Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 5191–5196

Distributed Low-Effort Load Balancing in the Presence of Time-Delays ⋆

Themistoklis Charalambous * **Stefano Panzieri** ** **Gabriele Oliva** ***

* *Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus. E-mail:* charalambous.themistoklis@ucy.ac.cy

** *University Roma Tre, via della Vasca Navale 79, 00146 Rome, Italy. E-mail:* stefano.panzieri@uniroma3.it

*** *Unit of Automatic Control, Department of Engineering, Università Campus Bio-Medico di Roma, Rome, Italy. E-mail:* g.oliva@unicampus.it

Abstract: In this paper, we investigate the problem of distributed load balancing under network capacity constraints, where the participating agents cooperate with the aim of jointly minimizing both the workload disparity among them as well as the overall workload transfer. Classical approaches for asymptotic convergence to the global optimum in a distributed fashion typically assume timely exchange of information between neighboring agents of a given multi-agent system. This assumption is not necessarily valid in practical settings due to non-commensurate (heterogeneous) communication and processing delays that might affect transmissions at different times. More specifically, we consider what effect multiple heterogeneous time-varying delays, among the agents can have on the distributed load balancing problem. We show that the distributed load balancing problem under bounded heterogeneous time-varying delays is globally asymptotically stable, but the rate of convergence is affected. Bounds on the convergence rate are provided with respect to an upper bound on the delays. Simulation examples are provided to show the validity and performance of our theoretical results.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Distributed load balancing, time-delays, distributed algorithms, minimum effort.

1. INTRODUCTION

1.1 Motivation

Load balancing is a topic well investigated since the 1970's, with seminal works for centralized approaches; see, e.g., Chow et al. (1979); Perkins and Kumark (1989). Load balancing is inherently an optimization problem and the, recently, enlarged scale and heterogeneity of multi-agent systems (such as modern cloud infrastructures) makes it very challenging to solve such an optimization problem in a centralized fashion. Indeed, gathering all the required information from thousands of agents centrally and solving the problem by a single solver is not ideal as *i)* the solver may become a single point of failure, *ii)* the uplink of data to the solver may become congested, and *iii)* the gathered data may become obsolete by the time the optimization is solved.

Nowadays, the increased computational capabilities of processors (being computers or machines) and the advancement of networking/communication technologies facilitate the consideration of practical distributed algorithms, in which agents can take decisions cooperatively, with no need for central coordination achieving scalability and low complexity.

1.2 Related Work

In the literature, a number of distributed load balancing (DLB) strategies have been put suggested. References such as Antonis et al. (2004); Jiang et al. (2015); Sohn and Lee (2016) serve as examples of hierarchical approaches that have as basis the construction of virtual trees among the nodes in the network in order to balance loads; Jiang et al. (2015) also provides an algorithm that balances loads despite network unreliability; Sohn and Lee (2016) also proposes an algorithm to balance the loads in terms of users at the base stations of cellular networks. An approach deploying noncooperative game theory was utilized in Penmatsa and Chronopoulos (2011) for modeling and solving the problem of load balancing, deriving a distributed algorithm which delivers a near-optimal solution. Additionally, Suraci et al. (2017); Pietrabissa et al. (2018) proposed non-cooperative algorithms based on Wardrop equilibria and mean-field game theory. In Christ et al. (2019), an algorithm based on a min-max optimization approach is suggested for workload balancing on non-identical parallel processors. In Hanada et al. (2019) the problem of dis-

⋆ This work was partly supported by the project MINERVA, funded by the European Research Council (ERC) under the European Union's Horizon 2022 research and innovation programme (Grant agreement No. 101044629). This work was also partly supported by POR FESR Lazio Region Project RESIM under grant n. 228 A0375-2020-36673 (CUP: F89J21004860008).

2405-8963 Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2023.10.114

tributed power dispatch with load balancing is considered, focusing on the identification of a power dispatch of a number of generators in a grid that has limits on the total required load and on the capacities. The approach is distributed and is based on a weighted average consensus protocol over a noisy network. Other recent work at the state of the art include: Zhang et al. (2021), where a game-theoretical approach to balance the loads of edge computing servers is provided, even though each server can interact with each other; Li et al. (2022), where the authors develop an optimal but centralized data placement problem with capacity and load balancing constraints; Ornatelli et al. (2021) where a hierarchical load balancing approach that relies on a centralized controller is provided.

Finally, in Oliva et al. (2022), a variation of the classical distributed load balancing optimization problem is considered, where agents try to minimize both the workload disparity from other agents as well as the overall workload transfer, while satisfying some network capacity limitations. This is often desirable, since transferring load to achieve balancing may require an effort (and an associated cost). Oliva et al. (2022) proposed a distributed algorithm for solving the considered distributed optimization problem, but they did not consider what happens when the exchange of information among the agents experiences delays.

