k=\big([Sx_1]_k,[Sx_2]_k,\cdots,[Sx_N]_k\big)^T$, $k\in\mathcal{I}_n$. Here $[Sx_i]_k$ is the k-th entry of the vector Sx_i.

B. Algebraic Graph Theory

A weighted *digraph* [34] (short for *directed graph*) $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ consists of the node set $\mathcal{V}=\mathcal{I}_N$, the edge set $\mathcal{E}=\{e_{ij}|i\to j, i\neq j\}$, and the weighted adjacency matrix $\mathcal{W}=(w_{ij})\in\mathbb{R}^{N\times N}$ where $w_{ij}>0$ if $e_{ji}\in\mathcal{E}$, and $w_{ij}=0$ otherwise. The node i is an in-neighbor of j ($i\in\mathcal{N}_{\rm in}(j)$) if $e_{ij}\in\mathcal{E}$, and in return, j is an out-neighbor of i ($j\in\mathcal{N}_{\rm out}(i)$). The Laplacian matrix $\mathcal{L}=(\mathcal{L}_{ij})\in\mathbb{R}^{N\times N}$ of $\mathcal{G}$ is defined as follows: $\mathcal{L}_{ij}=-w_{ij}, i\neq j$, and $\mathcal{L}_{ii}=\sum_{k=1,k\neq i}^N w_{ik}, i=1,\cdots,N$. A path is a series of edges connecting a pair of nodes. A digraph $\mathcal{G}$ is *strongly connected* if there exists a directed path between any pair of nodes. Moreover, $\mathcal{G}$ is *weight-balanced* if, for any $i\in\mathcal{V}$, there holds $\sum_{j\in\mathcal{N}_{\rm in}(i)}w_{ij}=\sum_{j\in\mathcal{N}_{\rm out}(i)}w_{ji}$.

A *DST* (short for *directed spanning tree*) $\bar{\mathcal{G}}(\mathcal{V},\bar{\mathcal{E}},\bar{\mathcal{W}})$ of $\mathcal{G}$ is a subgraph that involves a node called the root, which has no in-neighbors, such that there exists one and only one directed path from the root to any other node. Without loss of generality, we label the root as node 1, and use p_k to represent the unique parent (in-neighbor) of node $k+1$ in $\bar{\mathcal{G}}$, $k \in \mathcal{I}_{N-1}$. Clearly, $\bar{\mathcal{E}} = \{e_{\mathrm{p}_k,k+1} | k \in \mathcal{I}_{N-1}\} \subseteq \mathcal{E}$. Correspondingly, $\bar{\mathcal{L}}$ (resp. $\bar{\mathcal{W}}$) is the Laplacian (resp. weighted adjacency) matrix of $\bar{\mathcal{G}}$ and $\bar{\mathcal{N}}_{\text{out}}(i)$ is the set of out-neighbors of i in $\bar{\mathcal{G}}$.

The graph theory notation allows us to introduce two lemmas useful for stability analysis. Lemma 3 will be used to analyze the DST-based algorithm of Section III. Lemma 4 will be used to analyze the node-based algorithm of Section IV.

Lemma 3 ([28], [34]): Consider a digraph $\mathcal{G}$ that contains a DST $\bar{\mathcal{G}}$. Then, the following statements hold:

1) The Laplacian $\mathcal{L}$ has a simple zero eigenvalue corresponding to the right eigenvector $\mathbf{1}_N$, and the other eigenvalues have positive real parts.
2) Define a matrix $\Xi \in \mathbb{R}^{(N-1)\times N}$ as
$$\Xi_{kj} = \begin{cases} -1, & \text{if } j = k+1, \\ 1, & \text{if } j = \mathrm{p}_k, \\ 0, & \text{otherwise.} \end{cases}$$
Then, there exists a unique $Q \in \mathbb{R}^{(N-1)\times(N-1)}$ such that $\Xi\mathcal{L} = Q\Xi$.
3) The eigenvalues of Q are exactly the nonzero eigenvalues of $\mathcal{L}$, thus $Q^TQ \succ 0$ and $Q^{\mathrm{s}} \succ 0$. As a consequence, $\underline{\lambda}(Q^{\mathrm{s}}) = \lambda_2(\mathcal{L}^{\mathrm{s}})$, where $\lambda_2(\mathcal{L}^{\mathrm{s}})$ is the smallest nonzero eigenvalue of $\mathcal{L}^{\mathrm{s}}$.
4) The matrix Q is explicitly given by $Q = \tilde{Q} + \bar{Q}$ with
$$Q_{kj} = \underbrace{\sum_{c\in\bar{\mathcal{V}}_{j+1}} (\tilde{\mathcal{L}}_{k+1,c} - \tilde{\mathcal{L}}_{\mathrm{p}_k,c})}_{\tilde{Q}_{kj}} + \underbrace{\sum_{c\in\bar{\mathcal{V}}_{j+1}} (\bar{\mathcal{L}}_{k+1,c} - \bar{\mathcal{L}}_{\mathrm{p}_k,c})}_{\bar{Q}_{kj}},$$
where $\tilde{\mathcal{L}} = \mathcal{L} - \bar{\mathcal{L}}$. Here, $\bar{\mathcal{V}}_{j+1}$ is the node set of the subtree of $\bar{\mathcal{G}}$ rooting at $j+1$. Furthermore, the matrix $\bar{Q}$ is related to $\bar{\mathcal{L}}$ through
$$\bar{Q}_{kj} = \begin{cases} \bar{\mathcal{L}}_{j+1,j+1}, & \text{if } j = k, \\ -\bar{\mathcal{L}}_{j+1,j+1}, & \text{if } j = \mathrm{p}_k - 1, \\ 0, & \text{otherwise.} \end{cases}$$

Remark 1: The existence of the matrix Q is guaranteed by Lemma 9 of [35], and the uniqueness of Q is guaranteed by the fact that Ξ has full row rank.

Lemma 4 ([25], [36]): Suppose $\mathcal{G}$ is strongly connected. Then, the following statements hold:

1) There exists a positive left eigenvector $r = (r_1, r_2 \cdots, r_N)^T \in \mathbb{R}_+^N$ of $\mathcal{L}$ associated with the zero eigenvalue. Let $R = \mathrm{diag}(r_1, \cdots, r_N)$. Then, $\hat{\mathcal{L}} \triangleq R\mathcal{L} + \mathcal{L}^TR \succeq 0$ is the symmetric Laplacian matrix associated with an undirected graph. Moreover, $r = r_0\mathbf{1}_N$ for some $r_0 \in \mathbb{R}^+$ if and only if $\mathcal{G}$ is weight-balanced.
2) For any $\varsigma \in \mathbb{R}_+^N$ and $x \in \mathbb{R}^N$, there holds
$$\min_{\varsigma^T x=0, x\neq 0} \frac{x^T\hat{\mathcal{L}}x}{x^Tx} > \frac{\lambda_2(\hat{\mathcal{L}})}{N}.$$
3) [Courant-Fischer] In the special case that ς is chosen as $\mathbf{1}_N$, i.e., the eigenvector of $\hat{\mathcal{L}}$ associated with the zero eigenvalue, then
$$\min_{\mathbf{1}_N^T x=0, x\neq 0} \frac{x^T\hat{\mathcal{L}}x}{x^Tx} = \lambda_2(\hat{\mathcal{L}}).$$

C. Problem Statement

Consider N agents interacting over a digraph $\mathcal{G}$. Each agent has an amount of local resources $d_i \in \mathbb{R}^n$ and is associated to a *local* cost function $f_i(\cdot) : \mathbb{R}^n \to \mathbb{R}$. In distributed resource allocation, the agents are cooperatively seeking a global allocation strategy with minimum cumulative cost $f(\cdot) : \mathbb{R}^{Nn} \to \mathbb{R}$ (referred to as the *global* cost function), while meeting the sum of the total resources:

$$\min_{x\triangleq\mathrm{col}(x_1,\cdots,x_N)} f(x) \triangleq \sum_{i=1}^N f_i(x_i), \qquad (1)$$
$$\text{subject to } \sum_{i=1}^N x_i = d,$$

where $d = \sum_{i=1}^N d_i$.

The following assumption is standard in the distributed resource allocation literature, see e.g. [4], [6], [11], [17].

Assumption 1: Each local cost function $f_i(\cdot)$ is continuously differentiable and strictly convex.

Lemma 5 (Solution of (1)): Under Assumption 1, problem (1) has a unique solution x^*. Moreover, there exists a unique $y^* \in \mathbb{R}^n$, i.e., the Lagrangian multiplier, such that

$$\begin{aligned} &\nabla f(x^*) + \mathbf{1}_N \otimes y^* = 0; \\ &(\mathbf{1}_N^T \otimes \mathbf{I}_n)(x^* - D) = 0. \end{aligned} \qquad (2)$$

where $\nabla f(x) = \mathrm{col}(\nabla f_1(x_1), \cdots, \nabla f_N(x_N))$ according to the definition of $f(\cdot)$ and $D = \mathrm{col}(d_1, \cdots, d_N)$.

Remark 2: Equation (2) is known in the literature as the Karush-Kuhn-Tucker (KKT) condition (see e.g., [37, Chap. 5]). Specifically, given the Lagrangian function of problem (1), i.e., $L(x,y) = f(x) + y^T(\mathbf{1}_N^T \otimes \mathbf{I}_n)(x-D)$, the KKT condition (2) consists of $\nabla_x L(x,y) = 0$ (tangency) and $\nabla_y L(x,y) = 0$ (feasibility).

In this paper, the following assumption is made regarding the communication graph.

Assumption 2: The communication digraph $\mathcal{G}$ is strongly connected and weight-balanced.

Remark 3: This assumption is standard in distributed resource allocation as well as distributed optimization problems [14]–[18], and is considerably more general than the assumption of $\mathcal{G}$ being undirected and connected [4]–[12]. Note that there have been some results on weight-unbalanced digraphs [19]–[21], which require sufficiently small step sizes for gradient descent and can raise the “two-time-scale” issue [19], [20], or rely on constant coupling gain selected according to the Laplacian eigenvalues [21]. These limitations are not desired for an effective *distributed* methodology. Note that, if $\mathcal{G}$ is weight-unbalanced, one can recover Assumption 2 by first performing a finite-time weight-balancing algorithm along a DST, cf. [31].

D. Primary Analysis

To solve problem (1), one can in principle use saddle-point dynamics, i.e., a gradient descent of the Lagrangian function

$L(x,y)$ in the primal variable x and a gradient ascent in the dual variable $y_0 \in \mathbb{R}^n$:

$$\begin{aligned}\dot{x} &= -\nabla f(x) - \mathbf{1}_N \otimes y_0;\\ \dot{y}_0 &= (\mathbf{1}_N^T \otimes \mathbf{I}_n)(x - D).\end{aligned} \tag{3}$$

However, one problem of (3) is that the update of y_0 cannot be performed in a distributed way. To make the saddle-point algorithm (3) distributed, several algorithms have been proposed, such as endowing each agent a copy of the dual variable as $y_i \in \mathbb{R}^n$, $i \in \mathcal{V}$, while incorporating an integral feedback action of y_i, see [4], [17].

