Automatica 146 (2022) 110621

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Non-oscillating quantized average consensus over dynamic directed topologies[☆,☆☆]

Apostolos I. Rikos[a,b,*], Christoforos N. Hadjicostis[c], Karl H. Johansson[a,b]

[a] Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[b] Digital Futures, SE-100 44 Stockholm, Sweden
[c] Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

ARTICLE INFO

Article history:
Received 5 March 2021
Received in revised form 16 January 2022
Accepted 29 July 2022
Available online 5 October 2022

Keywords:
Quantized average consensus
Distributed algorithms
Event-triggered
Quantization
Digraphs
Multi-agent systems

ABSTRACT

In this paper we study the distributed average consensus problem in multi-agent systems with dynamically-changing directed communication links that are subject to quantized information flow. We present and analyze a distributed averaging algorithm which operates exclusively with quantized values (i.e., the information stored, processed and exchanged between neighboring agents is subject to deterministic uniform quantization) and relies on event-driven updates (e.g., to reduce energy consumption, communication bandwidth, network congestion, and/or processor usage). We characterize the properties of the proposed distributed algorithm over dynamic directed communication topologies subject to some connectivity conditions and we show that its execution allows each agent to reach, in finite time, a fixed state that is equal (within one quantization level) to the average of the initial states. The main idea of the proposed algorithm is that each agent (i) models its initial state as two quantized fractions which have numerators equal to the agent's initial state and denominators equal to one, and (ii) transmits one fraction randomly while it keeps the other stored. Then, every time an agent receives one or more fractions, it averages their numerators with the numerator of the fraction it stored, and then transmits them to randomly selected out-neighbors. Finally, we provide examples to illustrate the operation, performance, and potential advantages of the proposed algorithm. We compare against various quantized average consensus algorithms and show that our algorithm's convergence speed is among the fastest in the current literature.

© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, there has been a growing interest for control and coordination of networks consisting of multiple agents, like groups of sensors (Xiao, Boyd, & Lall, 2005) or mobile autonomous agents (Olfati-Saber & Murray, 2004). A problem of particular interest in distributed control is the *consensus* problem where the objective is to develop distributed algorithms that can be used by a group of agents in order to reach agreement to a common decision. The agents start with different initial states/information and are allowed to communicate locally via inter-agent information exchange under some constraints on connectivity. Consensus processes play an important role in many problems, such as leader election (Lynch, 1996), motion coordination (Blondel, Hendrickx, Olshevsky, & Tsitsiklis, 2005; Olfati-Saber & Murray, 2004), and clock synchronization (Schenato & Gamba, 2007).

One special case of the consensus problem is the distributed averaging problem, where each agent (initially endowed with a numerical state) can send/receive information to/from other agents in its neighborhood and update its state iteratively, so that eventually, all agents compute the average of the initial states. Average consensus is an important problem and has been studied extensively, primarily in settings where each agent processes and transmits real-valued states with infinite precision (Blondel et al., 2005; Charalambous et al., 2013; Hadjicostis, Domínguez-García, & Charalambous, 2018; Liu, Mou, Morse, Anderson & Yu,

☆ The material in this paper was presented at the 21st IFAC World Congress (IFAC 2020), July 12–17, 2020, Berlin, Germany. This paper was recommended for publication in revised form by Associate Editor Claudio De Persis under the direction of Editor Christos G. Cassandras. An early version of the algorithm in this paper appears in the conference paper (Rikos and Hadjicostis, 2020). The main differences of this paper with Rikos and Hadjicostis (2020) are: (i) the current version of the proposed algorithm operates over a dynamically changing directed communication topology, (ii) the proposed algorithm avoids oscillating behavior regarding the nodes' states while maintaining fast convergence speed (similar to Rikos and Hadjicostis (2020)), (iii) detailed proofs for convergence and for the results on avoiding oscillatory behavior are provided (not given in Rikos and Hadjicostis (2020)).

☆☆ This work was supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council.

* Corresponding author at: Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden.

E-mail addresses: rikos@kth.se (A.I. Rikos), chadjic@ucy.ac.cy (C.N. Hadjicostis), kallej@kth.se (K.H. Johansson).

https://doi.org/10.1016/j.automatica.2022.110621
0005-1098/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

2011; Sundaram & Hadjicostis, 2008). However, most existing average consensus algorithms are only able to guarantee asymptotic convergence, implying that they cannot be readily applied to real-world distributed control and coordination applications. Furthermore, constraints on the bandwidth of communication links and the capacity of physical memories require both communication and computation to be performed assuming finite precision. For these reasons, researchers have studied the case where network links can only allow messages of limited length to be transmitted between agents, effectively extending techniques for average consensus towards the direction of quantized average consensus (Aysal, Coates, & Rabbat, 2007; Cai & Ishii, 2011; Carli, Fagnani, Speranzon, & Zampieri, 2008; Chamie, Liu, & Basar, 2016; Garcia, Cao, Yu, Antsaklis & Casbeer, 2013; Kashyap, Basar, & Srikant, 2007; Lavaei & Murray, 2012). In addition, the demand for more efficient usage of network resources, has led to an increasing interest for novel event-triggered algorithms for distributed control (Liu, Chen, & Yuan, 2012; Nowzari & Cortés, 2016; Seyboth, Dimarogonas, & Johansson, 2013).

Distributed algorithms that achieve quantized average consensus in an event-driven fashion have a wide variety of applications. They can be used as the basis for various encoding schemes, such as quantized privacy protocols for guaranteeing additional levels of security without significantly increasing communication overhead (Rikos, Charalambous, Johansson, & Hadjicostis, 2020; Ruan, Gao, & Wang, 2019). Furthermore, in recent years there has been a tremendous growth in distributed optimization (Grammenos, Charalambous, & Kalyvianaki, 2020; Khatana & Salapaka, 2020; Nedic, Olshevsky, Ozdaglar, & Tsitsiklis, 2008; Rabbat & Nowak, 2005) and machine learning algorithms (Jiang & Agrawal, 2018; Sun, Chen, Giannakis, & Yang, 2019). The distributed operation of these algorithms over directed graphs requires exchange of the agents' states without any error in order to guarantee convergence to a desired solution. However, as the network size becomes larger, e.g., to speed up the training of deep learning algorithms, the communication overhead of each iteration becomes a major bottleneck. Quantization and event-driven communication are effective approaches to tackle this issue, since reduction of communication and processing costs leads to bandwidth and energy efficient algorithms.

The emerging importance of the aforementioned approaches can be further seen in various recent works. In Nylof, Rikos, Gracy, and Johansson (2022), Rikos et al. (2021) and Taheri, Mokhtari, Hassani, and Pedarsani (2020) researchers present distributed optimization algorithms where quantized communication (i) reduces the communication overhead between nodes in the network, (ii) leads to fast finite time convergence, and (iii) facilitates the usage of privacy protocols for guaranteeing additional levels of security. Furthermore, in Elgabli et al. (2021), Shlezinger, Chen, Eldar, Poor and Cui (2020), Sun, Chen, Giannakis, Yang and Yang (2020) and Reisizadeh, Mokhtari, Hassani, Jadbabaie and Pedarsani (2020) researchers present machine learning algorithms that employ quantization strategies to tackle the large communication overhead and reduce communication payload size, while maintaining fast convergence rates and possible privacy preserving guarantees.

1.1. Literature review

In recent years, quite a few *probabilistic* distributed algorithms for averaging under quantized communication, have been proposed. Specifically, the probabilistic quantizer in Aysal et al. (2007) converges to a common state with a random quantization level for the case where the topology forms a directed graph. In Kar and Moura (2010) the authors present a distributed algorithm which adds a dither to the agents' measurements (before the quantization process) and show that the mean square error can be made arbitrarily small. In Benezit, Thiran, and Vetterli (2011) the authors present a distributed algorithm that guarantees that all agents reach consensus to a value on the interval in which the average lies after a finite number of time steps. In Lavaei and Murray (2012) the authors present a quantized gossip algorithm which deals with the distributed averaging problem over a connected weighted graph, and calculate lower and upper bounds on the expected value of the convergence time, which depend on the principal submatrices of the Laplacian matrix of the weighted graph.

The available literature concerning *deterministic* distributed algorithms for averaging under quantized communication comprises less publications. In Li, Fu, Xie, and Zhang (2011), the authors present a distributed averaging algorithm with dynamic encoding and decoding schemes. They show that for a connected undirected dynamic graph, average consensus is achieved asymptotically with as few as one bit of information exchange between each pair of adjacent agents at each time step, and that the convergence rate depends on the number of network nodes, the number of quantization levels and the synchronizability of the network. In Thanou, Kokiopoulou, Pu, and Frossard (2013) the authors present a novel quantization scheme for solving the average consensus problem when sensors exchange quantized state information. The proposed scheme is based on progressive reduction of the range of a uniform quantizer and leads to progressive refinement of the information exchanged by the sensors. In Carli et al. (2008) the authors derive bounds on the rate of convergence to average consensus for a team of mobile agents exchanging information over time-invariant or randomly time-varying communication networks with symmetries. Furthermore, they study the control performance when agents also exchange logarithmically quantized data over static communication topologies with symmetries. In Nedic, Olshevsky, Ozdaglar, and Tsitsiklis (2009) the authors study distributed algorithms for the averaging problem over dynamic topologies, with a focus on tight bounds on the convergence time of a general class of averaging algorithms. They consider algorithms for the case where agents can exchange and store continuous or quantized states, establish a tight convergence rate, and show that these algorithms guarantee convergence to the average of the initial states, within some error that depends on the number of quantization levels.