1.3 Contributions

In this paper, we investigate the same problem as in Oliva et al. (2022) in the presence of bounded delays. The distributed algorithm proposed in Oliva et al. (2022) now is modified such that each agent updates its state at a specific time step by utilizing its own value combined with the possibly delayed data obtained at that time step by its neighbors. We establish that this form of the distributed algorithm proposed in Oliva et al. (2022) converges to the optimal solution, despite the presence of arbitrary but bounded time-delays. Bounds on the convergence rate of the algorithm are provided with respect to an upper bound on the delays. The validity of our result and performance of the distributed algorithm are demonstrated via an illustrative example.

2. NOTATION AND PRELIMINARIES

2.1 Notation

We use boldface lowercase (uppercase) letters to denote vectors (matrices). By $\mathbf{0}_n$ ($\mathbf{1}_n$) we represent vectors with n entries, all equal to zero (one). Moreover, by $0_{n\times m}$ ($1_{n\times m}$) we represent the $n \times m$ matrix containing just zeros (ones). By $\mathbb{R}$ ($\mathbb{R}_{\geq 0}$) and $\mathbb{N}$ ($\mathbb{N}_0$) we denote the sets of real (nonnegative) and natural (nonnegative) numbers, respectively. The i-th component of a vector $\boldsymbol{x}$ is denoted by x_i. We use the notation $\boldsymbol{x} \geq \mathbf{0}$ ($\boldsymbol{x} > \mathbf{0}$) when all of the components of $\boldsymbol{x}$ are non-negative (positive). The inequality $\boldsymbol{x} \geq \mathbf{y}$ implies that $x_i \geq y_i$ for all components i. We refer to the (i, j)-th entry of a matrix A by A_{ij}.

We use $\|\boldsymbol{w}\|_2$ to denote the Euclidean norm of a vector $\boldsymbol{w}$. Given a vector $\boldsymbol{w} \in \mathbb{R}^n$ with $\boldsymbol{w} > \mathbf{0}_n$, $\|\cdot\|_\infty^{\boldsymbol{w}}$ stands for the weighted maximum norm, i.e., $\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} = \max_i |\frac{x_i}{w_i}|$. The vector norm $\|\cdot\|_\infty^{\boldsymbol{w}}$ induces a matrix norm, also denoted by $\|\cdot\|_\infty^{\boldsymbol{w}}$, defined by

$$\|M\|_\infty^{\boldsymbol{w}} = \max_{\boldsymbol{x}\neq 0} \frac{\|M\boldsymbol{x}\|_\infty^{\boldsymbol{w}}}{\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}}.$$

When $w_i = 1$ for all i, we suppress the superscript $\mathbf{w}$. A sequence $\{\mathbf{x}(k)\} \in \mathbb{R}^n$ is said to converge *linearly* to $\mathbf{x}^\star$ if there exists a constant $c \in (0, 1)$ such that

$$\lim_{n\to\infty} \frac{\|\mathbf{x}(k+1) - \mathbf{x}^\star\|}{\|\mathbf{x}(k) - \mathbf{x}^\star\|} = c,$$

where $\|\cdot\|$ is some norm on $\mathbb{R}^n$.

2.2 Graph Theory

In multi-agent systems in which the communication links (edges) do not change, the exchange of information between the agents is often represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of order n ($n \geq 2$), with $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges. Edge $(v_i, v_j) \in \mathcal{E}$ denotes that there exists a link from agent v_i to agent v_j. If the existence of a link from agent v_i to agent v_j implies that there exists also a link from agent v_j to agent v_j, then the graph is said to be *undirected*; otherwise, it is said to be *directed*.

A directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is said to be *strongly connected* if there exists a path from each agent to every other agent in the network. In a directed graph, the set of in-neighbors $\mathcal{N}_i^-$, often referred to as *in-neighborhood*, is defined as $\mathcal{N}_i^- = \{v_j \in V \,|\, (v_j, v_i) \in \mathcal{E}\}$. Similarly, the *out-neighborhood* $\mathcal{N}_i^+$ is defined as $\mathcal{N}_i^+ = \{v_j \in V \,|\, (v_i, v_j) \in \mathcal{E}\}$. For a given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the set of matrices compatible with $\mathcal{G}$ is define as

$$\mathbb{A}_{\mathcal{G}} = \left\{ M \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|} \,|\; M_{ij} = 0, \quad \forall\, (v_i, v_j) \notin \mathcal{E} \right\}.$$

2.3 Contractions and fixed points

We consider iterative algorithms on the form

$$\mathbf{x}(k+1) = \mathbf{F}\big(\mathbf{x}(k)\big), \; k \in \mathbb{N}_0, \tag{1}$$

where $\mathbf{F}$ is a mapping from a subset $\mathcal{X}$ of $\mathbb{R}^n$ into itself. A vector $\mathbf{x}^\star$ is called a fixed point of $\mathbf{F}$ if $\mathbf{F}(\mathbf{x}^\star) = \mathbf{x}^\star$. A sufficient condition for $\mathbf{x}^\star$ is a fixed point of $\mathbf{F}$ is that $\mathbf{F}$ is continuous at $\mathbf{x}^\star$ and that the sequence $\{\mathbf{x}(k)\}$ converges to $\mathbf{x}^\star$. Therefore, iteration (1) can be conceived as an iterative algorithm for reaching the fixed point. $\mathbf{F}$ is called a *contraction mapping*, if it satisfies

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\| \leq c \, \|\mathbf{x} - \mathbf{y}\| \,, \forall x, y \in \mathcal{X},$$

where $\|\cdot\|$ is some norm on $\mathcal{X}$, and $c \in [0, 1)$.