Therefore, let us consider the system resulting from incorporating a distributed integral feedback action of local dual variables on top of (3), as follows:

$$\begin{aligned}\mathcal{O}: \quad &\dot{x} = -\kappa_1(\nabla f(x) + y) &\text{(4a)}\\ &\dot{y} = x - D - (\Upsilon \otimes \mathbf{I}_n)y - (\mathcal{L} \otimes \mathbf{I}_n)z &\text{(4b)}\\ &\dot{z} = (\Upsilon \otimes \mathbf{I}_n)y &\text{(4c)}\end{aligned}$$

where $\kappa_1 \in \mathbb{R}^+$, and $y = \mathrm{col}(y_1, \cdots, y_N) \in \mathbb{R}^{Nn}$, $z = \mathrm{col}(z_1, \cdots, z_N) \in \mathbb{R}^{Nn}$ contain the local auxiliary variables y_i and z_i for agent i. The matrix $\mathcal{L}$ in (4b) is the Laplacian of $\mathcal{G}$. We will refer to system (4) as *uncertain* since the matrix $\Upsilon \in \mathcal{M}_{\mathrm{r}}^N$ is unknown a priori. More specifically, the matrix Υ represents the coupling between y_i and y_j, which is the result of an adaptation mechanism to be designed so as to guarantee stable attractive behavior of (4).

Remark 4: State-of-the-art distributed algorithms to solve problem (1) directly involve the Laplacian matrix $\mathcal{L}$ in place of Υ (see e.g., [4], [17]). However, a unitary step size of the gradient descent is required and, in the case of [17], the global knowledge of Laplacian eigenvalues is also required.

Let us define the generalized equilibrium points (GEP) of the uncertain system (4) as follows:

Definition 1 (GEP): The triple $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}^{Nn} \times \mathbb{R}^{Nn} \times \mathbb{R}^{Nn}$ is called a generalized equilibrium point of (4), if for any $\Upsilon \in \mathcal{M}_{\mathrm{r}}^N$, there holds $\mathcal{O}|_{(\tilde{x},\tilde{y},\tilde{z})} = 0$.

Lemma 6 (GEPs of (4)): Under Assumptions 1-2, the uncertain system (4) has infinitely many GEPs. Moreover, if $(\tilde{x}, \tilde{y}, \tilde{z})$ is a GEP of (4), then $(\tilde{x}, \tilde{y}) = (x^*, \mathbf{1}_N \otimes y^*)$, i.e., $\tilde{x}$ is the optimizer of problem (1). The latter statement implies that $(\tilde{x}, \tilde{y})$ is unique.

Proof: Since $(\Upsilon \otimes \mathbf{I}_n)\tilde{y} = 0$ for any $\Upsilon \in \mathcal{M}_{\mathrm{r}}^N$, we have $\tilde{y} = \mathbf{1}_N \otimes y_0$ for some $y_0 \in \mathbb{R}^n$. Substituting $(\tilde{x}, \tilde{y}, \tilde{z})$ into (4b) and left-multiplying $(\mathbf{1}_N^T \otimes \mathbf{I}_n)$ to both sides lead to $(\mathbf{1}_N^T \otimes \mathbf{I}_n)(\tilde{x} - D) - (\mathbf{1}_N^T \mathcal{L} \otimes \mathbf{I}_n)\tilde{z} = 0$. Under Assumption 2, we have $\mathbf{1}_N^T \mathcal{L} = 0$, implying that $(\mathbf{1}_N^T \otimes \mathbf{I}_n)(\tilde{x} - D) = 0$, which together with $\nabla f(\tilde{x}) + \mathbf{1}_N \otimes y_0 = 0$, results exactly in the KKT condition (2). By Lemma 5, we know that $(\tilde{x}, \tilde{y}) = (x^*, \mathbf{1}_N \otimes y^*)$ exists and is unique. Furthermore, since $\mathrm{rank}(\mathcal{L}) = N-1$, there exist infinitely many solutions $\tilde{z}$ such that $(\mathcal{L} \otimes \mathbf{I}_m)\tilde{z} = x - D$: in fact, if $(\tilde{x}, \tilde{y}, \tilde{z})$ is a GEP of (4), also $(\tilde{x}, \tilde{y}, \tilde{z} + \mathbf{1}_N \otimes \Delta z)$ is a GEP of (4) for any $\Delta z \in \mathbb{R}^n$. ∎

Lemma 6 states that distributed optimal resource allocation can be realized by steering the uncertain saddle-point dynamics (4) to its GEPs. In the following two sections, we will propose two continuous realizations of Υ in (4b), that are DST-based and node-based, respectively, and guarantee stable attractive behavior of the GEPs of (4).

III. Distributed Adaptive Resource Allocation: DST-based Design

Recall that, with the strongly connected property, a DST can be identified in a distributed fashion without any prior knowledge of the Laplacian matrix [38]. Based on any DST $\bar{\mathcal{G}}$ of $\mathcal{G}$, consider the distributed adaptive resource allocation (DARA) algorithm for agent $i \in \mathcal{V}$, $j \neq i$ ($k \in \mathcal{I}_{N-1}$), as follows:

$$\begin{aligned}&\mathcal{O}^a:\\ &\dot{x} = -\kappa_1(\nabla f(x) + y) &\text{(5a)}\\ &\dot{y} = x - D - (\mathcal{L}^a \otimes \mathbf{I}_n)y - (\mathcal{L} \otimes \mathbf{I}_n)z &\text{(5b)}\\ &\dot{z} = (\mathcal{L}^a \otimes \mathbf{I}_n)y &\text{(5c)}\end{aligned}$$

$$\dot{a}_{ij} = \begin{cases} \kappa_2\Big((y_j - y_i) - \sum\limits_{c \in \bar{\mathcal{N}}_{\mathrm{out}}(i)} (y_i - y_c)\Big)^T (y_j - y_i) \triangleq \dot{\bar{a}}_{k+1,\mathrm{p}_k}, & \text{if } e_{ji} \in \bar{\mathcal{E}} \\ 0, & \text{if } e_{ji} \in \mathcal{E} \setminus \bar{\mathcal{E}} \end{cases} \tag{5d}$$

where $\kappa_2 \in \mathbb{R}^+$ and $\mathcal{L}^a$ is the gain-dependent Laplacian matrix defined as follows:

$$\begin{aligned}\mathcal{L}_{ij}^a &= -a_{ij} w_{ij}, i \neq j;\\ \mathcal{L}_{ii}^a &= \sum_{j=1, j\neq i}^{N} a_{ij} w_{ij}, i = 1, \cdots, N.\end{aligned} \tag{6}$$

The weight w_{ij} multiplied by the gain a_{ij} determines the feedback gain of the relative error vector $(y_i - y_j)$ for agent i to update y_i and z_i. Note that we did not define a_{ii} in (5)-(6) since there are no self-loops. According to (5d), the gain a_{ij} is updated only when $e_{ji} \in \bar{\mathcal{E}}$. Such an update law is distributed, i.e. it depends on agent i, agent j and all the out-neighbors of agent i in the DST [24], [31]. One can refer to Algorithm 1 for the implementation of (5).

Theorem 1: Under Assumptions 1-2, the adaptive algorithm (5) drives (x, y) to $(x^*, \mathbf{1}_N \otimes y^*)$ asymptotically for any initial condition $(x(0), y(0), z(0) \in \mathbb{R}^{Nn} \times \mathbb{R}^{Nn} \times \mathbb{R}^{Nn}$ and any $a_{ij}(0) \in \mathbb{R}$ provided there exists a scalar $m \in \mathbb{R}^+$, such that the following condition (referred to as *spanning-tree-based m-strongly convexity*) holds $\forall x, y \in \mathbb{R}^{Nn}$:

$$\begin{aligned}(x - y)^T(\bar{\mathcal{L}}^{\mathrm{U}} \otimes \mathbf{I}_n)(\nabla f(x) - \nabla f(y)) \geq\\ m(x - y)^T(\bar{\mathcal{L}}^{\mathrm{U}} \otimes \mathbf{I}_n)(x - y)\end{aligned} \tag{7}$$

where $\bar{\mathcal{L}}^{\mathrm{U}} = \Xi^T \Xi$ is the un-weighted Laplacian matrix of the undirected spanning tree $\bar{\mathcal{G}}^{\mathrm{U}}$ based on $\bar{\mathcal{G}}$ (Ξ is defined as in Lemma 3). Moreover, the adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}$, $k \in \mathcal{I}_{N-1}$, converge to some finite constant values.

Proof: We conduct the proof by showing that each trajectory of (5a)-(5c) converges to a GEP of (4). Let us define the error vectors between the trajectory of (5a)-(5c) and any GEP $(\tilde{x}, \tilde{y}, \tilde{z})$ of (4), following a change of coordinates:

$$\mu = x - \tilde{x}, \quad \nu = y - \tilde{y}, \quad \eta = z - \tilde{z} \tag{8a}$$

$$\bar{\mu} = (\Xi \otimes \mathbf{I}_n)\mu, \quad \bar{\nu} = (\Xi \otimes \mathbf{I}_n)\nu, \quad \bar{\eta} = (\Xi \otimes \mathbf{I}_n)\eta \tag{8b}$$

Algorithm 1: DARA: DST-based

Data: (1) initialization: $x_i(0)$, $y_i(0)$, $z_i(0)$, $a_{ij}(0)$; (2) parameters: κ_1, κ_2; (3) structure: a DST $\bar{\mathcal{G}}(\mathcal{V}, \bar{\mathcal{E}})$

Result: Optimal resource allocation solution $x_i \to x^*$

$s \leftarrow 1$;
while $s \cdot h \leq T_{tml}$ **do** // h is the integration step and T_{tml} is the terminal time
  for $i \leftarrow 1$ **to** N **do**
    $\mathrm{d}x_i \leftarrow -\kappa_1(\nabla f_i(x_i) + y_i)$;
    $\mathrm{d}y_i \leftarrow x_i - d_i - \sum_{j\in\mathcal{V}} \mathcal{L}^a_{ij} y_j - \sum_{j\in\mathcal{V}} \mathcal{L}_{ij} z_j$;
    $\mathrm{d}z_i \leftarrow \sum_{j\in\mathcal{V}} \mathcal{L}^a_{ij} y_j$;
    for $j \leftarrow 1$ **to** N *and* $j \neq i$ **do**
      $\mathrm{d}a_{ij} \leftarrow 0$;
      if $e_{ji} \in \bar{\mathcal{E}}$ **then** // $\exists k \in \mathcal{I}_{N-1}$ such that $i = k+1$ and $j = \mathrm{p}_k$
        $\mathrm{d}a_{ij} \leftarrow \kappa_2\Big((y_j - y_i) - \sum_{c\in\bar{\mathcal{N}}_{\text{out}}(i)} (y_i - y_c)\Big)^T (y_j - y_i)$;
      end
      $a_{ij} \leftarrow a_{ij} + h \cdot \mathrm{d}a_{ij}$;
    end
    $x_i \leftarrow x_i + h \cdot \mathrm{d}x_i$;
    $y_i \leftarrow x_i + h \cdot \mathrm{d}y_i$;
    $z_i \leftarrow x_i + h \cdot \mathrm{d}z_i$;
  end
  $s \leftarrow s + 1$;
end

where Ξ is defined as in Lemma 3. In a component-wise form, $\bar{\mu} = \mathrm{col}(\bar{\mu}_1, \cdots, \bar{\mu}_{N-1})$ where $\bar{\mu}_k = \mu_{\mathrm{p}_k} - \mu_{k+1}$, $k \in \mathcal{I}_{N-1}$.