Recent papers have studied the quantized average consensus problem with the additional constraint that the state of each node is an integer value. In Kashyap et al. (2007) the authors present a probabilistic algorithm which allows every agent to reach quantized consensus almost surely over a static and undirected communication topology, while in Etesami and Basar (2016) and Basar, Etesami, and Olshevsky (2016) the authors analyze and further improve its convergence rate. In Aysal, Coates, and Rabbat (2008) the authors present a deterministic algorithm for calculating the quantized average of the initial values. The algorithm utilizes probabilistic quantization and operates over an undirected connected communication topology. In Frasca, Carli, Fagnani, and Zampieri (2009) the authors present a distributed algorithm in which the exact average is calculated asymptotically. The algorithm operates over a directed graph and each node stores real values but transmits quantized values and refines, if necessary, the quantization step. In Carli, Fagnani, Frasca, and Zampieri (2010) the authors present a gossip algorithm for calculating the average of the initial states. Each node stores real values, transmits quantized values and either converges to the quantized average, or performs oscillations around the average. In Cai and Ishii (2011) a probabilistic algorithm was proposed to solve the quantized consensus problem over static directed graphs for the case where the agents exchange quantized information and store the changes of their states in an additional (also quantized) variable called "surplus". The authors of Chamie et al. (2016) present a deterministic distributed averaging algorithm subject to quantization on the links and show that, depending on initial conditions, the system either converges in finite time to quantized consensus, or the agents (nodes) enter into a periodic behavior with their states oscillating around the average. In Mou, Garcia, and Casbeer (2017), the authors present two distributed algorithms; one for fixed tree graphs with finite time convergence, and one for a dynamic directed graph with exponential convergence. The algorithms proposed in this work calculate the initial average as the ratio of two scaled sums obtained by running in parallel two iterations. In Dibaji, Ishii, and Tempo (2017) the authors present a distributed algorithm in which the nodes reach consensus (but not necessarily to the average). The algorithm operates over a directed graph and quantization is performed in a probabilistic manner. In Rikos and Hadjicostis (2018) and Rikos and Hadjicostis (2021) the authors present two distributed algorithms, one probabilistic and one deterministic, which calculate the exact quantized average of the initial states (i.e., there is no quantization error) in a finite number of time steps, which is explicitly calculated. In Rikos and Hadjicostis (2020) the authors present a distributed randomized algorithm which calculates the quantized average of the initial states with high probability. The algorithm is shown to outperform other algorithms but the states of the nodes exhibit oscillating behavior (between the ceiling and the floor of the real-valued average of the initial states).

1.2. Main contributions

In this paper, we present a novel distributed average consensus algorithm in which processing, storing, and exchange of information between neighboring agents is event-driven and subject to uniform quantization. The main contribution of this paper is threefold.

A. We introduce a novel distributed algorithm that allows all agents to almost surely reach quantized average consensus in finite time under a dynamic directed communication topology (see Algorithm 1 in Section 4).

B. We show that, unlike existing algorithms in the literature, the proposed algorithm allows each agent to calculate, in finite time and with no oscillations, either the ceiling or the floor of the real average of the initial states (see Theorem 1 in Section 4).

C. We present experimental results in which we compare the proposed algorithm against existing schemes. In static networks we observe that its convergence speed significantly outperforms most finite-time distributed algorithms for average consensus under quantized communication (see Section 5).

The main idea behind the proposed algorithm is the following. Initially each node stores two fractions. Each fraction has numerator equal to the node's initial quantized state and denominator equal to one. Then, the node transmits one fraction randomly while it keeps the other stored. Every time a node receives one or more fractions, it averages their numerators with the numerator of the fraction it keeps stored, and then transmits them to randomly selected out-neighbors.

We show that our proposed algorithm converges almost surely in a finite number of time steps. Furthermore, we present a probabilistic upper bound on the number of time steps each node requires to converge to the average of the initial quantized states. Finally, we elaborate on our claim regarding the fast convergence of the proposed algorithm by presenting simulation results over static and dynamic directed communication topologies. In these simulations, we observe that the convergence speed of our presented algorithm is among the fastest finite-time distributed algorithms for average consensus under quantized communication in the current literature.

Most work dealing with quantization has concentrated on the scenario where the agents have real-valued states but can only transmit quantized values through limited rate channels (e.g., Carli et al., 2008; Chamie et al., 2016). By contrast, our setup covers the case where the states are stored in digital memories of finite capacity (as in Cai & Ishii, 2011; Kashyap et al., 2007; Nedic et al., 2009). Specifically, we assume that states are integer-valued (which comprises a class of quantization effects such as uniform quantization) and the control actuation of each node is event-based, which enables more efficient use of available resources. Furthermore, many papers in the literature (e.g., Chamie et al., 2016; Frasca et al., 2009; Rikos & Hadjicostis, 2020), do not suppress small oscillations around the average consensus value. On the contrary, our algorithm allows each node state to stabilize to a specific value rather than perform oscillations around the average consensus value. This characteristic is important for distributed optimization schemes (e.g., Nylof et al., 2022; Rikos et al., 2021), since it facilitates convergence to a specific state rather than leading to oscillations between two different states/decisions.

1.3. Outline

The remainder of this paper is organized as follows. In Section 2, we introduce the notation used throughout the paper, while in Section 3 we formulate the quantized average consensus problem. In Section 4, we present a probabilistic distributed algorithm which operates over a dynamic digraph and allows the agents to reach consensus to the quantized average of the initial states, in finite time, almost surely. We demonstrate its performance with an illustrative example, analyze its operation, and establish its finite time termination. In Section 5, we present simulation results and comparisons against various other quantized average consensus algorithms. We conclude in Section 6 with a brief summary and remarks about future work.

2. Preliminaries

2.1. Notation

The sets of real, rational, integer and natural numbers are denoted by $\mathbb{R}, \mathbb{Q}, \mathbb{Z}$ and $\mathbb{N}$, respectively. The symbol $\mathbb{Z}_+$ denotes the set of nonnegative integers and the symbol $\mathbb{N}_0$ denotes the set of natural numbers that includes zero. For any $a \in \mathbb{R}$, the floor $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a while the ceiling $\lceil a \rceil$ denotes the least integer greater than or equal to a.

The multi-agent system consists of n ($n \geq 2$) agents communicating only with their immediate neighbors at a given time. The communication topology is directed and *dynamic* (i.e., it changes over time). The dynamically changing directed topology can be captured by a sequence of directed graphs (digraphs)[1]. In the remainder of this paper, we will call a sequence of directed graphs as a *dynamic digraph*. In a dynamic digraph we assume that the set of nodes is fixed while the set of edges among them might change at various points in time. Specifically, a *dynamic* digraph is defined as a sequence of digraphs $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$, $k = 0, 1, 2, \ldots$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (representing the agents of the multi-agent system) and $\mathcal{E}[k] \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges at time step k (self-edges excluded). A directed edge from node v_i to node v_j is denoted by $m_{ji} \triangleq (v_j, v_i) \in \mathcal{E}[k]$, and captures the fact that node v_j can receive information from node v_i (but not the

[1] From Rikos and Hadjicostis (2021), a directed graph (digraph) is defined as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (representing the agents) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges (self-edges excluded).

other way around) at time step k. This means that at each time instant k, each node v_j has possibly different sets of in- and out-neighbors, denoted respectively by $\mathcal{N}_j^-[k]$ and $\mathcal{N}_j^+[k]$ and defined as $\mathcal{N}_j^-[k] = \{v_i \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}[k]\}$ and $\mathcal{N}_j^+[k] = \{v_l \in \mathcal{V} \mid (v_l, v_j) \in \mathcal{E}[k]\}$. The cardinality of $\mathcal{N}_j^-[k]$, at time step k, is called the *in-degree* of v_j and is denoted by $\mathcal{D}_j^-[k] = |\mathcal{N}_j^-[k]|$, while the cardinality of $\mathcal{N}_j^+[k]$, at time step k, is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+[k] = |\mathcal{N}_j^+[k]|$. Given a dynamic digraph $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$ for $k = 1, 2, \ldots, m$, where $m \in \mathbb{N}$, its *union graph* is defined as $\mathcal{G}_d^{1,2,\ldots,m} = (\mathcal{V}, \cup_{k=1}^m \mathcal{E}[k])$. A dynamic digraph over $k = 1, 2, \ldots, m$ is said to be *jointly strongly connected*, if its corresponding union graph $\mathcal{G}_d^{1,2,\ldots,m}$ forms a strongly connected digraph (i.e., for each pair of nodes $v_j, v_i \in \mathcal{V}$, $v_j \neq v_i$, there exists a directed *path* from v_i to v_j).

2.2. Agent operation

With respect to quantization of information flow, each node $v_j \in \mathcal{V}$ maintains, at time step k, $5 + 2\mathcal{D}_j^+$ variables, as follows:

(i) The mass variables $y_j[k], z_j[k]$, where $y_j[k] \in \mathbb{Z}$ and $z_j[k] \in \mathbb{N}_0$, are used for processing and calculating the average of the initial states.

(ii) The state variables $y_j^s[k], z_j^s[k], q_j^s[k]$, where $y_j^s[k] \in \mathbb{Z}$, $z_j^s[k] \in \mathbb{N}$ and $q_j^s[k] \in \mathbb{Z}$ (with $q_j^s[k] = \lfloor \frac{y_j^s[k]}{z_j^s[k]} \rfloor$ or $q_j^s[k] = \lceil \frac{y_j^s[k]}{z_j^s[k]} \rceil$), are used for storing the values of the received mass variables and for calculating the state variable q_j^s, which is the variable that becomes equal to the quantized average of the initial states.

(iii) The transmission variables $c_{lj}^y[k]$ and $c_{lj}^z[k]$ for each $v_l \in \mathcal{N}_j^+[k]$, where $c_{lj}^y[k] \in \mathbb{Z}$ and $c_{lj}^z[k] \in \mathbb{N}_0$, are used for transmitting v_j's mass variables towards its out-neighbors.