In the following lemma two main properties that contraction mappings have are presented: *a)* existence and uniqueness of fixed points, and *b)* linear convergence rates.

Lemma 1. (Bertsekas and Tsitsiklis (1997), Chapter 3). If $\mathbf{F} : \mathcal{X} \to \mathcal{X}$ is a contraction mapping and that $\mathcal{X}$ is a closed subset of $\mathbb{R}^n$, then:

- The mapping $\mathbf{F}$ has a unique fixed point $\mathbf{x}^\star \in \mathcal{X}$.
- For every initial condition $\mathbf{x}(0) \in \mathcal{X}$, the sequence $\{\mathbf{x}(k)\}$ generated by $\mathbf{x}(k+1) = F\big(\mathbf{x}(k)\big)$ converges to $\mathbf{x}^\star$ linearly. In particular,

$$\|\mathbf{x}(k) - \mathbf{x}^\star\| \leq c^k \|\mathbf{x}(0) - \mathbf{x}^\star\|.$$

3. PROBLEM STATEMENT

3.1 DLB without delays

We consider a multi-agent system consisting of n agents, each with an initial load (representing, e.g., jobs to be processed) $x_i \geq 0$. Each agent v_i has load processing capacity of $b_i \geq 0$. Within this multi-agent system, which is depicted by a directed and strongly connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, each agent v_i is able to transfer a portion of its load to an out-neighbor v_j; this load transfer is denoted by $W_{ij} > 0$, and we assume it cannot exceed certain threshold $W_{ij}^{UB} > 0$, due to limitation on link (v_i, v_j). The aim of the multi-agent system is to transfer loads distributively in the network in order to satisfy two competing objectives:

(a) minimize the difference among the agents' loads in a least-squares sense;
(b) minimize the effort (load transfer) required for achieving a).

Towards this end, each node aims at selecting W_{ij} for all its out-neighbors v_j. Hence, the load, $\ell_i(W)$, at each node is given by

$$\ell_i(W) = x_i - \sum_{j \in \mathcal{N}_i^+} W_{ij} + \sum_{l \in \mathcal{N}_i^-} W_{li}, \tag{2}$$

which is equal to the original load x_i plus the difference between the incoming and the outgoing ones. As for the *effort*, we define it as

$$\epsilon(W) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i^+} c_{ij} W_{ij}^2, \tag{3}$$

where the coefficients $c_{ij} \geq 0$ represent heterogeneous costs associated to different edges. The described problem is cast as an optimization problem in which the overall objective function is a linear combination of two conflicting convex objectives: the load balancing and minimum effort. It is described in Problem 1.

Problem 1. Let $\boldsymbol{x} \triangleq [x_1 \ x_2 \ \cdots \ x_n]^T \in \mathbb{R}^n_{\geq 0}$ be given such that $\mathbf{1}_n^T \boldsymbol{x} > 0$. Find $W^* \in \mathbb{A}_G$ that solves

$$\begin{aligned} \min_{W \in \mathbb{A}_G} \quad & \frac{\alpha}{2} \sum_{i=1}^{n} \ell_i^2(W) + \frac{1-\alpha}{2} \epsilon(W), \ \alpha \in (0,1) \\ \text{subject to} \quad & 0 \leq \ell_i(W) \leq b_i, \ \forall v_i \in \mathcal{V} \\ & 0 \leq W_{ij} \leq W_{ij}^{UB}, \ \forall (v_i, v_j) \in \mathcal{E}. \end{aligned}$$

For Problem 1, we make the following assumptions.

Assumption 1. A feasible solution to Problem 1 exists.

Assumption 2. While the transfer of loads takes place according to directed graph $\mathcal{G}$, the communication graph of the agents corresponds to the undirected counterpart of graph $\mathcal{G}$.

Assumption 1 is necessary for guaranteeing that there exists a solution to the problem, i.e., the total load do not exceed the total capacity of the system. Assumption 2 states basically that agents are able to exchange information with both their in- and out-neighbors, but they can transfer loads only to their out-neighbors. In Oliva et al. (2022), it is shown that the global optimum to Problem 1 is equivalent to a solution of Problem 2.