Note that $\mathcal{L}^a \in \mathcal{M}^N_{\mathrm{r}}$. By Definition 1, we have $\mathcal{O}^a|_{(\tilde{x},\tilde{y},\tilde{z})} = 0$. Then, in the new coordinates (8b), the dynamics of $\mathcal{O}^a$ is equivalent to

$$\dot{\bar{\mu}} = -\kappa_1(\Xi \otimes \mathbf{I}_n)h - \kappa_1\bar{\nu} \tag{9a}$$
$$\dot{\bar{\nu}} = \bar{\mu} - (Q^a \otimes \mathbf{I}_n)\bar{\nu} - (Q \otimes \mathbf{I}_n)\bar{\eta} \tag{9b}$$
$$\dot{\bar{\eta}} = (Q^a \otimes \mathbf{I}_n)\bar{\nu} \tag{9c}$$
$$\dot{\bar{a}}_{k+1,\mathrm{p}_k} = \kappa_2\Big(\bar{\nu}_k - \sum_{j\in\bar{\mathcal{N}}_{\text{out}}(k+1)} \bar{\nu}_{j-1}\Big)^T \bar{\nu}_k,\ k \in \mathcal{I}_{N-1} \tag{9d}$$

where $h = \nabla f(\mu + \tilde{x}) - \nabla f(\tilde{x})$ in (9a), and Q (resp. Q^a), is defined as in Lemma 3 based on the DST $\bar{\mathcal{G}}$ and the (resp. gain-dependent) Laplacian matrix. More specifically, $Q^a = \tilde{Q}^a + \bar{Q}^a$ contains the fixed matrix $\tilde{Q}^a$ (note that $\dot{a}_{ij} = 0$ if $e_{ji} \in \mathcal{E} \setminus \bar{\mathcal{E}}$), and the time-varying matrix

$$\bar{Q}^a_{kj} = \begin{cases} \bar{a}_{j+1,\mathrm{p}_j} w_{j+1,\mathrm{p}_j}, & \text{if } j = k, \\ -\bar{a}_{j+1,\mathrm{p}_j} w_{j+1,\mathrm{p}_j}, & \text{if } j = \mathrm{p}_k - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Here, statement 2) of Lemma 3 and the properties of the Kronecker product have been used to get (9b)-(9c); and the fact that $(\Xi \otimes \mathbf{I}_n)\tilde{y} = 0$ has been used to get (9d).

Consider the following candidate Lyapunov function:

$$V_1 = \frac{1 + 3\bar{\lambda}(Q^TQ)}{\epsilon_1 \underline{\lambda}^2(Q^{\mathrm{s}})} V_{\bar{\mu}} + V^a_{\bar{\nu}} + \frac{3\bar{\lambda}(Q^TQ)}{\underline{\lambda}(Q^{\mathrm{s}})} V_{\bar{\eta}} \tag{11}$$

where

$$\begin{aligned} V_{\bar{\mu}} &= \frac{1}{2}\bar{\mu}^T\bar{\mu} \\ V^a_{\bar{\nu}} &= \frac{1}{2}\bar{\nu}^T\bar{\nu} + \sum_{k=1}^{N-1} \frac{w_{k+1,\mathrm{p}_k}}{2\kappa_2}\big(\bar{a}_{k+1,\mathrm{p}_k}(t) - \phi_{k+1,\mathrm{p}_k}\big)^2 \\ V_{\bar{\eta}} &= \frac{1}{2}(\bar{\nu} + \bar{\eta})^T(\bar{\nu} + \bar{\eta}) \end{aligned} \tag{12}$$

and $Q^{\mathrm{s}} > 0$ is guaranteed by 3) of Lemma 3, and $\epsilon_1, \phi_{k+1,\mathrm{p}_k} \in \mathbb{R}^+$, $k = 1, \cdots, N-1$, will be determined later.

The time derivative of $V_{\bar{\mu}}$ can be obtained as

$$\dot{V}_{\bar{\mu}} = -\kappa_1\bar{\mu}^T(\Xi \otimes \mathbf{I}_n)h - \kappa_1\bar{\mu}^T\bar{\nu}. \tag{13}$$

By (8b) and (7), we have

$$\bar{\mu}^T(\Xi \otimes \mathbf{I}_n)h \geq m\bar{\mu}^T\bar{\mu}. \tag{14}$$

Then,

$$\begin{aligned} \dot{V}_{\bar{\mu}} &\leq -\kappa_1 m\bar{\mu}^T\bar{\mu} - \kappa_1\bar{\mu}^T\bar{\nu} \\ &\leq (\epsilon_2 - \kappa_1 m)\bar{\mu}^T\bar{\mu} + \frac{\kappa_1^2}{4\epsilon_2}\bar{\nu}^T\bar{\nu} \end{aligned} \tag{15}$$

where $\epsilon_2 \in \mathbb{R}^+$ is to be decided later, and Lemma 1 was used to get the second inequality.

The time derivative of $V^a_{\bar{\nu}}$ can be obtained as

$$\begin{aligned} \dot{V}^a_{\bar{\nu}} =& \bar{\nu}^T\bar{\mu} - \bar{\nu}^T(Q^a \otimes \mathbf{I}_n)\bar{\nu} - \bar{\nu}^T(Q \otimes \mathbf{I}_n)\bar{\eta} \\ &+ \sum_{k=1}^{N-1} w_{k+1,\mathrm{p}_k}(\bar{a}_{k+1,\mathrm{p}_k} - \phi_{k+1,\mathrm{p}_k})\Big(\bar{\nu}_k \\ &- \sum_{j+1\in\bar{\mathcal{N}}_{\text{out}}(k+1)} \bar{\nu}_j\Big)^T \bar{\nu}_k. \end{aligned} \tag{16}$$

From (10), one has

$$\begin{aligned} &\sum_{k=1}^{N-1} w_{k+1,\mathrm{p}_k}\bar{a}_{k+1,\mathrm{p}_k}\Big(\bar{\nu}_k - \sum_{j+1\in\bar{\mathcal{N}}_{\text{out}}(k+1)} \bar{\nu}_j\Big)^T \bar{\nu}_k \\ =& \sum_{k=1}^{N-1} (\bar{Q}^a_{kk}\bar{\nu}_k + \sum_{j=1, j\neq k}^{N-1} \bar{Q}^a_{jk}\bar{\nu}_j)^T\bar{\nu}_k \\ =& \sum_{k=1}^{N-1}\sum_{j=1}^{N-1} \bar{Q}^a_{jk}\bar{\nu}_j^T\bar{\nu}_k = \bar{\nu}^T(\bar{Q}^a \otimes \mathbf{I}_n)\bar{\nu}. \end{aligned} \tag{17}$$

Following the procedure in [24], [28], [31], let us define $\Phi \in \mathbb{R}^{(N-1)\times(N-1)}$ as

$$\Phi_{kj} = \begin{cases} \phi_{j+1,\mathrm{p}_j} w_{j+1,\mathrm{p}_j}, & \text{if } j = k, \\ -\phi_{j+1,\mathrm{p}_j} w_{j+1,\mathrm{p}_j}, & \text{if } j = \mathrm{p}_k - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Then, it follows from (16)-(18) that

$$\begin{aligned} \dot{V}^a_{\bar{\nu}} =& \bar{\nu}^T\bar{\mu} - \bar{\nu}^T(Q^a \otimes \mathbf{I}_n)\bar{\nu} - \bar{\nu}^T(Q \otimes \mathbf{I}_n)\bar{\eta} \\ &+ \bar{\nu}^T\big((\bar{Q}^a - \Phi) \otimes \mathbf{I}_n\big)\bar{\nu} \\ =& \bar{\nu}^T\bar{\mu} - \bar{\nu}^T\big((\tilde{Q}^a + \Phi) \otimes \mathbf{I}_n\big)\bar{\nu} - \bar{\nu}^T(Q \otimes \mathbf{I}_n)\bar{\eta}. \end{aligned} \tag{19}$$

Note that the time-varying matrix $\bar{Q}^a$ has been canceled, and all the matrices left are constant. Based on Lemma 1, we have

$$\begin{aligned}\dot{V}_{\bar{\nu}}^a \leq & \bar{\nu}^T\bar{\mu} - \bar{\nu}^T\big((\tilde{Q}^a+\Phi)\otimes \mathbf{I}_n\big)\bar{\nu} \\ & + \frac{\bar{\nu}^T\bar{\nu}}{2} + \frac{\bar{\eta}^T(Q^TQ\otimes \mathbf{I}_n)\bar{\eta}}{2} \\ \leq & \frac{1}{\underline{\lambda}^2(Q^{\mathrm{s}})}\bar{\mu}^T\bar{\mu} + \Big(\frac{\underline{\lambda}^2(Q^{\mathrm{s}})}{4}+\frac{1}{2}\Big)\bar{\nu}^T\bar{\nu} \\ & - \bar{\nu}^T\big((\tilde{Q}^a+\Phi)\otimes \mathbf{I}_n\big)\bar{\nu} + \frac{\bar{\lambda}(Q^TQ)}{2}\bar{\eta}^T\bar{\eta}. \end{aligned} \tag{20}$$

where we have also used the property that $x^TAx \leq \bar{\lambda}(A)x^Tx$ for a matrix $A \succ 0$ and for all x to get the last inequality.