2.3. Transmission strategy

Under the dynamic communication topology case, each node v_j assigns a nonzero probability $b_{lj}[k]$ to each of its outgoing edges $m_{lj}[k]$ (including a virtual self-edge) at each time step k, where $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$. This probability assignment for all nodes can be captured, at each time step k, by an $n \times n$ column stochastic matrix $\mathcal{B}[k] = [b_{lj}[k]]$. A simple choice would be to set these probabilities to be equal, i.e.,

$$b_{lj}[k] = \begin{cases} \frac{1}{1+\mathcal{D}_j^+[k]}, & \text{if } v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}, \\ 0, & \text{otherwise.} \end{cases}$$

Each nonzero entry $b_{lj}[k]$ of matrix $\mathcal{B}[k]$ represents the probability of node v_j transmitting towards out-neighbor $v_l \in \mathcal{N}_j^+[k]$ through the edge $m_{lj}[k]$ at time step k, or transmitting to itself (i.e., performing no transmission with probability $b_{jj}[k]$). Let us note here that the dynamic nature of the underlying communication topology implies that the matrix $\mathcal{B}[k]$ is not necessarily primitive at each time step k (whereas for a static strongly connected topology, the corresponding $\mathcal{B}$ will necessarily be primitive).

3. Problem formulation

Consider a digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where each node $v_j \in \mathcal{V}$ has an initial quantized state $y_j[0]$ (for simplicity, we take $y_j[0] \in \mathbb{Z}$) and q is the real average of the initial states:

$$q = \frac{\sum_{l=1}^n y_l[0]}{n}. \tag{1}$$

In this paper, we aim to develop distributed algorithms that address the following problem **P1**.

P1. Given a dynamic digraph $\mathcal{G}_d[k]$ which is jointly strongly connected (e.g., for some finite l we have that the union graph $\mathcal{G}_d^{ml,ml+1,ml+2,\ldots,ml+l-1}$ is strongly connected for all $m = 0, 1, 2, \ldots$), design an algorithm which allows the nodes to obtain, after a finite number of steps, a quantized state q^s which is equal to the ceiling or the floor of the actual average q of the initial states in (1). Specifically, we require that there exists k_0 so that for every $v_j \in \mathcal{V}$ we have

$$(q_j^s[k] = \lfloor q \rfloor \text{ for } k \geq k_0) \text{ or } (q_j^s[k] = \lceil q \rceil \text{ for } k \geq k_0). \tag{2}$$

The quantized average q^s is defined as the ceiling $\lceil q \rceil$ or the floor $\lfloor q \rfloor$ of the true average q of the initial states in (1). Let $S \triangleq \mathbf{1}^{\mathrm{T}} y[0]$, where $\mathbf{1} = [1 \ \ldots \ 1]^{\mathrm{T}}$ is the vector of all ones, and let $y[0] = [y_1[0] \ \ldots \ y_n[0]]^{\mathrm{T}}$ be the vector of the quantized initial states. We can write S uniquely as

$$S = nL + R \tag{3}$$

where L and R are both integers and $0 \leq R < n$. Thus, we have that either L or $L+1$ may be viewed as an integer approximation of the average of the initial states $q = S/n$ (which may not be integer in general).

Remark 1. Note here that our definition of quantized average consensus is different than in some literature (Cai & Ishii, 2011; Chamie et al., 2016; Kashyap et al., 2007; Rikos & Hadjicostis, 2020, 2021). More specifically, we require that all agents states converge to a specific integer, either $\lfloor q \rfloor$ or $\lceil q \rceil$ where q satisfies (1). Apart from Cai and Ishii (2011), this is not achieved by existing finite-time algorithms since they either exhibit oscillating behavior of the agent states between the values $\lfloor q \rfloor$ or $\lceil q \rceil$ (Chamie et al., 2016; Kashyap et al., 2007; Rikos & Hadjicostis, 2020), or calculate the average in the form of a quantized fraction (Rikos & Hadjicostis, 2021).

4. Quantized averaging over dynamic digraphs

In this section,we present a distributed algorithm (detailed as Algorithm 1) which addresses problem **(P1)** presented in Section 3. We assume that, at each time step k, the interconnections between components in the multi-component system are captured by a digraph $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$ in which the set of nodes is fixed but the communication links may change.

Assumption 1.

A_1. *At each time step k, each node v_j has knowledge of the set of its out-neighbors $\mathcal{N}_j^+[k]$ and the number of its out-neighbors $\mathcal{D}_j^+[k]$.*

A_2. *Given an infinite sequence of $\mathcal{G}_d[1]$, $\mathcal{G}_d[2]$, …, $\mathcal{G}_d[k]$, …, describing a dynamic digraph there is a finite window length $l \in \mathbb{N}$ and an infinite sequence of time instants t_0, t_1, …, t_m, …, where $t_0 = 0$, such that for any $m \in \mathbb{Z}_+$, we have $0 < t_{m+1} - t_m < l < \infty$ and the union graph $\mathcal{G}_d^{t_m,\ldots,t_{m+1}-1}$, is equal to the nominal digraph $\mathcal{G}_d$ which is assumed to be strongly connected. The diameter of the strongly connected union graph $\mathcal{G}_d^{t_m,\ldots,t_{m+1}-1}$ is denoted as D^{un} and is the longest shortest path between any two nodes $v_j, v_i \in \mathcal{V}$ (note that D^{un} is also the diameter of the nominal digraph $\mathcal{G}_d$).*

$\overline{A_2}$. *Each $\mathcal{G}_d[k]$, $k = 0, 1, 2, \ldots$, of a dynamic digraph takes a value among a finite set of instances, $\{\mathcal{G}_{d_1}, \mathcal{G}_{d_2}, \ldots, \mathcal{G}_{d_M}\}$. The union graph $(\mathcal{V}, \cup_{i=1}^M \mathcal{E}_{d_i})$, is strongly connected and at each time step k one such topology $\mathcal{G}_{d_i} = (\mathcal{V}, \mathcal{E}_{d_i})$ is selected independently in an i.i.d. manner. Specifically, at time step k, we have $G_d[k] = G_{d_\theta}$ for some $\theta \in \{1, 2, \ldots, M\}$ with probability $p_\theta > 0$ where $\sum_{\theta=1}^M p_\theta = 1$.*

Assumption A_1 implies that the transmitting node knows the number of nodes it transmits messages to at each time instant. In an undirected graph setting, this is not difficult and can be done straightforwardly; in a directed graph setting, this is challenging but there are ways in which knowledge of the out-degree might be possible. For example, there can be an acknowledgment signal via a *distress signal* (special tone in a control slot or some separate control channel) sent at higher power than normal so that it is received by transmitters in its vicinity (Bambos, Chen, & Pottie, 2000). Knowledge of the out-degree is also possible if the nodes periodically perform checks to determine the number of their out-neighbors (e.g., by periodically transmitting the distress signals mentioned above).

Assumption A_2 (or $\overline{A_2}$) is sufficient for the existence of at least one directed path between any pair of nodes infinitely often.

Under the above assumptions, during the operation of Algorithm 1, each node v_j is required to calculate the nonzero probabilities $b_{lj}[k]$ for each of its outgoing edges $m_{lj}[k]$ (where $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$) at each time step k. This calculation is due to the dynamic nature of the communication topology $\mathcal{G}_d[k]$. Note that since each transmitting node v_j has instant knowledge of its out-degree, it can set the weights $b_{lj}[k]$ to be equal to $b_{lj}[k] = \frac{1}{1+\mathcal{D}_j^+[k]}$ for $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$. This choice satisfies $\sum_{l=1}^{n} b_{lj}[k] = 1$ for all $v_j \in \mathcal{V}$ which means that the transition matrix $\mathcal{B}[k] = [b_{lj}[k]]$ is column-stochastic at every time step k. Furthermore, unspecified weights in $\mathcal{B}[k]$ are set to zero and correspond to pairs of nodes (v_l, v_j) that are not connected at time step k, i.e., $b_{lj}[k] = 0$ for $v_l \notin \mathcal{N}_j^+[k] \cup \{v_j\}$.

The intuition behind Algorithm 1 is the following. Initially, each node v_j doubles its mass variables (i.e., it sets $y_j[0] := 2y_j[0]$ and $z_j[0] := 2$). At each time step k, each node v_j assigns nonzero probabilities to its outgoing edges. Then, each v_j checks if $z_j[k] > 1$ in which case (i) it updates its state variables to be equal to the mass variables, and (ii) it *splits* $y_j[k]$ into $z_j[k]$ equal integer pieces (with the exception of some pieces whose value might be greater than others by one). It chooses one piece with minimum y-value and transmits it to itself, and it transmits each of the remaining $z_j[k] - 1$ pieces to randomly selected out-neighbors or to itself. Finally, it receives the values $y_i[k]$ and $z_i[k]$ from its in-neighbors, sums them with its stored $y_j[k]$ and $z_j[k]$ values and repeats the operation. Note here that the local state update of each node v_j is equal to the fraction of its mass variables y_j divided by z_j at time step k for which the condition $z_j[k] > 1$ holds in Iteration Step 2.1.

Remark 2. Note that Algorithm 1 solves the quantized average consensus problem (i.e., the state of each node satisfies (2)) in finite time, also for the special case where the network is a static strongly connected directed graph. In that case, the operation of Algorithm 1 resembles the one in Rikos and Hadjicostis (2020). However, the proposed Algorithm 1 also guarantees convergence in finite time and avoids oscillating behavior of the node states. This means that for a static directed graph, Algorithm 1 exhibits the same performance and convergence rate as Rikos and Hadjicostis (2020), while the state of each node stabilizes to be equal either to the ceiling or the floor of the real average q of the initial states in (1).