Problem 2. Find $\boldsymbol{\ell}^* \in \mathbb{R}^n$, $W^* \in \mathbb{A}_G$, $\boldsymbol{\gamma}^* \in \mathbb{R}^n$ that satisfy

$$\begin{aligned} x_i &= \sum_{j \in \mathcal{N}_i^+} W_{ij}^* - \sum_{l \in \mathcal{N}_i^-} W_{li}^* + \ell_i^* = 0, \ \forall v_i \in \mathcal{V} \\ \ell_i^* &= \min\left\{ \frac{1}{\alpha} \gamma_i^*, b_i \right\}, \ \forall v_i \in \mathcal{V} \\ W_{ij}^* &= \min\left\{ \max\left\{ 0, \frac{\gamma_i^* - \gamma_j^*}{(1-\alpha) c_{ij}} \right\}, W_{ij}^{UB} \right\}, \ \forall (v_i, v_j) \in \mathcal{E}. \end{aligned}$$

This problem, under Assumptions 1 and 2, is solved in a distributed fashion in Oliva et al. (2022), by assuming timely exchange of information between neighboring agents, i.e., in the absence of any communication delays. Specifically, the distributed algorithm for each agent v_i is given by

$$\gamma_i(k+1) = f_i(\boldsymbol{\gamma}(k)), \tag{4}$$

where $\boldsymbol{\gamma}(k) = [\gamma_1(k) \ \gamma_2(k) \ \dots \ \gamma_n(k)]^T$ and the function $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ is given by

$$f_i(\boldsymbol{\gamma}) = \gamma_i + \beta_i \left(x_i - \sum_{j \in \mathcal{N}_i^+} W_{ij}(\boldsymbol{\gamma}) + \sum_{l \in \mathcal{N}_i^-} W_{li}(\boldsymbol{\gamma}) - \ell_i(\gamma_i) \right). \tag{5}$$

where $\beta_i \in \mathbb{R}_{\geq 0}$ is a convergence gain to be defined later,

$$\ell_i(\gamma_i) = \min\left\{ \frac{1}{\alpha} \gamma_i, b_i \right\}, \tag{6a}$$

$$W_{ij}(\boldsymbol{\gamma}) = \min\left\{ \max\left\{ 0, \frac{\gamma_i - \gamma_j}{(1-\alpha) c_{ij}} \right\}, W_{ij}^{UB} \right\}. \tag{6b}$$

Compactly, the dynamics for all agents can be written as

$$\boldsymbol{\gamma}(k+1) = \boldsymbol{f}(\boldsymbol{\gamma}(k)), \tag{7}$$

where $\boldsymbol{f}(\cdot) = [f_1(\cdot), \ \cdots, \ f_n(\cdot)]^T$.

3.2 DLB with delays

We consider a situation in which all agents update their states at each iteration but the information exchanged is delayed, i.e., the communication on the link from agent v_i to agent v_j at time step k may undergo an *a priori unknown* delay $\tau_i^j(k)$, where $\tau_i^j(k)$ is an nonnegative integer, i.e., $\tau_i^j(k) \in \mathbb{N}_0$.

Assumption 3. Delay $\tau_i^j(k)$ satisfies $0 \leq \tau_i^j(k) \leq \bar{\tau}_i^j < \infty$ (i.e., we assume that the communication delays are bounded), where $\bar{\tau}_i^j \in \mathbb{N}_0$ is an upper bound on the delay on the link from agent v_i to agent v_j. The maximum delay is denoted by

$$\bar{\tau} = \max_{(v_i, v_j) \in \mathcal{E}} \bar{\tau}_i^j.$$

Note that $\tau_j^j(k) = 0$, $\forall v_j \in \mathcal{V}, \forall k$, i.e., every agent knows its own value with no delay.

Thus, at time step k, node v_j combines its own value and all values received by its neighbors by that time, i.e., a subset of the values in the set

$$\{\gamma_j(s) \mid s + \tau_i^j(s) = k \ , v_j \in \mathcal{N}_i^- \cup \mathcal{N}_i^+\}.$$

As a result, given that the communication links may experience delays, the distributed algorithm (4) proposed in Oliva et al. (2022) becomes

$$\begin{aligned}\gamma_i(k+1) &= f_i\left(\gamma_1(k-\tau_1^i(k)),\ldots,\gamma_n(k-\tau_n^i(k))\right),\\ \gamma_i(k) &= \varphi_i(k), \quad k\in\{-\bar{\tau},\ldots,-1,0\},\end{aligned} \tag{8}$$

where $\boldsymbol{\varphi}(\cdot) = [\varphi_1(\cdot),\ldots,\varphi_n(\cdot)]^T$ is the vector sequence specifying the initial state of the system. The distributed algorithm now has each node v_i update its information state at time step k to $\gamma_i(k+1)$ by combining its own value $\gamma_i(k)$ and the possibly delayed information received at that time step by its in- and out-neighbors.

System (8) is said to be *positive* if for every non-negative initial condition $\boldsymbol{\varphi}(\cdot) \in \mathbb{R}^n_{\geq 0}$, then $\gamma(k) \geq \mathbf{0}_n$ for all $k \in \mathbb{N}$ (i.e., the corresponding state trajectory is non-negative). Note for ensuring that the state trajectory of (8) remains positive throughout, it is essential that the initial conditions are guaranteed to be nonnegative. Therefore, we set the following assumption.