The time derivative of $V_{\bar{\eta}}$ can be obtained as

$$\begin{aligned}\dot{V}_{\bar{\eta}} = & \bar{\nu}^T\bar{\mu} - \bar{\nu}^T(Q\otimes \mathbf{I}_n)\bar{\eta} + \bar{\eta}^T\bar{\mu} - \bar{\eta}^T(Q\otimes \mathbf{I}_n)\bar{\eta} \\ \leq & \frac{1}{2\underline{\lambda}(Q^{\mathrm{s}})}\bar{\mu}^T\bar{\mu} + \frac{\underline{\lambda}(Q^{\mathrm{s}})}{2}\bar{\nu}^T\bar{\nu} \\ & + \frac{\bar{\lambda}(Q^TQ)}{\underline{\lambda}(Q^{\mathrm{s}})}\bar{\nu}^T\bar{\nu} + \frac{\underline{\lambda}(Q^{\mathrm{s}})}{4}\bar{\eta}^T\bar{\eta} \\ & + \frac{\underline{\lambda}(Q^{\mathrm{s}})}{2}\bar{\eta}^T\bar{\eta} + \frac{1}{2\underline{\lambda}(Q^{\mathrm{s}})}\bar{\mu}^T\bar{\mu} - \underline{\lambda}(Q^{\mathrm{s}})\bar{\eta}^T\bar{\eta} \\ \leq & \frac{1}{\underline{\lambda}(Q^{\mathrm{s}})}\bar{\mu}^T\bar{\mu} + \Big(\frac{\underline{\lambda}(Q^{\mathrm{s}})}{2}+\frac{\bar{\lambda}(Q^TQ)}{\underline{\lambda}(Q^{\mathrm{s}})}\Big)\bar{\nu}^T\bar{\nu} \\ & - \frac{\underline{\lambda}(Q^{\mathrm{s}})}{4}\bar{\eta}^T\bar{\eta} \end{aligned} \tag{21}$$

where we have repeatedly used Lemma 1 to get the inequality.

Based on (11), (15), (20), and (21) and with some manipulations, the time derivative of V_1 along the trajectory of (9) is upper bounded by

$$\begin{aligned}\dot{V}_1 \leq & -\frac{\big(1+3\bar{\lambda}(Q^TQ)\big)(\kappa_1 m - \epsilon_1 - \epsilon_2)}{\epsilon_1\underline{\lambda}^2(Q^{\mathrm{s}})}\bar{\mu}^T\bar{\mu} \\ & - \bar{\nu}^T\big((\Phi^{\mathrm{s}} - \gamma\mathbf{I}_{N-1} + (\tilde{Q}^a)^{\mathrm{s}})\otimes \mathbf{I}_n\big)\bar{\nu} - \frac{\bar{\lambda}(Q^TQ)}{4}\bar{\eta}^T\bar{\eta} \end{aligned} \tag{22}$$

where $\gamma \in \mathbb{R}^+$ is given by

$$\begin{aligned}\gamma = & \frac{\kappa_1^2\big(1+3\bar{\lambda}(Q^TQ)\big)}{4\epsilon_1\epsilon_2\underline{\lambda}^2(Q^{\mathrm{s}})} + \frac{3\bar{\lambda}^2(Q^TQ)}{\underline{\lambda}^2(Q^{\mathrm{s}})} \\ & + \frac{3\bar{\lambda}(Q^TQ)}{2} + \frac{\underline{\lambda}^2(Q^{\mathrm{s}})}{4} + \frac{1}{2}. \end{aligned} \tag{23}$$

The next procedure is to find some appropriate $\epsilon_1, \epsilon_2, \phi_{k+1,\mathrm{p}_k} \in \mathbb{R}^+$ to stabilize the system. First, we can always select ϵ_1, ϵ_2 such that $\epsilon_1+\epsilon_2 \leq \kappa_1 m$. Next, we select ϕ_{k+1,p_k}, $k = 1, \cdots, N-1$, such that $\Phi^{\mathrm{s}} - \gamma\mathbf{I}_{N-1} + (\tilde{Q}^a)^{\mathrm{s}} \succ 0$. Since γ and $(\tilde{Q}^a)^{\mathrm{s}}$ are both fixed, it is sufficient to prove that by choosing appropriate ϕ_{k+1,p_k}, $\Phi^{\mathrm{s}} - \bar{\gamma}\mathbf{I}_{N-1} \succ 0$ for any $\bar{\gamma} \in \mathbb{R}^+$. This latter statement is indeed guaranteed following similar mathematical induction procedures as in [28]. Specifically, let

$$\phi_{2,\mathrm{p}_1} > \frac{\bar{\gamma}}{w_{2,\mathrm{p}_1}}, \quad \phi_{k+1,\mathrm{p}_k} > \bar{\gamma} + \frac{\sum_{j=2}^k \phi_{j,\mathrm{p}_{j-1}}^2 w_{j,\mathrm{p}_{j-1}}^2}{4w_{k+1,\mathrm{p}_k}\underline{\lambda}(\Omega_{k-1})}, \tag{24}$$

where $\Omega_1 = \big(\ \phi_{2,\mathrm{p}_1}w_{2,\mathrm{p}_1} - \bar{\gamma}\ \big)$, and

$$\Omega_k = \begin{pmatrix} \Omega_{k-1} & \varphi_k \\ \varphi_k^T & \phi_{k+1,\mathrm{p}_k}w_{k+1,\mathrm{p}_k} - \bar{\gamma} \end{pmatrix} \tag{25}$$

with $\varphi_k = \frac{1}{2}(\phi_{k1}w_{k1}, \phi_{k2}w_{k2}, \cdots, \phi_{k,k-1}w_{k,k-1})^T$, $k = 2, \cdots, N-1$. Then, the positive definiteness of $\Phi^{\mathrm{s}} - \bar{\gamma}\mathbf{I}_{N-1}$ (Ω_{N-1}) is guaranteed by the Schur complement [39] and the induction principle. Then, it follows that $\dot{V}_1 \leq 0$, implying that V_1 has a finite limit and all the signals $\bar{\mu}$, $\bar{\nu}$, $\bar{\eta}$, and $\bar{a}_{k+1,\mathrm{p}_k}$ are bounded. Note that since V_1 is continuously differentiable, it is guaranteed by LaSalle's invariance principle that each trajectory of (9) converges to the set such that $\dot{V}_1 = 0$, which by (22), implies that $(\bar{\mu}, \bar{\nu}, \bar{\eta}) \to (0,0,0)$, and the adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}$, $k \in \mathcal{I}_{N-1}$, converge to some finite constant values. Back to the original coordinates of (5), $(x,y,z) \to (\tilde{x} + \mathbf{1}_N\otimes\Delta_x, \tilde{y}+\mathbf{1}_N\otimes\Delta_y, \tilde{z}+\mathbf{1}_N\otimes\Delta_z) \triangleq (x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}})$, for some $\Delta_x, \Delta_y, \Delta_z \in \mathbb{R}^n$

Next, we show that $\Delta_x = \Delta_y = 0$. The steady-state dynamics of Δ_x and Δ_y are governed by

$$\dot{\Delta}_x = \frac{1}{N}(\mathbf{1}_N^T\otimes\mathbf{I}_n)\dot{x}_{\mathrm{s}}, \quad \dot{\Delta}_y = \frac{1}{N}(\mathbf{1}_N^T\otimes\mathbf{I}_n)\dot{y}_{\mathrm{s}}. \tag{26}$$

Substitute (5a)-(5b) evaluated at $(x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}})$ into the above, and note that $\mathcal{O}^a|_{(\tilde{x},\tilde{y},\tilde{z})} = 0$. Then we obtain

$$\begin{aligned}&\dot{\Delta}_x = -\frac{\kappa_1}{N}(\mathbf{1}_N^T\otimes\mathbf{I}_n)\big(\nabla f(x_{\mathrm{s}}) - \nabla f(\tilde{x})\big) - \kappa_1\Delta_y = 0, \\ &\dot{\Delta}_y = \Delta_x = 0. \end{aligned} \tag{27}$$

This gives $\Delta_x = \Delta_y = 0$, i.e., $(x_{\mathrm{s}}, y_{\mathrm{s}}) = (\tilde{x}, \tilde{y})$, which implies that each trajectory of (5a)-(5c) converges to a GEP of (4). By Lemma 6, we know that $(x,y) \to (x^*, \mathbf{1}_N\otimes y^*)$. This completes the proof. ■

Consider the special case of quadratic local costs

$$f_i(x) \triangleq x^T\Theta x + x^T\varphi_i, \qquad \Theta \succ 0,\ \varphi_i \in \mathbb{R}^n. \tag{28}$$

In this case, the spanning-tree-based m-strongly convex condition (7) holds with any $m \leq \underline{\lambda}(\Theta)$ and for any DST. Immediately, we have the following corollary:

Corollary 1: Under Assumptions 1-2, the resource allocation problem (1) with local costs (28) can be solved with the adaptive algorithm (5) for any initial conditions $(x(0), y(0), z(0) \in \mathbb{R}^{Nn}\times\mathbb{R}^{Nn}\times\mathbb{R}^{Nn}$ and any $a_{ij}(0) \in \mathbb{R}$, i.e., $(x,y) \to (x^*, \mathbf{1}_N\otimes y^*)$. Moreover, the adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}$, $k \in \mathcal{I}_{N-1}$, converge to some finite constant values.

Remark 5: The proposed adaptive resource allocation framework is essentially different from related literature [7], [10], [14]–[17], [21], which rely on the global knowledge of Laplacian eigenvalues to establish convergence results. The main idea behind the proof of Theorem 1 is to repeatedly use the Peter-Paul inequality (Lemma 1) so as to entrust stability to the adaptive coupling gains $\bar{a}_{k+1,\mathrm{p}_k}$. Thus, global stability can be derived by selecting sufficiently large ϕ_{k+1,p_k} with the help of the Schur complement and mathematical induction, as shown in the proof. As a consequence, the knowledge of the global Laplacian eigenvalues is successfully removed at the design stage. Note that the exact values of the parameters ϕ_{k+1,p_k}, $\forall k \in \mathcal{I}_{N-1}$, are not needed in the algorithm, they are only used for the purpose of stability analysis.

Remark 6: In addition to removing the knowledge of the global Laplacian eigenvalues as discussed above, it is worth noticing that the adaptive coupling gains (5d) overcome the need for unitary, or sufficiently small steps sizes to implement local gradient descent [4]–[6], [17]–[20]. The convergence of the proposed algorithm (5) is guaranteed globally for any parameters $\kappa_1, \kappa_2 \in \mathbb{R}^+$. These parameters can easily be tuned taking into account the fact that increasing κ_1 allows for larger step sizes towards decreasing the local costs (with constraint concerns), while increasing κ_2 enhances the importance of communicating Lagrangian multipliers. Generally speaking, a larger κ_1 would require a smaller integration step for practical implementation (i.e., smaller h in Algorithm 1), and larger κ_2 would induce higher steady-state coupling gains (cf. our simulations in Section V). Note that the above discussions also apply to the node-based case in Section IV.