Algorithm 1 Quantized Average Consensus Over Dynamic Digraphs

Input A dynamic digraph $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$ with $n = |\mathcal{V}|$ nodes and $m[k] = |\mathcal{E}[k]|$ edges, for each $k = 0, 1, 2, \ldots$. Each node $v_j \in \mathcal{V}$ has an initial state $y_j[0] \in \mathbb{Z}$.

Initialization: Each node $v_j \in \mathcal{V}_p$ sets $y_j[0] := 2y_j[0]$, $z_j[0] = 2$.

Iteration: For $k = 0, 1, 2, \ldots$, each node $v_j \in \mathcal{V}_p$ does the following:

1) assigns a nonzero probability $b_{lj}[k]$ to each of its outgoing edges $m_{lj}[k]$, where $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$, as follows

$$b_{lj}[k] = \begin{cases} \frac{1}{1+\mathcal{D}_j^+[k]}, & \text{if } l = j \text{ or } v_l \in \mathcal{N}_j^+[k], \\ 0, & \text{if } l \neq j \text{ and } v_l \notin \mathcal{N}_j^+[k]. \end{cases}$$

2) **if** $z_j[k] > 1$, **then**
   - 2.1) sets $z_j^s[k] = z_j[k]$, $y_j^s[k] = y_j[k]$, $q_j^s[k] = \left\lfloor \frac{y_j^s[k]}{z_j^s[k]} \right\rfloor$;
   - 2.2) sets (i) $mas^y[k] = y_j[k]$, $mas^z[k] = z_j[k]$; (ii) $c_{lj}^y[k] = 0$, $c_{lj}^z[k] = 0$, for every $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$; (iii) $\delta = \lfloor mas^y[k]/mas^z[k] \rfloor$, $mas^{rem}[k] = y_j[k] - \delta \; mas^z[k]$;
   - 2.3) **while** $mas^z[k] > 1$, **then**
     - 2.3*a*) chooses $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$ randomly according to b_{lj};
     - 2.3*b*) sets (i) $c_{lj}^z[k] := c_{lj}^z[k] + 1$, $c_{lj}^y[k] := c_{lj}^y[k] + \delta$; (ii) $mas^z[k] := mas^z[k] - 1$, $mas^y[k] := mas^y[k] - \delta$;
     - 2.3*c*) **if** $mas^{rem}[k] > 0$, **then** sets $c_{lj}^y[k] := c_{lj}^y[k] + 1$, $mas^{rem}[k] := mas^{rem}[k] - 1$.
   - 2.4) sets $c_{jj}^y[k] := c_{jj}^y[k] + mas^y[k]$, $c_{jj}^z[k] := c_{jj}^z[k] + mas^z[k]$;
   - 2.5) **if** $c_{lj}^z[k] > 0$, **then** transmits $c_{lj}^y[k]$, $c_{lj}^z[k]$ to out-neighbor v_l, for every $v_l \in \mathcal{N}_j^+[k]$;

- **else if** $z_j[k] \leq 1$, **then** sets $c_{jj}^y[k] = y[k]$, $c_{jj}^z[k] = z[k]$:

3) receives $c_{ji}^y[k]$, $c_{ji}^z[k]$ from $v_i \in \mathcal{N}_j^-[k]$ and updates $y_j[k+1]$, $z_j[k+1]$ as

$$y_j[k+1] = c_{jj}^y[k] + \sum_{v_i \in \mathcal{N}_j^-[k]} w_{ji}[k] \; c_{ji}^y[k], \tag{4}$$

$$z_j[k+1] = c_{jj}^z[k] + \sum_{v_i \in \mathcal{N}_j^-[k]} w_{ji}[k] \; c_{ji}^z[k], \tag{5}$$

where $w_{ji}[k] = 1$ if node v_j receives $c_{ji}^y[k]$, $c_{ji}^z[k]$ from $v_i \in \mathcal{N}_j^-[k]$ at iteration k (otherwise $w_{ji}[k] = 0$).

Output: (2) holds for every $v_j \in \mathcal{V}$.

Example 1. Consider the dynamic strongly connected digraph $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$ in Fig. 1 (borrowed from Rikos & Hadjicostis, 2018), where there are four nodes with initial quantized states $y_1[0] = 5$, $y_2[0] = 3$, $y_3[0] = 7$, and $y_4[0] = 2$, respectively. There are three different instances $\mathcal{G}_{d_1}$, $\mathcal{G}_{d_2}$, $\mathcal{G}_{d_3}$ of the dynamic digraph $\mathcal{G}_d[k]$ as presented at the bottom of Fig. 1. The nominal digraph (i.e., the union of $\mathcal{G}_{d_1}$, $\mathcal{G}_{d_2}$, $\mathcal{G}_{d_3}$) has $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{E} = \{m_{21}, m_{31}, m_{42}, m_{13}, m_{23}, m_{34}\}$, and is shown at the top of Fig. 1. Notice that the nominal digraph is strongly connected. The actual average q of the initial states of the nodes, is equal to $q = 4.25$, which means that the quantized state q^s is equal to $q^s = 4$ or $q^s = 5$ (see Section 3).

Each node $v_j \in \mathcal{V}$ follows the Initialization in Algorithm 1. This means that nodes v_1, v_2, v_3, v_4, set $y_1[0] = 10$, $z_1[0] = 2$, $y_2[0] = 6$, $z_2[0] = 2$, $y_3[0] = 14$, $z_3[0] = 2$, and $y_4[0] = 4$, $z_4[0] = 2$, respectively.

For the execution of Algorithm 1, at time step $k = 0$, let us assume that the dynamic digraph is equal to (a) at the bottom of Fig. 1. Each node v_j assigns to each of its outgoing edges $v_l \in \mathcal{N}_j^+[0] \cup \{v_j\}$ a nonzero probability value b_{lj} equal to $b_{lj} = \frac{1}{1+\mathcal{D}_j^+[0]}$.

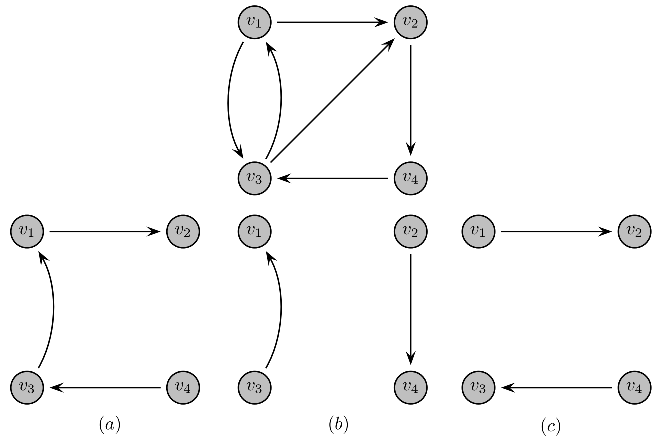

Fig. 1. Example of a dynamic digraph for quantized averaging: the nominal strongly connected digraph is shown at the top figure, and the 3 different instances of the dynamic digraph are shown at the bottom figure.

The assigned values can be seen in the following matrix

$$\mathcal{B}[0] = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}.$$

Then, each node v_j calculates its state variables $y_j^s[0]$, $z_j^s[0]$ and $q_j^s[0]$ as shown in Table 1. Subsequently, every node v_j calculates its transmission variables $c_{lj}^y[0]$ and $c_{lj}^z[0]$ for every $v_l \in \mathcal{N}_j^+[0] \cup \{v_j\}$. Specifically, each node v_j splits $y_j[0]$ in 2 equal pieces (since $z_j[0] = 2$), and keeps one piece (which has the minimum value) for itself and transmits the other piece to a randomly chosen out-neighbor or itself according to the matrix $\mathcal{B}[0]$. For this reason, in the analysis below we have $c_{jj}^z[0] > 0$ for every node v_j. Suppose that, following the random choices, nodes v_1, v_2, v_3, v_4 set

$$\begin{aligned} v_1 :\quad & c_{11}^y[0] = 5,\ c_{21}^y[0] = 5,\ c_{31}^y[0] = 0, \\ & c_{11}^z[0] = 1,\ c_{21}^z[0] = 1,\ c_{31}^z[0] = 0, \\ v_2 :\quad & c_{22}^y[0] = 6,\ c_{42}^y[0] = 0, \\ & c_{22}^z[0] = 2,\ c_{42}^z[0] = 0, \\ v_3 :\quad & c_{13}^y[0] = 7,\ c_{23}^y[0] = 0,\ c_{33}^y[0] = 7, \\ & c_{13}^z[0] = 1,\ c_{23}^z[0] = 0,\ c_{33}^z[0] = 1, \\ v_4 :\quad & c_{34}^y[0] = 2,\ c_{44}^y[0] = 2, \\ & c_{34}^z[0] = 1,\ c_{44}^z[0] = 1. \end{aligned}$$

We have that nodes v_1, v_3 and v_4 perform transmissions to nodes v_2, v_1 and v_3, respectively, whereas node v_2, transmits to itself. Then, each node v_j receives from its in-neighbors $v_i \in \mathcal{N}_j^-[0] \cup \{v_j\}$ the transmission variables $c_{ji}^y[0]$ and $c_{ji}^z[0]$ and, at time step $k = 1$, it calculates its state variables $y_j^s[1]$, $z_j^s[1]$ and $q_j^s[1]$. The mass and state variables are shown in Table 1 for $k = 1$.