Assumption 4. The vector describing the initial state of the system, $\boldsymbol{\varphi}(\cdot) = [\varphi_1(\cdot),\ldots,\varphi_n(\cdot)]^T$, has all its entries nonnegative, i.e., $\boldsymbol{\varphi}(\cdot) \in \mathbb{R}^n_{\geq 0}$.

We will show that if (8) is employed in place of (4), the resulting distributed algorithm can still be used to solve Problem 1, despite the presence of arbitrary, bounded delays during the exchange of information. In other words, we establish that (4) is a distributed algorithm which is tolerant to arbitrary bounded delays. Also, a bound on the convergence rate of (8) is derived.

4. MAIN RESULTS

Note that in Oliva et al. (2022) it is shown that $\boldsymbol{f}$ is positive and monotonically non-decreasing; moreover, it has a unique fixed point $\boldsymbol{\gamma}^* \in \mathbb{R}^n$ and is contractive, i.e.,

$$\|\boldsymbol{f}(\boldsymbol{\gamma}) - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty \leq c\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty, \quad 0 < c < 1. \tag{9}$$

Proposition 1. Suppose Assumptions 1, 2, 3, and 4 hold, and let $\beta_i < \overline{\beta}_i$, where

$$\overline{\beta}_i = \frac{1}{\frac{1}{\alpha} + \sum_{j\in\mathcal{N}_i^+} \frac{1}{(1-\alpha)c_{ij}}}. \tag{10}$$

Then, $\boldsymbol{f}(\cdot)$ in (8) is a contraction mapping and for all $\boldsymbol{\gamma}(0) \geq \mathbf{0}_n$ it converges linearly to the unique fixed point $\boldsymbol{\gamma}^*$ (which corresponds to the optimal solution of Problem 1), while it satisfies

$$\|\boldsymbol{f}(\boldsymbol{\gamma}) - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty \leq \bar{c}\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty, \tag{11}$$

where $\bar{c} = c^{\frac{1}{\bar{\tau}+1}}$ and c corresponds to c in (9).

Before proceeding to the proof for the convergence rate of (8) in Proposition 1, we review the following assumption (Assumption 5) and proposition (Proposition 2).

Assumption 5. There exist $\alpha \in [0,1)$ and $A \in [0,1)$ such that, for all $\boldsymbol{\gamma} \in \mathbb{R}^n$ and for all $i \in \{1,\ldots,n\}$ it holds

$$|f_i(\boldsymbol{\gamma}) - \gamma_i^*| \leq \max\left\{\alpha|\gamma_i - \gamma_i^*|, A\max_{j\neq i}|\gamma_j - \gamma_j^*|\right\}$$

Proposition 2. (Bertsekas and Tsitsiklis (1997), p. 461). The sequence of vectors generated by the asynchronous iteration satisfies

$$\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}^*\|_\infty \leq \rho_A^k\|\boldsymbol{\gamma}(0) - \boldsymbol{\gamma}^*\|_\infty$$

where, if Assumption 5 holds then ρ_A is the unique solution of the equation

$$\rho = \max\left\{\alpha, A\rho^{-\bar{\tau}}\right\}. \tag{12}$$

Proof. The fact that the state trajectory of (8) converges to the unique fixed-point under condition (10) is easily deduced by noticing that $\boldsymbol{f}(\cdot)$ is a contractive function (as defined in Feyzmahdavian et al. (2012) and proved in Oliva et al. (2022)), a function which determines max-norm contractions and converges even under total asynchronism (see, e.g., (Bertsekas and Tsitsiklis, 1997, Section 6.3)). Note, however, that in the case of totally asynchronous operation, the delays may become unbounded as k increases, due to different update rates. Nevertheless, even though it is easily deduced that the algorithm works under total asynchronism, no quantification of the reduced convergence rates can be obtained that one would expect with increasing time-delays.

We know already that

$$\|\boldsymbol{f}(\boldsymbol{\gamma}) - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty \leq c\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^*\|^{\mathbf{w}}_\infty \tag{13}$$

for some $c \in (0,1)$. This is equivalent to

$$\frac{|f_i(\boldsymbol{\gamma}) - \boldsymbol{\gamma}_i^\star|}{w_i} \leq c\max_{j\in\{1,\cdots,n\}}\frac{|\gamma_j - \gamma_j^\star|}{w_j}, \; i = 1,\ldots,n. \tag{14}$$

Let $\bar{\tau}_i$ be the maximum communication delay between agent v_i and neighboring agents $v_j \in \mathcal{N}_i^+ \cup \mathcal{N}_i^-$. Then, by Proposition 2, we have that the convergence rate of the asynchronous algorithm (8) is the unique solution of

$$\rho = \max\{c\rho^{-\bar{\tau}_1},\cdots,c\rho^{-\bar{\tau}_n}\}. \tag{15}$$

which emanates from (12). Since $\bar{\tau} = \max\{\bar{\tau}_1,\cdots,\bar{\tau}_n\}$, Eq. (15) can be written as $\rho = c\rho^{-\bar{\tau}}$ and letting $\bar{c} = \rho$ yields $\rho_A = c^{\frac{1}{\bar{\tau}+1}}$. The proof is complete. □