IV. Distributed Adaptive Resource Allocation: Node-based Design

The DST-based adaptive law (5d) in Section III relies on the structural information of a DST. Although a DST can be obtained in a distributed way [38], it is of interest to possibly remove this intermediate step: to this purpose, a node-based design is developed in this section. Consider the following DARA algorithm for agent $i \in \mathcal{V}$:

$$
\begin{aligned}
\mathcal{O}^\alpha: \quad & \dot{x} = -\kappa_1(\nabla f(x) + y) && \text{(29a)}\\
& \dot{y} = x - D - \big((\mathcal{A}+\mathcal{B})\mathcal{L}\otimes \mathbf{I}_n\big)y - (\mathcal{L}\otimes \mathbf{I}_n)z && \text{(29b)}\\
& \dot{z} = \big((\mathcal{A}+\mathcal{B})\mathcal{L}\otimes \mathbf{I}_n\big)y && \text{(29c)}\\
& \dot{\alpha}_i = \beta_i := \kappa_2 \xi_i^T \xi_i && \text{(29d)}
\end{aligned}
$$

where $\mathcal{A} = \operatorname{diag}(\alpha_1, \cdots, \alpha_N)$, $\mathcal{B} = \operatorname{diag}(\beta_1, \cdots, \beta_N)$ and $\xi_i = \sum_{j\in\mathcal{N}_{\text{in}}(i)} w_{ij}(y_i - y_j)$.

Theorem 2: Under Assumptions 1-2, the adaptive algorithm (29) drives (x, y) to $(x^*, \mathbf{1}_N \otimes y^*)$ asymptotically for any initial condition $(x(0), y(0), z(0) \in \mathbb{R}^{Nn} \times \mathbb{R}^{Nn} \times \mathbb{R}^{Nn}$ and any $\alpha_i(0) \in \mathbb{R}^+$ provided there exists a scalar $m \in \mathbb{R}^+$, such that the following condition (referred to as *jointly m-strongly convexity*) holds $\forall x, y \in \mathbb{R}^{Nn}$:

$$
\begin{aligned}
&(x-y)^T(\mathcal{L}^T\mathcal{L}\otimes \mathbf{I}_n)(\nabla f(x) - \nabla f(y)) \geq \\
&\qquad m(x-y)^T(\mathcal{L}^T\mathcal{L}\otimes \mathbf{I}_n)(x-y). \qquad (30)
\end{aligned}
$$

Moreover, the adaptive gains α_i, $i \in \mathcal{I}_N$, converge to some finite constant values.

Proof: Following similar lines as the proof of Theorem 1, define the error vectors between the trajectory of (29a)-(29c) and any GEP $(\tilde{x}, \tilde{y}, \tilde{z})$ of (4) as (μ, ν, η) defined in (8a), and apply a change of coordinates:

$$
\hat{\mu} = (\mathcal{L}\otimes \mathbf{I}_n)\mu, \quad \hat{\nu} = (\mathcal{L}\otimes \mathbf{I}_n)\nu, \quad \hat{\eta} = (\mathcal{L}\otimes \mathbf{I}_n)\eta. \qquad (31)
$$

Note that $(\mathcal{A}+\mathcal{B})\mathcal{L} \in \mathcal{M}_{\text{r}}^N$. By Definition 1, we have $\mathcal{O}^\alpha|_{(\tilde{x},\tilde{y},\tilde{z})} = 0$. Denote $\xi = \operatorname{col}(\xi_1, \cdots, \xi_N)$, we have $\xi = (\mathcal{L}\otimes \mathbf{I}_n)y = (\mathcal{L}\otimes \mathbf{I}_n)(y - \tilde{y}) = \nu$. Then, in the new

Algorithm 2: DARA: node-based

Data: (1) initialization: $x_i(0)$, $y_i(0)$, $z_i(0)$, $a_i(0)$; (2) parameters: κ_1, κ_2

Result: Optimal resource allocation solution $x_i \to x^*$

$s \leftarrow 1$;
while $s \cdot h \leq T_{tml}$ **do** `// h is the integration step and` T_{tml} `is the terminal time`
  for $i \leftarrow 1$ **to** N **do**
    $\xi_i \leftarrow \sum_{j\in\mathcal{V}} \mathcal{L}_{ij} y_j$;
    $\beta_i \leftarrow \kappa_2 \xi_i^T \xi_i$;
    $\mathrm{d}x_i \leftarrow -\kappa_1(\nabla f_i(x_i) + y_i)$;
    $\mathrm{d}y_i \leftarrow x_i - d_i - (\alpha_i + \beta_i)\sum_{j\in\mathcal{V}} \mathcal{L}_{ij} y_j - \sum_{j\in\mathcal{V}} \mathcal{L}_{ij} z_j$;
    $\mathrm{d}z_i \leftarrow (\alpha_i + \beta_i)\sum_{j\in\mathcal{V}} \mathcal{L}_{ij} y_j$;
    $\mathrm{d}\alpha_i \leftarrow \beta_i$;
    $x_i \leftarrow x_i + h \cdot \mathrm{d}x_i$;
    $y_i \leftarrow x_i + h \cdot \mathrm{d}y_i$;
    $z_i \leftarrow x_i + h \cdot \mathrm{d}z_i$;
    $\alpha_i \leftarrow \alpha_i + h \cdot \mathrm{d}\alpha_i$;
  end
  $s \leftarrow s + 1$;
end

coordinates (31), the dynamics of $\mathcal{O}^\alpha$ is equivalent to

$$
\begin{aligned}
& \dot{\hat{\mu}} = -\kappa_1(\mathcal{L}\otimes \mathbf{I}_n)h - \kappa_1\hat{\nu} && \text{(32a)}\\
& \dot{\hat{\nu}} = \hat{\mu} - \big(\mathcal{L}(\mathcal{A}+\mathcal{B})\otimes \mathbf{I}_n\big)\hat{\nu} - (\mathcal{L}\otimes \mathbf{I}_n)\hat{\eta} && \text{(32b)}\\
& \dot{\hat{\eta}} = \big(\mathcal{L}(\mathcal{A}+\mathcal{B})\otimes \mathbf{I}_n\big)\hat{\nu} && \text{(32c)}\\
& \dot{\alpha}_i = \beta_i := \kappa_2\hat{\nu}_i^T\hat{\nu}_i, \ i \in \mathcal{I}_N && \text{(32d)}
\end{aligned}
$$

where $h = \nabla f(\mu + \tilde{x}) - \nabla f(\tilde{x})$ in (32a).

Consider the following candidate Lyapunov function:

$$
V_2 = \frac{N\big(2{\lambda_2}^2(\mathcal{L}^s) + 5\bar{\lambda}(\mathcal{L}^T\mathcal{L})\big)}{\epsilon_1{\lambda_2}^3(\mathcal{L}^s)} V_{\hat{\mu}} + V_{\hat{\nu}}^\alpha + \frac{5N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{{\lambda_2}^2(\mathcal{L}^s)} V_{\hat{\eta}} \qquad (33)
$$

where

$$
\begin{aligned}
V_{\hat{\mu}} &= \frac{1}{2}\hat{\mu}^T\hat{\mu}\\
V_{\hat{\nu}}^\alpha &= \frac{1}{2}\hat{\nu}^T\big((2\mathcal{A}+\mathcal{B})\otimes \mathbf{I}_n\big)\hat{\nu} + \sum_{i=1}^N \frac{1}{2\kappa_2}\big(\alpha_i(t) - \bar{\alpha}\big)^2\\
V_{\hat{\eta}} &= \frac{1}{2}(\hat{\nu}+\hat{\eta})^T(\hat{\nu}+\hat{\eta}) \qquad (34)
\end{aligned}
$$

where $\bar{\alpha}, \epsilon_1 \in \mathbb{R}^+$ remains to be decided.

The time derivative of $V_{\bar{\mu}}$ can be obtained as

$$
\dot{V}_{\hat{\mu}} = -\kappa_1\hat{\mu}^T(\mathcal{L}\otimes \mathbf{I}_n)h - \kappa_1\hat{\mu}^T\hat{\nu}. \qquad (35)
$$

By (31) and (30), we have

$$
\hat{\mu}^T(\mathcal{L}\otimes \mathbf{I}_n)h \geq m\hat{\mu}^T\hat{\mu}. \qquad (36)
$$

Similar to (15), we have

$$
\dot{V}_{\hat{\mu}} \leq (\epsilon_2 - \kappa_1 m)\hat{\mu}^T\hat{\mu} + \frac{\kappa_1^2}{4\epsilon_2}\hat{\nu}^T\hat{\nu} \qquad (37)
$$

where $\epsilon_2 \in \mathbb{R}^+$ is to be decided later.

The time derivative of $V_{\hat{\nu}}^{\alpha}$ can be obtained as

$$\begin{aligned}\dot{V}_{\hat{\nu}}^{\alpha} =&\frac{1}{2\kappa_2}\sum_{i=1}^{N}\big(2\alpha_i\dot{\beta}_i + 2\dot{\alpha}_i\beta_i + 2\beta_i\dot{\beta}_i + 2(\alpha_i-\bar{\alpha})\dot{\alpha}_i\big)\\ =&\frac{1}{\kappa_2}\sum_{i=1}^{N}\big((\alpha_i+\beta_i)\dot{\beta}_i + (\alpha_i+\beta_i-\bar{\alpha})\dot{\alpha}_i\big)\\ =&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-\bar{\alpha}\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} + 2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\dot{\hat{\nu}}\\ =&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-\bar{\alpha}\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} + 2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\mu}\\ &-2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\mathcal{L}(\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\nu}\\ &-2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\mathcal{L}\otimes\mathbf{I}_n\big)\hat{\eta}. \end{aligned}\tag{38}$$