Assume that, at time step $k = 1$, the dynamic digraph is equal to (b) at the bottom of Fig. 1. We have that each node v_j assigns to each of its outgoing edges $v_l \in \mathcal{N}_j^+[1] \cup \{v_j\}$ a nonzero probability value b_{lj} equal to $b_{lj} = \frac{1}{1+\mathcal{D}_j^+[1]}$. Also, it calculates and transmits its transmission variables $c_{lj}^y[1]$ and $c_{lj}^z[1]$ for every $v_l \in \mathcal{N}_j^+[1] \cup \{v_j\}$.

In Fig. 2, following a random choice among one of the topologies in Fig. 1 at each time step, we plot the resulting state variable $q_j^s[k]$ of each node $v_j \in \mathcal{V}$, from which it can be seen that for $k \geq 9$ we have

$$q_j^s[k] = \lfloor q \rfloor \quad \text{or} \quad q_j^s[k] = \lceil q \rceil,$$

Table 1
Mass and State Variables for Fig. 1.

Nodes v_j	Mass and State Variables for $k = 0$ $y_j[0]$	$z_j[0]$	$y_j^s[0]$	$z_j^s[0]$	$q_j^s[0]$
v_1	10	2	10	2	5
v_2	6	2	6	2	3
v_3	14	2	14	2	7
v_4	4	2	4	2	2
Nodes v_j	Mass and State Variables for $k = 1$ $y_j[1]$	$z_j[1]$	$y_j^s[1]$	$z_j^s[1]$	$q_j^s[1]$
v_1	12	2	12	2	6
v_2	11	3	11	3	3
v_3	9	2	9	2	4
v_4	2	1	2	1	2

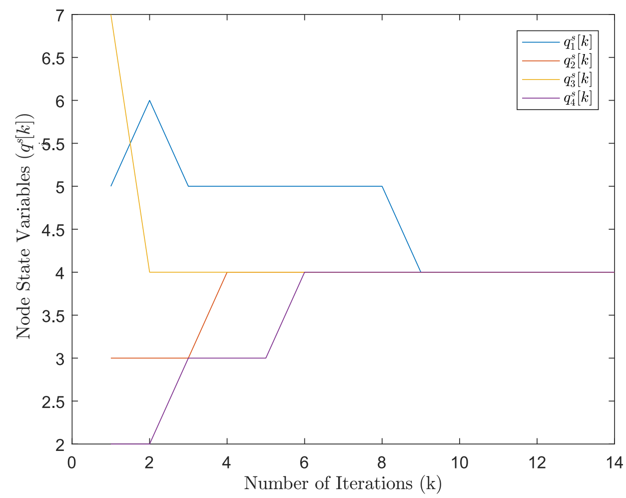

Fig. 2. Node state variables plotted against the number of iterations for Algorithm 1 for the dynamic digraph shown in Fig. 1.

for every $v_j \in \mathcal{V}$. This means that every node v_j obtains, after a finite number of iterations, a quantized state q_j^s, which is equal either to the ceiling or to the floor of the real average q of the initial states of the nodes.

4.1. Convergence of Algorithm 1

We now show that, during the operation of Algorithm 1, each agent v_j reaches a consensus state which is equal either to the ceiling or the floor of the actual average q of the initial states of the nodes (i.e., we address problem **P1** presented in Section 3). We analyze the operation of Algorithm 1 considering the set of assumptions A_1, A_2 (the set of assumptions $A_1, \overline{A_2}$ can be proven similarly). We first consider Lemma 1, which is necessary for our subsequent development. The intuition behind the proof of Lemma 1 is identical to Lemma 1 in Rikos and Hadjicostis (2020) and is omitted due to space considerations.

Lemma 1. *Consider a sequence of digraphs $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$, $k = 0, 1, 2, \ldots$, with $n = |\mathcal{V}|$ nodes, $m[k] = |\mathcal{E}[k]|$ edges, so that assumptions A_1, A_2 hold for $\mathcal{G}_d[k]$ over all k. At each time step k, suppose that each node v_j assigns a nonzero probability $b_{lj}[k]$ to each of its outgoing edges $m_{lj}[k]$, where $v_l \in \mathcal{N}_j^+[k] \cup \{v_j\}$, as follows*

$$b_{lj} = \begin{cases} \frac{1}{1+\mathcal{D}_j^+[k]}, & \text{if } l = j \text{ or } v_l \in \mathcal{N}_j^+[k], \\ 0, & \text{if } l \neq j \text{ and } v_l \notin \mathcal{N}_j^+[k]. \end{cases}$$

At time step $k = 0$, node v_j holds a "token" while the other nodes $v_l \in \mathcal{V} - \{v_j\}$ do not. At each time step k, each node v_j transmits the *"token" (if it has the token, otherwise it performs no transmission) according to the nonzero probability $b_{lj}[k]$ it assigned to its outgoing edges $m_{lj}[k]$. The probability $P_{DT_i}^{D^{un}}$ that the token is at node v_i after lD^{un} time steps satisfies*

$$P_{DT_i}^{lD^{un}} \geq (1+\mathcal{D}_{max}^{+})^{-(lD^{un})} > 0,$$

where l is the time window defined in Assumption A_2 (for which the union graph $\mathcal{G}_d^{t_m,\dots,t_{m+1}-1}$ is equal to the nominal digraph $\mathcal{G}_d$ which is strongly connected), and $\mathcal{D}_{max}^{+}$ is the maximum out-degree of every node in the nominal digraph $\mathcal{G}_d$.

We now present the following theorem in which we calculate the number of time steps required by Algorithm 1 in order to guarantee convergence according to a given probability p_0.

Theorem 1. *Consider a sequence of digraphs $\mathcal{G}_d[k] = (\mathcal{V}, \mathcal{E}[k])$, $k = 0, 1, 2, \dots$, with $n = |\mathcal{V}|$ nodes, $m[k] = |\mathcal{E}[k]|$ edges so that assumptions A_1, A_2 hold for $\mathcal{G}_d[k]$ over all k. Every node $v_j \in \mathcal{V}$ has the variables $z_j[0] = 1$ and $y_j[0] \in \mathbb{Z}$ at time step $k = 0$, and it follows the Initialization and Iteration steps as described in Algorithm 1. For any probability p_0, where $0 < p_0 < 1$, there exists $k_0 \in \mathbb{Z}_+$ (or more precisely $k_0(p_0)$), so that with probability at least p_0 we have*

$$(q_j^s[k] = \lfloor q \rfloor \;\; \text{for} \;\; k \geq k_0) \;\; \text{or} \;\; (q_j^s[k] = \lceil q \rceil \;\; \text{for} \;\; k \geq k_0),$$

for every $v_j \in \mathcal{V}$, where q fulfills (1). The number of required time steps for convergence k_0 depends on the network size and structure, and the nodes initial states, and is equal to $(y^{init} + n)\tau l D^{un}$, where

$$y^{init} = \sum_{\{v_j \in \mathcal{V}: y_j[0] > \lceil q \rceil\}} (y_j[0] - \lceil q \rceil) + \sum_{\{v_j \in \mathcal{V}: y_j[0] < \lfloor q \rfloor\}} (\lfloor q \rfloor - y_j[0]), \tag{6}$$

and τ, l, D^{un} are parameters of the network structure.

Proof. In this proof we compute a bound on the required time steps as a function of the probability of achieving the consensus state. The calculated bound on the required time steps and the probability of achieving the consensus state depend on the network structure and the initial state of every node.

The operation of Algorithm 1 can be interpreted as the "random walk" of n "tokens" in a *dynamic* (inhomogeneous) Markov chain (i.e., interconnections change over time) with $n = |\mathcal{V}|$ states. Each node v_j at time step $k = 0$ holds two "tokens", T_j^{ins} (which is stationary) and T_j^{out} (which performs a random walk), and they each contain a pair of values $y_j^{ins}[k]$, $z_j^{ins}[k]$, and $y_j^{out}[k]$, $z_j^{out}[k]$, respectively, for which it holds that $y_j^{ins}[0] = y_j^{out}[0] = y_j[0] \in \mathbb{Z}$ and $z_j^{ins}[0] = z_j^{out}[0] = z[0] = 1$. At each time step k, each node v_j keeps the token T_j^{ins} (i.e., it never transmits it) while it transmits the token T_j^{out}, according to the nonzero probability $b_{lj}[k]$ it assigned to its outgoing edges $m_{lj}[k]$ during time step k. If v_j receives one or more tokens T_i^{out} from its in-neighbors the values $y_i^{out}[k]$ and $y_j^{ins}[k]$ become equal (or with maximum difference equal to 1); then v_j transmits each received token T_i^{out} to a randomly selected out-neighbor according to the nonzero probability $b_{lj}[k]$. Note here that during the operation of Algorithm 1 we have

$$\sum_{j=1}^{n} y_j^{out}[k] + \sum_{j=1}^{n} y_j^{ins}[k] = 2 \sum_{j=1}^{n} y_j[0], \;\; \forall k \in \mathbb{Z}_+, \tag{7}$$

(i.e., the sum of the $y_j[k]$ values of the tokens at any given k is equal to twice the initial sum).