5. SIMULATION RESULTS

In order to numerically validate the theoretical results, let us consider the directed graph $\mathcal{G} = \{\mathcal{V},\mathcal{E}\}$ with $|\mathcal{V}| = 10$ nodes and $|\mathcal{E}| = 24$ edges, reported in Figure 1. In particular, we assume each link has a maximum capacity $W_{ij}^{UB} = 90$ and a costs $c_{ij} = 1$ and we select $\alpha = 0.5$, to represent a case in which the two objectives (i.e., load balancing and minimization of the effort) are equally important. The initial loads $\boldsymbol{x}$ and the nodes' capacities $\boldsymbol{b}$, along with the optimal value $\boldsymbol{\gamma}^*$ of the Lagrange multipliers (obtained for reference via the centralized quadratic programming solver `quadprog` in Matlab) and the corresponding optimal loads $\boldsymbol{\ell}^*$ are as follows:

$$\boldsymbol{x} = \begin{bmatrix}30\\60\\50\\70\\40\\80\\90\\10\\100\\20\end{bmatrix}, \boldsymbol{b} = \begin{bmatrix}50\\80\\80\\50\\80\\50\\80\\80\\50\\50\end{bmatrix}, \boldsymbol{\gamma}^* = \begin{bmatrix}28.6296\\31.2407\\35.0926\\36.0185\\24.0370\\56.8148\\41.8148\\27.6111\\60.0926\\17.0185\end{bmatrix}, \boldsymbol{\ell}^* = \begin{bmatrix}50.0000\\62.4815\\70.1852\\50.0000\\48.0741\\50.0000\\80.0000\\55.2222\\50.0000\\34.0370\end{bmatrix}.$$

Moreover, the optimal values W_{ij}^* for the loads transferred along the edges are reported below (optimal entries being equal to zero are omitted for brevity)

$$\begin{aligned} W_{21}^* &= 5.2222, \quad W_{41}^* = 14.7778, \quad W_{32}^* = 7.7037;\\ W_{93}^* &= 50.0000, \quad W_{74}^* = 11.5926, \quad W_{35}^* = 22.1111;\\ W_{67}^* &= 30.0000, \quad W_{48}^* = 16.8148, \quad W_{78}^* = 28.4074;\\ &W_{5,10}^* = 14.0370. \end{aligned}$$

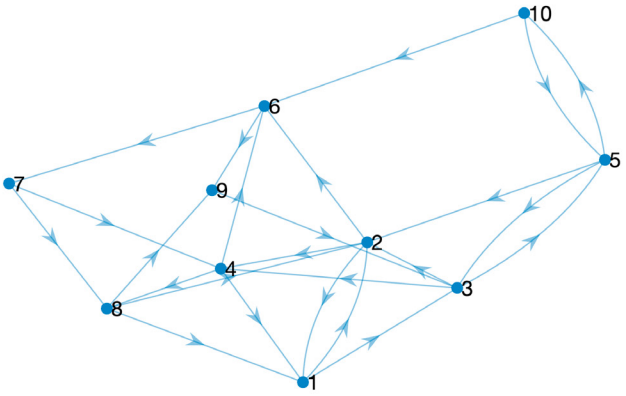

Fig. 1. Sample instance with $|V| = 10$ nodes and $|E| = 24$ edges.

Let us now discuss the choice of the parameters β_i. In particular, we observe that the condition given in Eq. (10) is satisfied if all $\beta_i < 0.1$. Therefore, in this example, we chose to set all $\beta_i = 0.09$. Finally, in this example, we assume that the messages are affected by uniformly random delays in the range $[0, \bar{\tau}]$, where $\bar{\tau} = 5$.

Figures 2 and 3 show with solid lines the temporal evolution of the Lagrange multipliers $\gamma_i(k)$ and the weights $W_{ij}(k)$, respectively, while the corresponding optimal values are shown by gray dashed lines. According to the figures, in spite of the delays, the Lagrange multipliers and the weights converge to the optimal values.

Figure 4 shows the evolution of the error $\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}^*\|_\infty$ between the state vector $\boldsymbol{\gamma}(k)$ computed by the agents via Eq. (8) and the optimal Lagrange multipliers $\boldsymbol{\gamma}^*$, for different choices of the maximum delay (i.e., for $\bar{\tau} = 5$, $\bar{\tau} = 10$, and $\bar{\tau} = 15$) and considering initial conditions chosen uniformly at random in $[0, 1]^n$ (the three curves are shown via solid lines in Figure 4). According to the figure, the error exhibits asymptotic convergence to zero, and the convergence rate is inversely proportional to $\bar{\tau}$. For the sake of comparison, the blue dotted line in Figure 4 reports the evolution of the algorithm in Eq. (7), i.e., the evolution without delays. Based on the latter dynamics, we experimentally estimate an upper bound c on the convergence rate by considering $\boldsymbol{w} = \mathbf{1}_n$ (i.e., by considering the usual infinity norm) and by computing