Let $\tilde{\nu} = \big((\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\nu} \triangleq \mathrm{col}(\tilde{\nu}_1,\cdots,\tilde{\nu}_N)$. Based on Lemma 2, we have

$$\begin{aligned}&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\mathcal{L}(\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\nu}\\ =&\tilde{\nu}^T(\mathcal{L}^{\mathrm{s}}\otimes\mathbf{I}_n)\tilde{\nu} = \sum_{k=1}^{n}\tilde{\delta}_k^T\mathcal{L}^{\mathrm{s}}\tilde{\delta}_k\end{aligned}\tag{39}$$

where $\tilde{\delta}_k = \big([S\tilde{\nu}_1]_k,[S\tilde{\nu}_2]_k,\cdots,[S\tilde{\nu}_N]_k\big)^T$. Here $S\in\mathbb{R}^{n\times n}$ is an orthogonal matrix. Note that $[S\tilde{\nu}_i]_k = (\alpha_i+\beta_i)\sum_{j\in\mathcal{N}_{\rm in}(i)}w_{ij}([S\nu_i]_k - [S\nu_j]_k)$. If we denote $\delta_k = \big([S\nu_1]_k,[S\nu_2]_k,\cdots,[S\nu_N]_k\big)^T$, then $\tilde{\delta}_k = (\mathcal{A}+\mathcal{B})\mathcal{L}\delta_k$, $\forall k\in\mathcal{I}_n$. Under Assumption 2, there holds $\big((\mathcal{A}+\mathcal{B})^{-1}\mathbf{1}_N\big)^T\tilde{\delta}_k = \mathbf{1}_N^T\mathcal{L}\delta_k = 0$ for any k. Note that $(\mathcal{A}+\mathcal{B})^{-1}\mathbf{1}_N\in\mathbb{R}_+^N$ ($\alpha_i(0)\in\mathbb{R}^+$ and $\dot{\alpha}_i\geq 0$). So, it follows from (39) and statements 1)-2) of Lemma 4 that

$$\begin{aligned}&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\mathcal{L}(\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\nu}\\ =&\sum_{k=1}^{n}\tilde{\delta}_k^T\mathcal{L}^{\mathrm{s}}\tilde{\delta}_k\\ \geq&\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{N}\sum_{k=1}^{n}\tilde{\delta}_k^T\tilde{\delta}_k\\ =&\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{N}\tilde{\nu}^T\tilde{\nu}\\ =&\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu}.\end{aligned}\tag{40}$$

Then, it follows from (38) and (40) that

$$\begin{aligned}\dot{V}_{\hat{\nu}}^{\alpha} \leq&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-\bar{\alpha}\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} + 2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\otimes\mathbf{I}_n\big)\hat{\mu}\\ &-\frac{2\lambda_2(\mathcal{L}^{\mathrm{s}})}{N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu}\\ &-2\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})\mathcal{L}\otimes\mathbf{I}_n\big)\hat{\eta}\\ \leq&\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-\bar{\alpha}\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} + \frac{2N}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\mu}^T\hat{\mu}\\ &+\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{2N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu} + \frac{N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\eta}^T\hat{\eta}\\ &-\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu}\\ =&\frac{2N}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\mu}^T\hat{\mu} - \frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{2N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu}\\ &+\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-\bar{\alpha}\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} + \frac{N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\eta}^T\hat{\eta}\end{aligned}\tag{41}$$

where we have repeatedly used Lemma 1 to get the second inequality.

Similar to (21), the time derivative of $V_{\hat{\eta}}$ can be obtained as

$$\begin{aligned}\dot{V}_{\hat{\eta}} =&\hat{\nu}^T\hat{\mu} - \hat{\nu}^T(\mathcal{L}\otimes\mathbf{I}_n)\hat{\eta} + \hat{\eta}^T\hat{\mu} - \hat{\eta}^T(\mathcal{L}\otimes\mathbf{I}_n)\hat{\eta}\\ \leq&\frac{1}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\mu}^T\hat{\mu} + \Big(\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{2} + \frac{\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\Big)\hat{\nu}^T\hat{\nu} - \frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{4}\hat{\eta}^T\hat{\eta}.\end{aligned}\tag{42}$$

Here we have used the fact that $\hat{\eta}^T(\mathcal{L}\otimes\mathbf{I}_n)\hat{\eta}\geq\lambda_2(\mathcal{L}^{\mathrm{s}})\hat{\eta}^T\hat{\eta}$, which is guaranteed by statement 3) of Lemma 4, and $(\mathbf{1}_N^T\otimes\mathbf{I}_n)\hat{\eta}=0$ under Assumption 2.

Based on (33), (37), (41), and (42) and with some manipulations, the time derivative of V_2 along the trajectory of (32) is upper bounded by

$$\begin{aligned}\dot{V}_2\leq&-\frac{N\big(2{\lambda_2}^2(\mathcal{L}^{\mathrm{s}}) + 5\bar{\lambda}(\mathcal{L}^T\mathcal{L})\big)(\kappa_1 m - \epsilon_1 - \epsilon_2)}{\epsilon_1{\lambda_2}^3(\mathcal{L}^{\mathrm{s}})}\hat{\mu}^T\hat{\mu}\\ &-\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{2N}\hat{\nu}^T\big((\mathcal{A}+\mathcal{B})^2\otimes\mathbf{I}_n\big)\hat{\nu}\\ &+\hat{\nu}^T\big((\mathcal{A}+\mathcal{B}-(\bar{\alpha}-\gamma')\mathbf{I}_N)\otimes\mathbf{I}_n\big)\hat{\nu} - \frac{N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{4\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\eta}^T\hat{\eta}\end{aligned}\tag{43}$$

where $\gamma'\in\mathbb{R}^+$ is given by

$$\begin{aligned}\gamma' =&\frac{N\kappa_1^2\big(2{\lambda_2}^2(\mathcal{L}^{\mathrm{s}}) + 5\bar{\lambda}(\mathcal{L}^T\mathcal{L})\big)}{4\epsilon_1\epsilon_2{\lambda_2}^3(\mathcal{L}^{\mathrm{s}})}\\ &+\frac{5N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{2\lambda_2(\mathcal{L}^{\mathrm{s}})} + \frac{5N\bar{\lambda}^2(\mathcal{L}^T\mathcal{L})}{{\lambda_2}^3(\mathcal{L}^{\mathrm{s}})}.\end{aligned}\tag{44}$$

Let us select ϵ_1,ϵ_2 such that $\epsilon_1+\epsilon_2\leq\kappa_1 m$, and $\bar{\alpha}\geq\gamma' + \frac{N}{2\lambda_2(\mathcal{L}^{\mathrm{s}})}$. Then, it follows from (43) that

$$\begin{aligned}\dot{V}_2\leq&-\frac{\lambda_2(\mathcal{L}^{\mathrm{s}})}{2N}\hat{\nu}^T\Big(\big(\mathcal{A}+\mathcal{B}-\frac{N}{\lambda_2(\mathcal{L}^{\mathrm{s}})}\mathbf{I}_N\big)^2\otimes\mathbf{I}_n\Big)\hat{\nu}\\ &-\frac{N\bar{\lambda}(\mathcal{L}^T\mathcal{L})}{4\lambda_2(\mathcal{L}^{\mathrm{s}})}\hat{\eta}^T\hat{\eta}\leq 0,\end{aligned}\tag{45}$$

implying that V_2 has a finite limit and all the signals $\hat{\mu}$, $\hat{\nu}$, $\hat{\eta}$, and α_i are bounded. The rest of the proof follows similarly to that of Theorem 1. ■

Note that for local costs (28), the jointly m-strongly convex condition (30) also holds with any $m\leq\underline{\lambda}(\Theta)$, resulting in the following corollary:

Corollary 2: Under Assumptions 1-2, the resource allocation problem (1) with quadratic local costs (28) can be solved with the adaptive algorithm (29) for any initial condition $(x(0),y(0),z(0)\in\mathbb{R}^{Nn}\times\mathbb{R}^{Nn}\times\mathbb{R}^{Nn}$ and any $\alpha_i(0)\in\mathbb{R}^+$, i.e., $(x,y)\to(x^*,\mathbf{1}_N\otimes y^*)$. Moreover, the adaptive gains α_i, $i\in\mathcal{I}_N$, converge to some finite constant values.

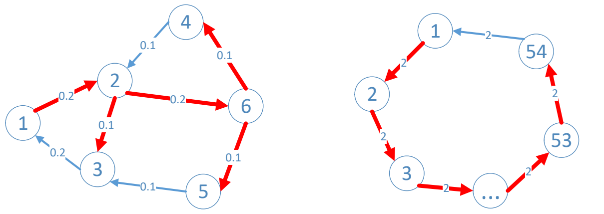
Remark 7: Although both algorithms can be recast as uncertain saddle-point dynamics, the ideas behind the DST-based and node-based designs for promoting the consensus over y_i are intrinsically different. In the DST-based case, the root of the DST plays the role of a leader; while in the node-based case, there is no leader and all the nodes play the same role. This shows the flexibility of the uncertain saddle-point dynamics viewpoint to accommodate for different design perspectives.

Remark 8: The proposed conditions, either the spanning-tree-based or the jointly strongly convexity, are slightly more conservative as compared with classical strongly convexity. The reason is due to the induced Laplacian matrices. Nevertheless, Corollaries 1-2 have shown a relatively standard class of local cost functions that automatically satisfies the proposed strongly convexity conditions.

V. Simulations

In this section, we give two examples to show the effectiveness of the proposed methods. For each example, we consider two cases to model networks of different scales (cf. Fig. 1). The first example considers cost functions with randomly generated coefficients, while the second example is inspired by the benchmark power networks IEEE 30-bus and IEEE 118-bus for which the systems diagrams and data sets are available online at [40] and [41], respectively.

In addition to validate the effectiveness of the proposed algorithms, some other goals of the experiments include: to support Remark 6 in that the convergence of the proposed algorithms is guaranteed globally for any parameters $\kappa_1, \kappa_2 \in \mathbb{R}^+$ (cf. Fig. 4); to highlight the necessity of introducing the adaptive coupling strategies (cf. Fig. 7); to give a comparison with the method proposed in the literature [14] (cf. Fig. 10).



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$

Fig. 1. Two balanced digraphs. The selected DSTs for the DST-based protocol (5) are highlighted with thicker red lines.

Example 1: Consider a total resource d to be allocated over a network of N agents that communicate via a weight-balanced digraph $\mathcal{G}$. The local cost function for each agent is given by $f_i(x_i) = a_i x_i^2 + b_i x_i + c_i$, where $a_i = 0.1$, $c_i = 0$, and b_i are randomly selected in the interval $[1, 100]$. In the following, two cases will be simulated. In each case, the local resources are equally distributed as $d_i = \frac{d}{N}$; the initial (x_i, y_i, z_i) of the agents are chosen from a Gaussian distribution with standard deviation 5. For the DST-based design (5), the initial $a_{ij}(0)$ are chosen from a uniform distribution in $(-1, 1)$; for the node-based design (29), the initial $\alpha_i(0)$ are chosen from a uniform distribution in $(0, 1)$.

Case 1: $d = 1.5 \times 10^3$, $N = 6$, $\mathcal{G} = \mathcal{G}_1$ (Fig. 1(a));

Select $\kappa_1 = \kappa_2 = 1$ for both the DST-based and node-based designs. The states of the agents and the corresponding adaptive gains under (5) and (29) are provided in Figs. 2 and 3, respectively, where the dashed lines represent the local optimal allocation decisions. For comparison, Fig. 4 shows the results under (5) with a pair of different parameters $\kappa_1 = 10$ and $\kappa_2 = 0.1$.