Let us now define

$$Y[k] = Y_1[k] + Y_2[k], \tag{8}$$

where

$$Y_1[k] = \sum_{\{v_j \in \mathcal{V}: \lceil y_j[k]/z_j[k] \rceil > \lceil q \rceil\}} (\lceil y_j[k]/z_j[k] \rceil - \lceil q \rceil), \tag{9}$$

and

$$Y_2[k] = \sum_{\{v_j \in \mathcal{V}: \lfloor y_j[k]/z_j[k] \rfloor < \lfloor q \rfloor\}} (\lfloor q \rfloor - \lfloor y_j[k]/z_j[k] \rfloor), \tag{10}$$

where q satisfies (1). We have that $Y_1[k]$, $Y_2[k]$ denote the sum of the differences between the values $y[k]$ and $\lceil q \rceil$ of the tokens that have a y value higher than the ceiling of the real average $\lceil q \rceil$, and the sum of the differences between the values $y[k]$ and $\lfloor q \rfloor$ of the tokens that have y value less than the floor of the real average $\lfloor q \rfloor$, respectively. From Iteration Steps 2.3 and 2.4, we have that if two (or more) "tokens" T_i^{out}, T_l^{out} (where $v_i, v_l \in \mathcal{V}$) meet at the same node v_j with token T_j^{ins} during time step k, then their values $y[k]$ become equal (or with maximum difference equal to 1). For the scenario $\lceil q \rceil > \lfloor q \rfloor$, we have at time step k (note that similar arguments hold also for $\lceil q \rceil = \lfloor q \rfloor$):

Case (i): If $Y_1[k] > 0$ and a token which has $y[k] > \lceil q \rceil$ meets with a token that has $y[k] \leq \lfloor q \rfloor$ then we have $Y_1[k+1] \leq Y_1[k] - 1$.

Case (ii): If $Y_2[k] > 0$ and a token which has $y[k] < \lfloor q \rfloor$ meets with a token that has $y[k] \geq \lceil q \rceil$ then we have $Y_2[k+1] \leq Y_2[k] - 1$.

Case (iii): If $Y_1[k] > 0$ and $Y_2[k] > 0$ and a token which has $y[k] > \lceil q \rceil$ meets with a token that has $y[k] < \lfloor q \rfloor$ then we have $Y_1[k+1] \leq Y_1[k] - 1$ and $Y_2[k+1] \leq Y_2[k] - 1$.

Note that for the scenario $\lceil q \rceil = \lfloor q \rfloor$ we have that only Case (iii) above holds. Case (i) and Case (ii) do not hold since the difference between the values $y[k]$ of the tokens that meet might be equal to unity, which means that the values of $Y_1[k]$ and $Y_2[k]$ will not decrease.

Clearly, we have

$$0 \leq Y[k+1] \leq Y[k] \leq y^{init},$$

for all time steps k, where y^{init} fulfills (6) (i.e., y^{init} is the total initial state error). This means that if cases (i), (ii), (iii) hold y^{init} times, the value of Y becomes equal to zero (where Y is defined in (8)). As a result, for every token the values y become equal or have difference equal to one (recall that, during the operation of Algorithm 1, we also have that (7) holds for every k).

In this proof, we consider and analyze the probability that a specific token, with value y_i^{out}, visits a specific node v_j, with token value y_j^{ins}, in the network after a finite number of time steps and obtains equal values (or with maximum difference between them equal to 1) with the token y_j^{ins}, where for tokens T_i^{out} and T_j^{ins} it holds (i) $y_\lambda^{out} \geq \lceil q \rceil$, $y_i^{ins} < \lfloor q \rfloor$, or (ii) $y_\lambda^{out} > \lceil q \rceil$, $y_i^{ins} \leq \lfloor q \rfloor$, or (iii) $y_\lambda^{out} < \lfloor q \rfloor$, $y_i^{ins} \geq \lceil q \rceil$, or (iv) $y_\lambda^{out} \leq \lfloor q \rfloor$, $y_i^{ins} > \lceil q \rceil$. We show that (i) $\exists k_0' \in \mathbb{Z}_+$ for which with probability at least p_0, it holds that $Y_1[k] = 0$ and $Y_2[k] = 0$ for every $k \geq k_0'$, and (ii) $\exists k_0 \in \mathbb{Z}_+$ for which (2) holds with probability at least p_0, for every $k \geq k_0$. This means that after a finite number of time steps k_0 the value $y[k]$ of every token is equal either to $\lfloor q \rfloor$ or to $\lceil q \rceil$, and for the state variable $q_j^s[k]$ of every node v_j we have $q_j^s[k] = \lfloor q \rfloor$ or $q_j^s[k] = \lceil q \rceil$, respectively.

Let us consider tokens T_λ^{out} and T_i^{ins} for which it holds $|y_\lambda^{out} - y_i^{ins}| > 1$. During the operation of Algorithm 1, n "tokens" perform *independent* random walks over a dynamic digraph $\mathcal{G}_d[k]$. Since $b_{lj}[k] \geq (1+\mathcal{D}_{max}^{+})^{-1}$ (where $\mathcal{D}_{max}^{+}$ is defined in Lemma 1) and assumptions A_1, A_2 hold for $\mathcal{G}_d[k]$ during all k, we have that the probability $P_{DT^{out}}^{lD^{un}}$ that "the specific token T_λ^{out} is at node v_i after lD^{un} time steps" is

$$P_{DT^{out}}^{lD^{un}} \geq (1+\mathcal{D}_{max}^{+})^{-(lD^{un})} > 0. \tag{11}$$

This is mainly due to the fact that every l time steps, each edge is active for at least one time step. Since the nominal digraph $\mathcal{G}_d$ is strongly connected, it has a path of length at most D^{un} from each node v_ℓ to each node v_i. Thus, at the first l steps, we can select the first edge in this path (at the instant when it is active) and use self loops at the remaining instants; during the next l time steps, we can select the second edge on this path and use self loops at the remaining instants; and so forth.

From (11) we have that the probability $P^{lD^{un}}_{N\_DT^{out}}$ that "the specific token T^{out}_λ is not at node v_i after lD^{un} time steps" is

$$P^{lD^{un}}_{N\_DT^{out}} \leq 1 - (1 + \mathcal{D}^+_{max})^{-(lD^{un})}. \tag{12}$$

By extending this analysis, we choose ε (where $0 < \varepsilon < 1$) for which it holds that

$$\varepsilon \leq 1 - 2^{\frac{\log_2 p_0}{y^{init}+n}}. \tag{13}$$

We can state that for ε which fulfills (13) and after τlD^{un} time steps where

$$\tau \geq \left\lceil \frac{\log \varepsilon}{\log(1 - (1 + \mathcal{D}^+_{max})^{-(lD^{un})})} \right\rceil, \tag{14}$$

the probability $P^{\tau}_{N\_DT^{out}}$ that "the specific token T^{out}_λ has not visited node v_i after τlD^{un} time steps" is

$$P^{\tau}_{N\_DT^{out}} \leq [P^{lD^{un}}_{N\_T^{out}}]^{\tau} \leq \varepsilon. \tag{15}$$

This means that after τlD^{un} time steps, where τ fulfills (14), the probability that "the specific token T^{out}_λ has visited node v_i after τlD^{un} time steps" is equal to $1 - \varepsilon$.

As a result, after τlD^{un} time steps, where τ fulfills (14), we have that if $Y_1[k] > 0$ and/or $Y_2[k] > 0$ at time step k, then it holds that $Y_1[k+\tau lD^{un}] \leq Y_1[k]-1$ and/or $Y_2[k+\tau lD^{un}] \leq Y_2[k]-1$ with probability $1 - \varepsilon$. By extending this analysis, we have that for $k \geq y^{init}\tau lD^{un}$ time steps, where y^{init} is given by (6), we have $Y[k] = 0$ with probability $(1-\varepsilon)^{y^{init}}$. Since ε fulfills (13), we have that it holds $(1 - \varepsilon)^{y^{init}} \geq p_0$. Therefore, for $k \geq y^{init}\tau lD^{un}$, we have that the value $y[k]$ of every token is equal to either $\lfloor q \rfloor$ or $\lceil q \rceil$ with probability at least p_0. Since (7) holds, for $k \geq y^{init}\tau lD^{un}$ we have

$$\lfloor y^{out}_j[k]/z^{out}_j[k] \rfloor = \lfloor q \rfloor \ \text{ or } \ \lceil y^{out}_j[k]/z^{out}_j[k] \rceil = \lceil q \rceil, \tag{16}$$

and

$$\lfloor y^{ins}_j[k]/z^{ins}_j[k] \rfloor = \lfloor q \rfloor \ \text{ or } \ \lceil y^{ins}_j[k]/z^{ins}_j[k] \rceil = \lceil q \rceil, \tag{17}$$

for every $v_j \in \mathcal{V}$ with probability at least p_0. Furthermore, we have that for $k \geq y^{init}\tau lD^{un}$ it holds

$$\begin{aligned} &|\{T^{ins}_j, v_j \in \mathcal{V} | y^{ins}_j[k] = \lfloor q \rfloor\}| + \\ &|\{T^{out}_j, v_j \in \mathcal{V} | y^{out}_j[k] = \lfloor q \rfloor\}| = 2n - 2R \end{aligned} \tag{18}$$

with probability at least p_0, where $|\{T^{ins}_j, v_j \in \mathcal{V} | y^{ins}_j[k] = \lfloor q \rfloor\}|$ is the cardinality of the set of tokens T^{ins}_j which have y^{ins}_j value equal to $\lfloor q \rfloor$, $|\{T^{out}_j, v_j \in \mathcal{V} | y^{out}_j[k] = \lfloor q \rfloor\}|$ is the cardinality of the set of tokens T^{out}_j which have y^{out}_j value equal to $\lfloor q \rfloor$ and R is defined in (3). This means that the number of tokens with y_j value equal to $\lfloor q \rfloor$ is $2n - 2R$.

Let us consider now the following two cases

(1) $2n - 2R \geq n$ (or $R < n/2$),
(2) $2n - 2R < n$ (or $R > n/2$),

where n is the number of nodes and R is defined in (3).