$$c = \max_{k=0,\dots,\bar{k}} \frac{\|\boldsymbol{\gamma}(k+1) - \boldsymbol{\gamma}^*\|_\infty}{\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}^*\|_\infty},$$

where, to avoid precision errors, we limit analysis to the smallest step $\bar{k}$ at which $\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}^*\|_\infty \leq 10^{-12}$. As a result, we obtain an upper bound $c = 0.9724$. Let us now show that $\rho_A = \sqrt[\bar{\tau}+1]{c}$ is an upper bound on the convergence rate of the case with delays (in particular, we have $\rho_A = 0.9953$ for $\bar{\tau} = 5$, $\rho_A = 0.9975$ for $\bar{\tau} = 10$, and $\rho_A = 0.9983$ for $\bar{\tau} = 15$). To this end, the dashed curves in Figure 4 corresponds to the upper bounds $\sqrt[\bar{\tau}+1]{c^k}\|\boldsymbol{\gamma}(0) - \boldsymbol{\gamma}^*\|_\infty$ for the different choices of $\bar{\tau}$. Interestingly, it can be noted that such curves upper bound the corresponding error in the case with delays, even though the bound is not tight (for instance, the actual convergence rate for $\bar{\tau} = 15$ is also faster than the upper bound for the case $\bar{\tau} = 5$).

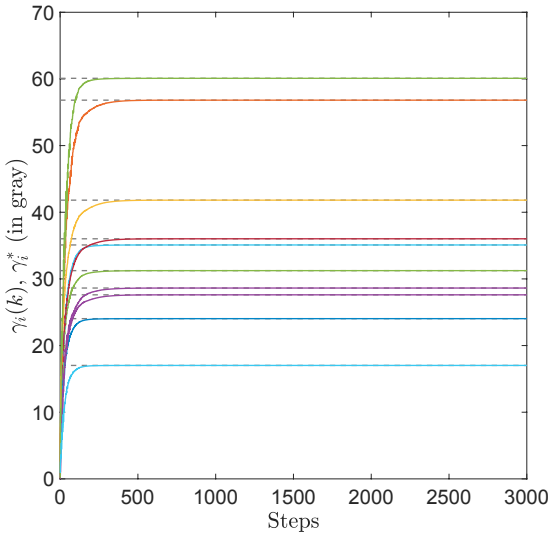

Fig. 2. Evolution of the Lagrangian multipliers $\gamma_i(k)$ (solid lines) and optimal values γ_i^* (gray dashed lines).

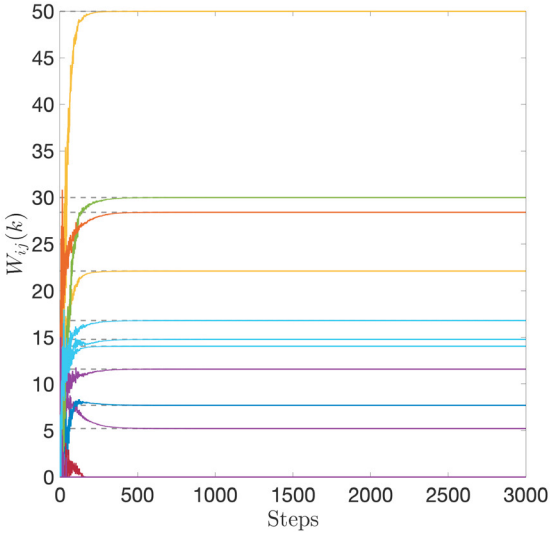

Fig. 3. Evolution of the weights $W_{ij}(k)$ (solid lines) and optimal values W_{ij}^* (gray dashed lines). Only weights with $W_{ij}^* \neq 0$ are reported.

6. CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

In this paper, we considered the problem of distributed load balancing in the presence of communication delays. More specifically, we considered the problem of distributed load balancing problem under network capacity constraints, where the participating agents cooperate with the aim of jointly minimizing both the workload disparity among them as well as the overall workload transfer, which was introduced in Oliva et al. (2022). We modified the distributed algorithm proposed in Oliva et al. (2022) to account for the possibly delayed information received at an agent by its in- and out-neighbors. We show that it is possible for the modified algorithm to converge to the optimal solution even in the presence of arbitrary but bounded delays at the communication links. Additionally,

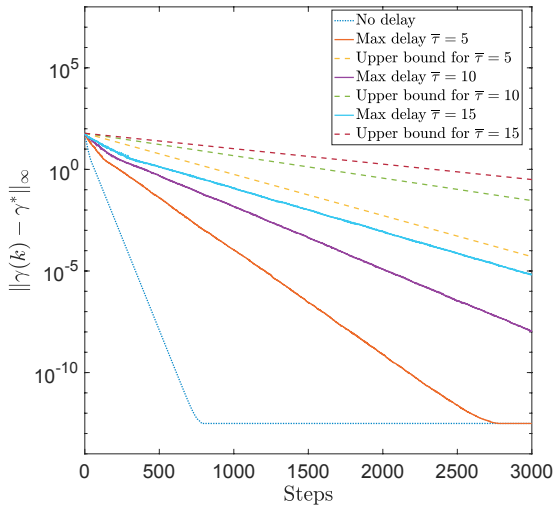

Fig. 4. Evolution of the error $\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}^*\|_\infty$ for the system without delay (dotted line), with delay (solid lines) and upper bounds on the convergence rate in the delayed case (dashed lines).

we derived bounds on the convergence rate of the algorithm. The validity of our results was demonstrated via an illustrative example.