Case 2: $d = 1.5 \times 10^4$, $N = 54$, $\mathcal{G} = \mathcal{G}_2$ (Fig. 1(b));

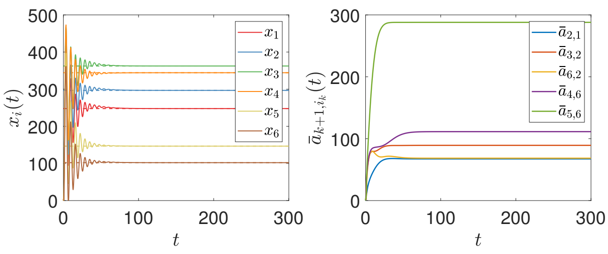


Fig. 2. Case 1: States $x_i(t)$ of the agents and adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ with DST-based protocol (5) and parameters $\kappa_1 = \kappa_2 = 1$. The states $x_i(t)$ converge to the corresponding optimal allocation decisions, and the adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ converge to finite constants.

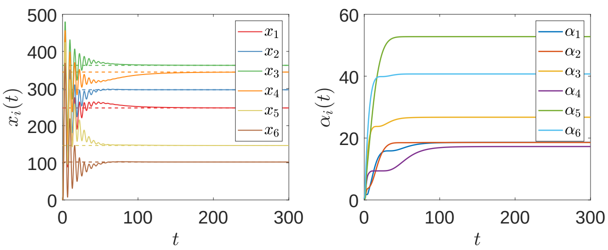


Fig. 3. Case 1: States $x_i(t)$ of the agents and adaptive gains $\alpha_i(t)$ with node-based protocol (29) and parameters $\kappa_1 = \kappa_2 = 1$. The states $x_i(t)$ converge to the corresponding optimal allocation decisions, and the adaptive gains $\alpha_i(t)$ converge to finite constants.

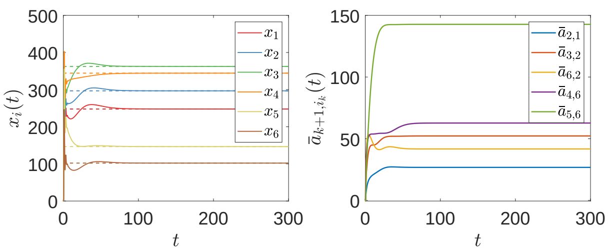


Fig. 4. Case 1: States $x_i(t)$ of the agents and adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ with DST-based protocol (5) and parameters $\kappa_1 = 10$, $\kappa_2 = 0.1$. A larger κ_1 leads to better transient performance of $x_i(t)$ and a smaller κ_2 leads to smaller steady values of $\bar{a}_{k+1,\mathrm{p}_k}(t)$, as compared to Fig. 2.

Select $\kappa_1 = \kappa_2 = 0.1$ for both the DST-based and node-based designs. The states of the agents and the corresponding adaptive gains under (5) and (29) are provided in Figs. 5 and 6, respectively. For comparison, let $\kappa_2 = 0$, which is the static strategy used in many related works, e.g., [4]–[6]. It can be seen from Fig. 7 that the resulting *nonadaptive* strategy fails to solve the resource allocation problem. The reason is that the results in the aforementioned works cannot be adapted to the case with directed communication graphs.

Example 2: In this example, we examine the proposed algorithms applied to the relaxed (i.e., without box constraints) economic dispatch (rED) problem. We consider two benchmark power networks, IEEE 30-bus and IEEE 118-bus, where N power generators must cooperatively minimize the cumulative cost, while meeting a total load demand d. In both benchmarks, the cost functions of the generators are of quadratic form: $f_i(x_i) = a_i x_i^2 + b_i x_i + c_i$.

Two observations follow when comparing our algorithms

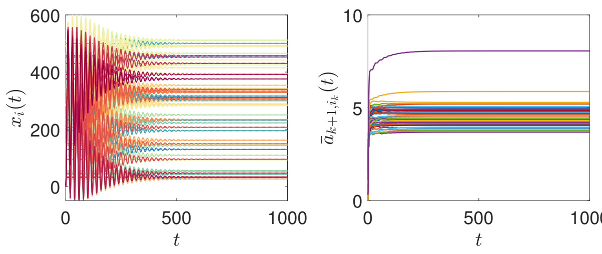


Fig. 5. Case 2: States $x_i(t)$ of the agents and adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ with DST-based protocol (5) and parameters $\kappa_1 = \kappa_2 = 0.1$.

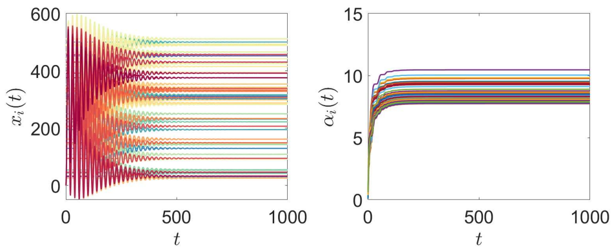


Fig. 6. Case 2: States $x_i(t)$ of the agents and adaptive gains $\alpha_i(t)$ with node-based protocol (29) and parameters $\kappa_1 = \kappa_2 = 0.1$.

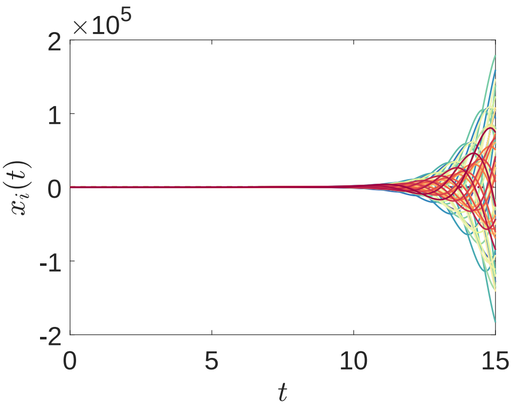


Fig. 7. Case 2: States $x_i(t)$ of the agents with *nonadaptive* protocol ($\kappa_1 = 0.1$, $\kappa_2 = 0$ in (29)). The states $x_i(t)$ diverge.

with the Laplacian-gradient dynamics proposed in [13] for the rED problem: first, in our algorithms the knowledge of the cost functions (or the corresponding gradients) of neighbors is not needed for each generator, which makes our algorithms more privacy-friendly; second, our algorithms are initialization-free (i.e., the initial decisions do not need to satisfy the total load demand). In fact, the initialization-free problem in [13] has also been overcome in [14] by a "dynamic average consensus + Laplacian-gradient" (DAC+LG) algorithm defined as follows:

$$
\begin{aligned}
\dot{x} &= -(\mathcal{L} \otimes \mathbf{I}_n)\nabla f(x) + \kappa_1 y \\
\dot{y} &= -\kappa_2 (x - D) - \alpha y - \beta(\mathcal{L} \otimes \mathbf{I}_n) y - z \\
\dot{z} &= \alpha\beta(\mathcal{L} \otimes \mathbf{I}_n) y
\end{aligned} \tag{46}
$$

where $\kappa_1, \kappa_2, \alpha, \beta \in \mathbb{R}^+$ are tuned based on the Laplacian eigenvalues. Nevertheless, the exchange of the gradients through the network is still needed. Besides, without the adjustable parameter for gradient descent, DAC+LG may suffer from a slower convergence rate (cf. our case study below).

Case 1 (IEEE 30-bus): $d = 10^3$, $N = 6$, $\mathcal{G} = \mathcal{G}_1$ (Fig. 1(a));

The power system contains 6 generators. The parameters of the local costs are described in vector form by $a_i = (0.00375, 0.0175, 0.0625, 0.00834, 0.025, 0.025)^T$, $b_i = (2, 1.75, 1, 3.25, 3, 3)^T$, and $c_i = 0$ [42]. The power allocation states and the corresponding adaptive gains under (5) and (29) are provided in Figs. 8 and 9, respectively, where the dashed lines represent the local optimal power allocation decisions.

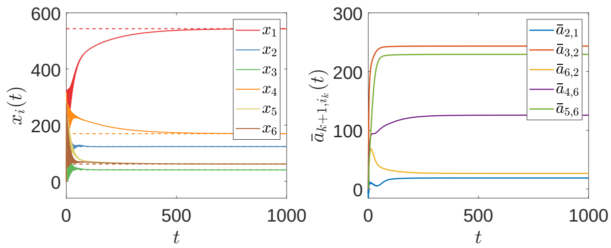


Fig. 8. Case 1 (IEEE 30-bus): Power allocation states $x_i(t)$ and adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ with DST-based protocol (5) and parameters $\kappa_1 = 20$, $\kappa_2 = 1$.

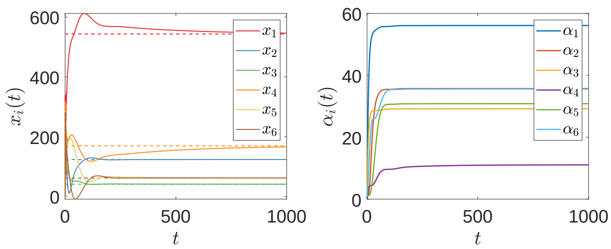


Fig. 9. Case 1 (IEEE 30-bus): Power allocation states $x_i(t)$ and adaptive gains $\alpha_i(t)$ with node-based protocol (29) and parameters $\kappa_1 = 20$, $\kappa_2 = 1$.

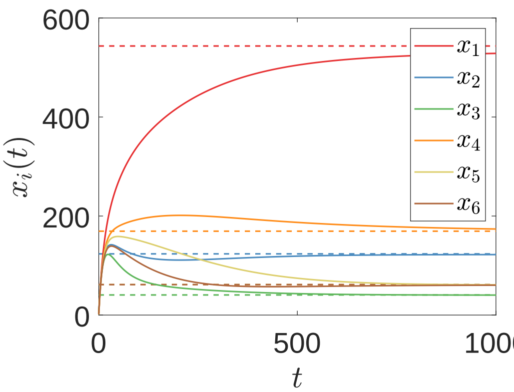


Fig. 10. Case 1 (IEEE 30-bus): Power allocation states $x_i(t)$ with DAC+LG (46) and parameters $\kappa_1 = \kappa_2 = 1$, $\alpha = 10$, $\beta = 60$. The parameters are tuned based on the Laplacian eigenvalues [14, Theorem 5.3].

Case 2 (IEEE 118-bus): $d = 10^5$, $N = 54$, $\mathcal{G} = \mathcal{G}_2$ (Fig. 1(b));

The power system contains 54 generators. The parameters of the local costs belong to the ranges $a_i \in (0.0024, 0.0697)$, $b_i \in (8.3391, 37.6961)$, and $c_i \in (6.78, 74.33)$ [41]. The power allocation states and the corresponding adaptive gains under (5) and (29) are provided in Figs. 11 and 12, respectively, where the dashed lines represent the local optimal power allocation decisions.