For the first case, we have that the number of tokens which have value equal to $\lfloor q \rfloor$ is greater than (or equal to) the number of nodes. This means that by executing Algorithm 1 for an additional number of $n\tau lD^{un}$ time steps, where τ fulfills (14), we have that every node will receive at least one token with value $\lfloor q \rfloor$ with probability $(1-\varepsilon)^n$, where ε fulfills (13). [The reason is that during the first τlD^{un} steps, one of the tokens with value $\lfloor q \rfloor$ will reach node v_1 with probability $1-\varepsilon$; during the second τlD^{un} steps, one of the tokens with value $\lfloor q \rfloor$ will reach node v_2 with probability $1 - \varepsilon$, and so on. During the last τlD^{un} steps, one of the tokens with value $\lfloor q \rfloor$ will reach node v_n with probability $1 - \varepsilon$.] From Iteration Steps 2.3 and 2.4 of Algorithm 1 we have that if node v_j receives a token with value $\lfloor q \rfloor$, then the value of its y^{ins}_j token becomes equal to $\lfloor q \rfloor$ which means that also the value of its state variable q^s_j becomes equal to $\lfloor q \rfloor$. As a result, since $2n - 2R \geq n$, for $k \geq (y^{init} + n)\tau lD^{un}$, where y^{init} fulfills (6) and τ fulfills (14), we have

$$y^{ins}_j[k] = \lfloor q \rfloor, \ \text{ for every } v_j \in \mathcal{V},$$

and

$$q^s_j[k] = \lfloor q \rfloor, \ \text{ for every } v_j \in \mathcal{V},$$

with probability $(1 - \varepsilon)^{(y^{init}+n)}$. Since ε fulfills (13), we have that it holds $(1 - \varepsilon)^{(y^{init}+n)} \geq p_0$.

For the second case, the number of tokens which have value equal to $\lfloor q \rfloor$ is less than the number of nodes. Identically to the first case, by executing Algorithm 1 for an additional number of $n\tau lD^{un}$ time steps, where τ fulfills (14), for $k \geq (y^{init}+n)\tau lD^{un}$ we have with probability $(1 - \varepsilon)^{(y^{init}+n)}$ (where $(1 - \varepsilon)^{(y^{init}+n)} \geq p_0$) that

$$y^{ins}_j[k] = \lfloor q \rfloor \text{ and } q^s_j[k] = \lfloor q \rfloor, \ \text{ for every } v_j \in \mathcal{V}',$$

where $\mathcal{V}' \subset \mathcal{V}$ and $|\mathcal{V}'| = 2n - 2R$, and

$$y^{ins}_j[k] = \lceil q \rceil \text{ and } q^s_j[k] = \lceil q \rceil, \ \text{ for every } v_j \in \mathcal{V}'',$$

where $\mathcal{V}'' \subset \mathcal{V}$ and $|\mathcal{V}''| = 2R - n$.

As a result, during the operation of Algorithm 1, for $k \geq (y^{init} + n)\tau lD^{un}$ we have

$$q^s_j[k] = \lfloor q \rfloor \ \text{ or } \ q^s_j[k] = \lceil q \rceil,$$

for every $v_j \in \mathcal{V}$, with probability at least p_0, (i.e., since ε fulfills (13), we have that $(1 - \varepsilon)^{(y^{init}+n)} \geq p_0$). □

Remark 3. Note that during the operation of Algorithm 1 if we adopt the set of assumptions $A_1, \overline{A_2}$ for $\mathcal{G}_d[k]$ during every k, then (11) becomes

$$P^{lD^{un}}_{DT_i} \geq [p_{\theta_{min}}(1 + \mathcal{D}^+_{max})]^{-(lD^{un})} > 0, \tag{19}$$

where $p_{\theta_{min}} = \min_{\theta \in \{1,2,\ldots,M\}} p_\theta > 0$. Since each digraph G_{d_θ}, for $\theta \in \{1, 2, \ldots, M\}$, is selected in an i.i.d. manner with probability $p_\theta \geq p_{\theta_{min}}$ and the union graph is strongly connected, we can first select a topology that includes the first edge on the path from node v_ℓ to node v_i (at least one such topology exists), then select a topology that includes the second edge on the path from node v_ℓ to node v_i (at least one such topology exists), and so forth (with self loops included if necessary). Then, (11) is replaced by (19) and the structure of the proof remains the same.

It is important to note here that Algorithm 1 converges in finite time in the presence of a dynamic digraph. Compared to Rikos and Hadjicostis (2020), the main difference is an increase on the required number of time steps for convergence (which will be shown explicitly in the next section). However, in practical applications, there is also a possible increase in processor usage at each node (due to the calculation of the nonzero probabilities $b_{lj}[k]$ for each of its outgoing edges $m_{lj}[k]$ during each time step k in Iteration Step 1 of Algorithm 1) and a possible increase on the required number of transmissions for convergence. Analysis of the requirements on processor usage and the number of transmissions will be considered in the future in order to highlight the proposed algorithm's operational advantages.

5. Simulation results

In this section, we illustrate the behavior and the advantages of Algorithm 1 for the following scenarios.

A. We execute Algorithm 1 (i) over a randomly chosen static digraph of 10 nodes, and (ii) a dynamic digraph of 10 nodes whose union graph is equal to the nominal digraph of the first case after $l = 5$ time steps (note that in the second case, we adopt the set of assumptions A_1, A_2). The initial quantized states of the nodes were randomly chosen between 1 and 50 (for each node, the initial state was a randomly chosen integer value between 1 and 50 with probability $\frac{1}{50}$) with the average of the initial states of the nodes turning out to be $q = \frac{368}{10} = 36.8$ which means that $\lfloor q \rfloor = 36$ and $\lceil q \rceil = 37$.

B. We compare the performance of Algorithm 1 against existing algorithms over *static* strongly connected directed networks. Specifically, we show the average number of time steps needed for quantized average consensus to be reached over 1000 randomly generated *static* digraphs of 20 nodes each. The initial quantized states of the nodes were also randomly chosen between 1 and 50 with the average of the initial states of the nodes turning out to be $q = \frac{388}{20} = 19.4$ (for convenience, the initial quantized state of each node remained the same for each one of the 1000 randomly generated digraphs). We compare the performance of Algorithm 1 against nine other algorithms: (a) the distributed averaging algorithm with quantized communication presented in Rikos and Hadjicostis (2020) in which, at each time step k, each agent splits its mass variables in equal pieces and then transmits all of the pieces to randomly chosen out-neighbors; (b) the distributed averaging algorithm with quantized communication presented in Rikos and Hadjicostis (2021) in which, at each time step k, each agent sends its mass variables towards an out-neighbor chosen according to a priority in the form of a quantized fraction; (c) the distributed averaging algorithm with quantized communication presented in Chamie et al. (2016) in which, at each time step k, each agent v_j broadcasts a quantized version of its own state towards its out-neighbors; (d) the quantized asymmetric averaging algorithm presented in Cai and Ishii (2011) in which, at each time step k, one edge, say edge (v_ℓ, v_j), is selected at random and node v_j sends its state information and surplus to node v_ℓ, which performs updates over its own state and surplus values; (e) the quantized gossip algorithm presented in Kashyap et al. (2007) in which, at each time step k, one edge is selected at random, independently from earlier instants, and the states of the nodes that the selected edge is incident upon are updated; (f) the deterministic algorithm in Aysal et al. (2008) where nodes use probabilistic quantization and operate over an *undirected* connected communication topology[2]; (g) the distributed algorithm in Frasca et al. (2009) where each node stores real values and transmits quantized values with refined quantization step, and the exact average is calculated asymptotically; (h) the algorithm in Carli et al. (2010) where nodes store real values, transmit quantized values and communicate in a gossip fashion; (i) the algorithm in Dibaji et al. (2017) where nodes quantize their states in a probabilistic manner and consensus is reached (which may not be equal to the average).

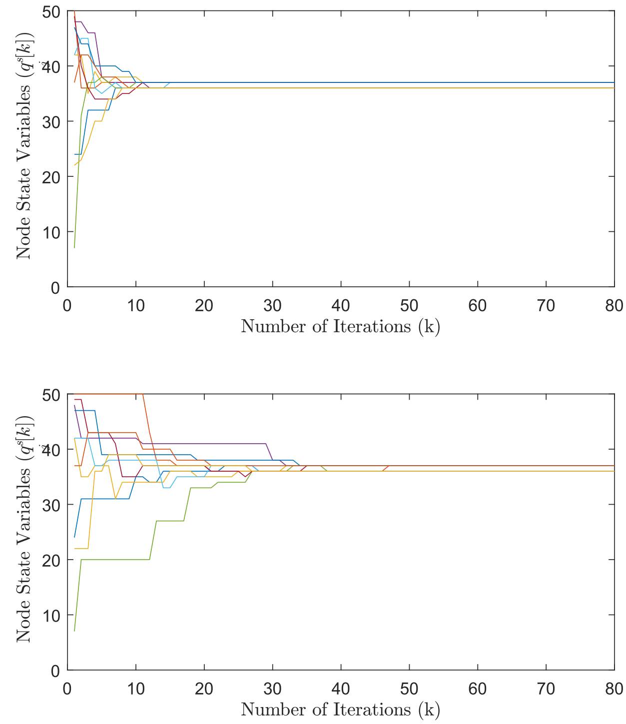

Fig. 3. Execution of Algorithm 1 over a randomly chosen static digraph of 10 nodes (*top*) and a dynamic digraph of 10 nodes (*bottom*).