6.2 Future Directions

It is often the case that communication links may disappear (due to, e.g., deep fading in wireless networks) or change (due to, e.g., movement of the agents) and hence the communication topology changes over time. Therefore, it would be interesting to study the convergence of the distributed algorithm considered under switching communication topologies. Moreover, an interesting future research direction could be to relax Assumption 2; to this end, approaches that rely on *ratio consensus* (see Hadjicostis et al. (2018)) could be investigated, e.g., Jiang and Charalambous (2022). Finally, strategies for balancing dynamically generated load are extremely desirable in several practical applications. Therefore, it would be interesting to investigate whether this algorithm or a modified version of it can balance the load dynamically.

REFERENCES

Antonis, K., Garofalakis, J., Mourtos, I., and Spirakis, P. (2004). A hierarchical adaptive distributed algorithm for load balancing. *Journal of Parallel and Distributed Computing*, 64(1), 151–162.

Bertsekas, D.P. and Tsitsiklis, J.N. (1997). *Parallel and distributed computation: numerical methods*. Athena Scientific.

Chow, Y.C. et al. (1979). Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Transactions on Computers*, 100(5), 354–361.

Christ, Q., Dauzère-Pérès, S., and Lepelletier, G. (2019). An iterated min-max procedure for practical workload balancing on non-identical parallel machines in manufacturing systems. *European Journal of Operational Research*.

Feyzmahdavian, H.R., Johansson, M., and Charalambous, T. (2012). Contractive interference functions and rates of convergence of distributed power control laws. *IEEE Transactions on Wireless Communications*, 11(12), 4494–4502.

Hadjicostis, C.N., Domínguez-García, A.D., Charalambous, T., et al. (2018). Distributed averaging and balancing in network systems: with applications to coordination and control. *Foundations and Trends® in Systems and Control*, 5(2-3), 99–292.

Hanada, K., Wada, T., Masubuchi, I., Asai, T., and Fujisaki, Y. (2019). Multi-agent consensus for distributed power dispatch with load balancing. *Asian Journal of Control*.

Jiang, W. and Charalambous, T. (2022). A fast finite-time consensus based gradient method for distributed optimization over digraphs. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 6848–6854. IEEE.

Jiang, Y., Zhou, Y., and Li, Y. (2015). Reliable task allocation with load balancing in multiplex networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 10(1), 3.

Li, C., Cai, Q., and Lou, Y. (2022). Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud. *Future Generation Computer Systems*, 127, 142–159.

Oliva, G., Charalambous, T., Faramndi, L., Setola, R., and Gasparri, A. (2022). Best Effort Workload Disparity Minimization in Multi-Agent Systems with Capacity Constraints. *IEEE Transactions on Automatic Control*, 1–8. doi:10.1109/TAC.2022.3214058.

Ornatelli, A., Tortorelli, A., Giuseppi, A., and Priscoli, F.D. (2021). Hierarchical rl for load balancing and qos management in multi-access networks. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, 886–891. IEEE.

Penmatsa, S. and Chronopoulos, A.T. (2011). Game-theoretic static load balancing for distributed systems. *Journal of Parallel and Distributed Computing*, 71(4), 537–555.

Perkins, J.R. and Kumark, P. (1989). Stable, distributed, real-time scheduling of flexible manufacturing/assembly/diassembly systems. *IEEE Transactions on Automatic Control*, 34(2), 139–148.

Pietrabissa, A., Celsi, L.R., Cimorelli, F., Suraci, V., Priscoli, F.D., Di Giorgio, A., Giuseppi, A., and Monaco, S. (2018). Lyapunov-based design of a distributed wardrop load-balancing algorithm with application to software-defined networking. *IEEE Transactions on Control Systems Technology*, (99), 1–13.

Sohn, I. and Lee, S.H. (2016). Distributed load balancing via message passing for heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 65(11), 9287–9298.

Suraci, V., Celsi, L.R., Giuseppi, A., and Di Giorgio, A. (2017). A distributed wardrop control algorithm for load balancing in smart grids. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, 761–767. IEEE.

Zhang, F., Deng, R., Zhao, X., and Wang, M.M. (2021). Load balancing for distributed intelligent edge computing: A state-based game approach. *IEEE Transactions on Cognitive Communications and Networking*, 7(4), 1066–1077.