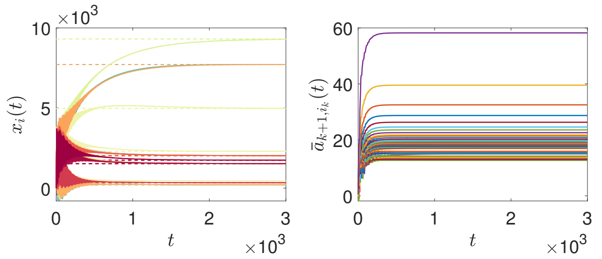


Fig. 11. Case 2 (IEEE 118-bus): Power allocation states $x_i(t)$ and adaptive gains $\bar{a}_{k+1,\mathrm{p}_k}(t)$ with DST-based protocol (5) and parameters $\kappa_1 = \kappa_2 = 0.3$.

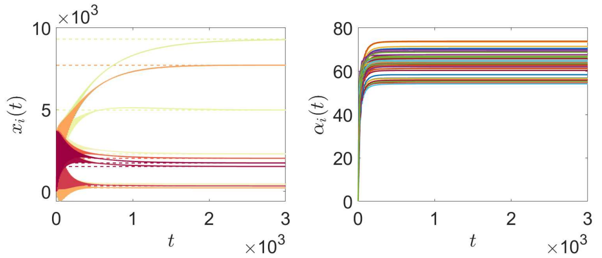


Fig. 12. Case 2 (IEEE 118-bus): Power allocation states $x_i(t)$ and adaptive gains $\alpha_i(t)$ with node-based protocol (29) and parameters $\kappa_1 = \kappa_2 = 0.3$.

When comparing Case 2 to Case 1, one can find that the steady-state gains for $N = 54$ have smaller orders of magnitude as those for $N = 6$. Therefore, we conclude the section by commenting on the lower bound $\bar{\alpha} \geq \gamma' + \frac{N}{2\lambda_2(\mathcal{L}^s)}$ introduced before (45). Although this bound increases for increasing N, it is only used for stability analysis of algorithm (29), and might be conservative in practice, as discussed in the literature [25], and as evident from our simulations. As a matter of fact, our simulations show that the actual values attained by the adaptive gains are not influenced by the scale N of the network, but mainly depend on the network structure and the parameter κ_2.

VI. Discussions

[**On further comparisons between DST- and node-based algorithms**] The DST-based method can in general lead to faster convergence (see Fig. 2-3 in Section V, and the simulation results in [31] for a distributed optimization problem). This is consistent with intuition since enhancing connections along a DST structure should be more efficient than enhancing connections of all links. Note that a DST structure is known in the literature to be beneficial for cooperative consensus [24], [36].

[**On the superiority between DST- and node-based algorithms**] Different constraints in real-world applications would decide the superiority between these two algorithms. If faster convergence speed is desired, the DST-based method would be preferable, where the DST structure could be identified via a breadth/depth first algorithm [43, Section 1.4.4] or distributed algorithms [38]. If a fully distributed strategy that does not rely on any a priori information is desired, the node-based algorithm would be preferable, since the DST-based method requires a priori knowledge of a DST structure.

[**On the open problems of the DARA algorithms**] Note that the Lyapunov functions in (11) and (33) are quadratic. Since results exist where a non-quadratic Lyapunov function may improve performance in adaptive schemes, see e.g. [44], [45], an open future direction is to improve the proposed adaptive resource allocation solutions via non-quadratic Lyapunov functions. Besides, the DARA algorithms in this paper have been formulated for resource allocation problems without local bound constraints. Such local bound constraints may appear in engineering applications such as economic dispatch in the field of power networks. Embedding local bound constraints in the proposed saddle-point dynamics viewpoint is thus a challenge for future work.

VII. Conclusions

Distributed optimal in-network resource allocation over weight-balanced digraphs was studied. Two novel distributed adaptive saddle-point algorithms named DST-based and node-based algorithms have been proposed. The asymptotic convergence of each algorithm has been theoretically proved and numerically tested. The proposed adaptive resource allocation frameworks successfully remove the knowledge of the underlying Laplacian eigenvalues, which has been widely used in related literature. Future work includes relaxing the proposed conditions (7) and (30), and studying resource allocation problems with local bound constraints.

Acknowledgment

The first author would like to thank Prof. Jie Mei for the valuable discussions and the anonymous reviewers for their constructive comments.

References

[1] T. Ibaraki and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*. Cambridge, MA, USA: MIT Press, 1988.

[2] J. W. Simpson-Porco, B. K. Poolla, N. Monshizadeh, and F. Dörfler, "Input-output performance of linear-quadratic saddle-point algorithms with application to distributed resource allocation problems," *IEEE Trans. Autom. Control*, vol. 65, no. 5, pp. 2032–2045, 2020.

[3] W.-T. Lin, Y.-W. Wang, C. Li, and X. Yu, "Distributed resource allocation via accelerated saddle point dynamics," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 9, pp. 1588–1599, 2021.

[4] P. Yi, Y. Hong, and F. Liu, "Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems," *Automatica*, vol. 74, pp. 259–269, 2016.

[5] B. Gharesifard, T. Başar, and A. D. Domínguez-García, "Price-based coordinated aggregation of networked distributed energy resources," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2936–2946, 2016.

[6] Z. Li and Z. Ding, "Distributed multiobjective optimization for network resource allocation of multiagent systems," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 5800–5810, 2020.

[7] Q. Liu, X. Le, and K. Li, "A distributed optimization algorithm based on multiagent network for economic dispatch with region partitioning," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2466–2475, 2021.

[8] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2407–2418, 2018.

[9] A. Nedić, A. Olshevsky, and W. Shi, "Improved convergence rates for distributed resource allocation," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 172–177.

[10] L. Bai, C. Sun, Z. Feng, and G. Hu, "Distributed continuous-time resource allocation with time-varying resources under quadratic cost functions," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 823–828.

[11] H. Yun, H. Shim, and H.-S. Ahn, "Initialization-free privacy-guaranteed distributed algorithm for economic dispatch problem," *Automatica*, vol. 102, pp. 86–93, 2019.

[12] P. Dai, W. Yu, G. Wen, and S. Baldi, "Distributed reinforcement learning algorithm for dynamic economic dispatch with unknown generation cost functions," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2258–2267, 2020.

[13] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 3, pp. 226–237, 2015.

[14] A. Cherukuri and J. Cortés, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, 2016.

[15] Z. Deng, S. Liang, and Y. Hong, "Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3116–3125, 2018.

[16] Z. Deng, X. Nian, and C. Hu, "Distributed algorithm design for nonsmooth resource allocation problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3208–3217, 2020.

[17] S. S. Kia, "Distributed optimal in-network resource allocation algorithm design via a control theoretic approach," *Syst. Control Lett.*, vol. 107, pp. 49–57, 2017.

[18] S. Liang, X. Zeng, and Y. Hong, "Distributed sub-optimal resource allocation over weight-balanced graph via singular perturbation," *Automatica*, vol. 95, pp. 222–228, 2018.

[19] H. Li, Q. Lv, and T. Huang, "Convergence analysis of a distributed optimization algorithm with a general unbalanced directed communication network," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 237–248, 2019.

[20] J. Zhang, K. You, and K. Cai, "Distributed dual gradient tracking for resource allocation in unbalanced networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2186–2198, 2020.

[21] Y. Zhu, W. Ren, W. Yu, and G. Wen, "Distributed resource allocation over directed graphs via continuous-time algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 2, pp. 1097–1106, 2019.

[22] J. Mei, W. Ren, J. Chen, and B. D. O. Anderson, "Consensus of linear multi-agent systems with fully distributed control gains under a general directed graph," in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 2993–2998.

[23] Z. Li, G. Wen, Z. Duan, and W. Ren, "Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1152–1157, 2015.

[24] W. Yu, J. Lü, X. Yu, and G. Chen, "Distributed adaptive control for synchronization in directed complex networks," *SIAM J. Control Optim.*, vol. 53, no. 5, pp. 2980–3005, 2015.

[25] J. Mei, W. Ren, and J. Chen, "Distributed consensus of second-order multi-agent systems with heterogeneous unknown inertias and control gains under a directed graph," *IEEE Trans. Autom. Control*, vol. 61, no. 8, pp. 2019–2034, 2016.

[26] Z. Li, L. Gao, W. Chen, and Y. Xu, "Distributed adaptive cooperative tracking of uncertain nonlinear fractional-order multi-agent systems," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 292–300, 2019.

[27] G. Wen, G. Hu, Z. Zuo, Y. Zhao, and J. Cao, "Robust containment of uncertain linear multi-agent systems under adaptive protocols," *Int. J. Robust Nonlinear Control*, vol. 27, no. 12, pp. 2053–2069, 2017.

[28] D. Yue, S. Baldi, J. Cao, Q. Li, and B. De Schutter, "A directed spanning tree adaptive control solution to time-varying formations," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 690–701, 2021.

[29] D. Yue, J. Cao, Q. Li, and M. Abdel-Aty, "Distributed neuro-adaptive formation control for uncertain multi-agent systems: node- and edge-based designs," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2656–2666, 2020.

[30] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1434–1441, 2017.

[31] D. Yue, S. Baldi, J. Cao, and B. De Schutter, "Distributed adaptive optimization with weight-balancing," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 2068–2075, 2022.

[32] Q. Yang, G. Chen, and T. Wang, "ADMM-based distributed algorithm for economic dispatch in power systems with both packet drops and communication delays," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 842–852, 2020.

[33] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*. Wiley New York, 1995.

[34] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*. Springer, 2008.

[35] C. W. Wu and L. O. Chua, "Synchronization in an array of linearly coupled dynamical systems," *IEEE Trans. Circuits Syst. I, Reg. Papers,*, vol. 42, no. 8, pp. 430–447, 1995.

[36] W. Yu, G. Chen, M. Cao, and J. Kurths, "Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 881–891, 2010.

[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. NewYork, NY, USA: Cambrige Univ. Press, 2004.

[38] P. Humblet, "A distributed algorithm for minimum weight directed spanning trees," *IEEE Trans. Commun.*, vol. 31, no. 6, pp. 756–762, 1983.

[39] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994, vol. 15.

[40] "IEEE 30 bus test case," https://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.htm.

[41] "IEEE 118 bus test case," https://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm.

[42] W. Yu, C. Li, X. Yu, G. Wen, and J. Lü, "Economic power dispatch in smart grids: a framework for distributed optimization and consensus dynamics," *Sci. China Inf. Sci.*, vol. 61, no. 1, pp. 1–16, 2018.

[43] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[44] G. Tao, "Model reference adaptive control with $L^{1+\alpha}$ tracking," *Int. J. Control*, vol. 64, no. 5, pp. 859–870, 1996.

[45] M. Hosseinzadeh and M. J. Yazdanpanah, "Performance enhanced model reference adaptive control through switching non-quadratic lyapunov functions," *Syst. Control Lett.*, vol. 76, pp. 47–55, 2015.