Fig. 3 shows the operation of Algorithm 1 over a randomly chosen static digraph of 10 nodes and over a dynamic digraph of 10 nodes whose union is equal to the nominal digraph after $l = 5$ time steps (i.e., assumptions A_1, A_2 hold). The initial values are the same in both cases and the average of the initial states of the nodes is equal to $q = \frac{368}{10} = 36.8$. On the top of Fig. 3 we can see that every node is able to reach quantized average consensus after 10 time steps and the states of the nodes stabilize to either $\lfloor q \rfloor = 36$ or $\lceil q \rceil = 37$ after 15 time steps. At the bottom of Fig. 3 we can see that Algorithm 1 requires more steps to converge due to the dynamic nature of the communication topology (since the union graph of the dynamic digraph is equal to the nominal digraph after $l = 5$ time steps) and each node's state is able to stabilize either to $\lfloor q \rfloor = 36$ or to $\lceil q \rceil = 37$ after 47 time steps. This makes Algorithm 1 the first algorithm in the literature to achieve oscillation-free quantized average consensus after a finite number of time steps over a dynamic digraph without any network requirements, since (i) in Cai and Ishii (2011) the calculation of the quantized average relies on a static threshold that depends on the number of nodes in the network; (ii) in Chamie et al. (2016) the operation requires a set of weights over the links of the dynamic digraph that form a doubly stochastic matrix, which need to be recalculated again during each time step (see Gharesifard & Cortés, 2012; Rikos, Charalambous, & Hadjicostis, 2014) while the states of the nodes exhibit an oscillating behavior; and (iii) in Kashyap et al. (2007) the operation requires bidirectional communication (i.e., undirected graph) and the states of the nodes also exhibit oscillating behavior.

Fig. 4 shows the average number of time steps needed for quantized average consensus to be reached over 1000 randomly generated digraphs of 20 nodes each, in which the average of the nodes initial states is equal to $q = \frac{388}{20} = 19.4$. In Fig. 4 we can see that Algorithm 1 is among the fastest algorithms in

[2] The algorithms in Aysal et al. (2008) and Kashyap et al. (2007) require the underlying graph to be undirected. For this reason, in Fig. 4, for Aysal et al. (2008) and Kashyap et al. (2007), we make the randomly generated underlying digraphs undirected (by enforcing that if $(v_j, v_i) \in \mathcal{E}$ then also $(v_i, v_j) \in \mathcal{E}$) while, for the algorithms in Cai and Ishii (2011), Carli et al. (2010), Chamie et al. (2016), Dibaji et al. (2017), Frasca et al. (2009) and Rikos and Hadjicostis (2020, 2021), the randomly generated underlying graph is generally directed.

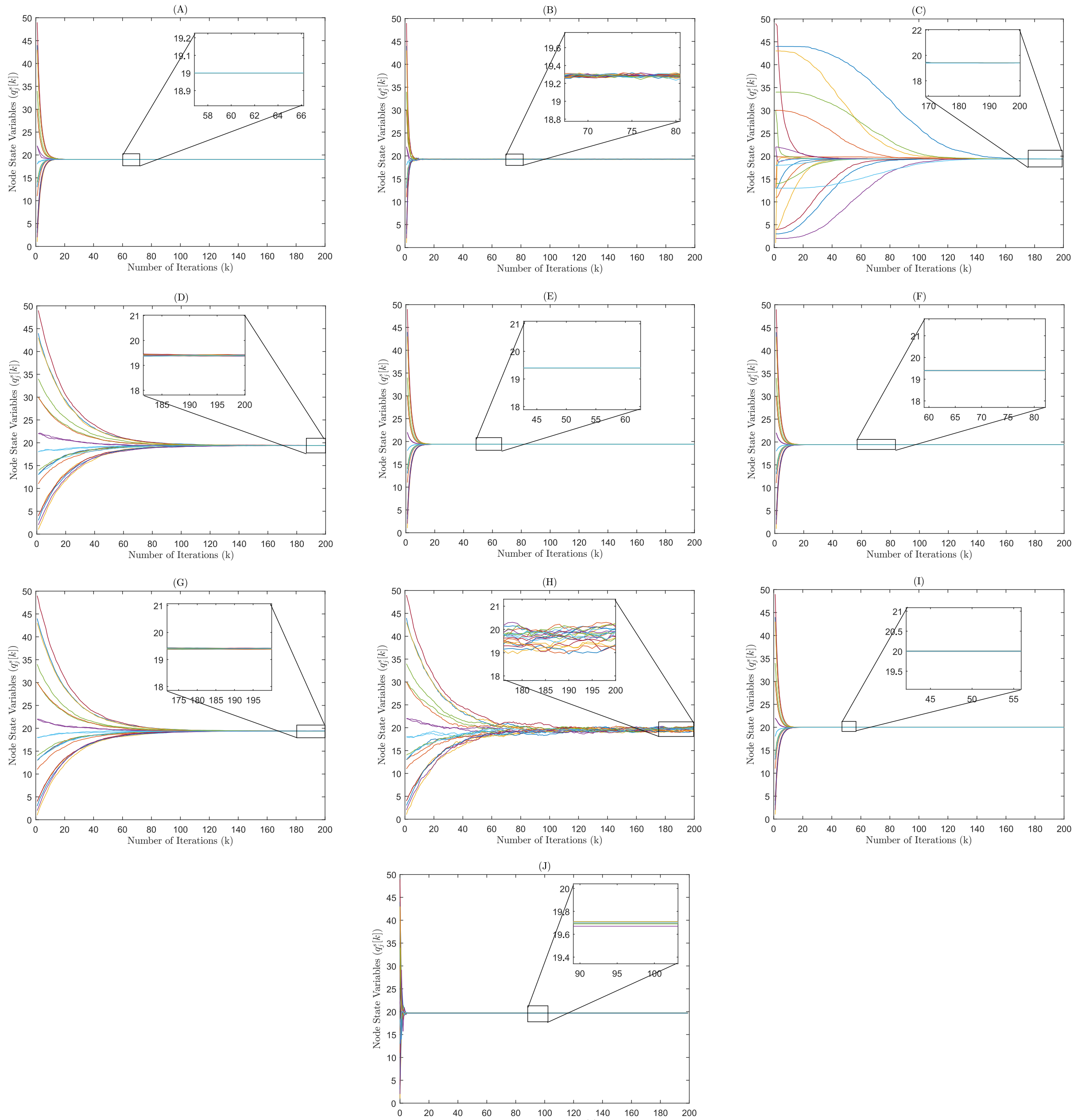

Fig. 4. Comparison between Algorithm 1 (A), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2020) (B), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2021) (C), the quantized gossip algorithm in Kashyap et al. (2007) (D), the deterministic algorithm in Aysal et al. (2008) (undirected graph) (E), the distributed algorithm in Frasca et al. (2009) (F), the distributed algorithm in Carli et al. (2010) (G), the quantized asymmetric averaging algorithm in Cai and Ishii (2011) (H), the distributed averaging algorithm with quantized communication in Chamie et al. (2016) (I), and the distributed algorithm in Dibaji et al. (2017) (undirected graph) (J), averaged over 1000 randomly generated strongly connected digraphs of 20 nodes each.

the current literature for the case where it operates over static communication networks. Its convergence speed is almost equal to that in Chamie et al. (2016), Frasca et al. (2009) and Aysal et al. (2008), with the difference, however, being that every node's state is able to stabilize either to $\lfloor q \rfloor = 19$ or $\lceil q \rceil = 20$ rather than oscillate between these states. Specifically, during Algorithm 1 (see (A)) each node's state becomes equal to $\lfloor q \rfloor = 19$. However, we have (i) for the algorithms in Rikos and Hadjicostis (2020) and Chamie et al. (2016), the state of each node does not become equal to a specific value due to oscillations between $\lfloor q \rfloor = 19$ or $\lceil q \rceil = 20$; (ii) in Dibaji et al. (2017) the nodes achieve consensus, which may not be the average of the initial states. Furthermore, Algorithm 1 has no prerequisites regarding the underlying communication topology (e.g., there is no need to obtain a set of weights that form a doubly stochastic matrix Aysal et al., 2008; Chamie et al., 2016; Frasca et al., 2009). This means that Algorithm 1 is among the fastest algorithms in the literature and also it (i) does not have prerequisites regarding the underlying communication topology (Aysal et al., 2008; Chamie et al., 2016; Frasca et al., 2009), (ii) does not oscillate between the ceiling and the floor of the real average of the initial states (Rikos & Hadjicostis, 2020), and (iii) is able to calculate the average of the initial states (Dibaji et al., 2017).

Fig. 5 considers the same experiments as Fig. 4, but plots the distance from average q in (1) defined as

$$e[k] = \left\lfloor \sqrt{\sum_{v_j \in \mathcal{V}} (A(j))^2} \right\rfloor, \tag{20}$$

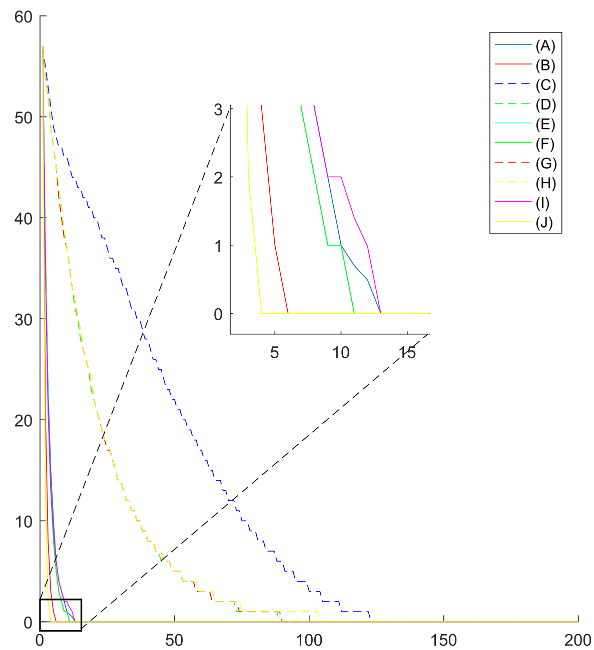

Fig. 5. Distance from Average $e[k]$ defined in (20) for Algorithm 1 (A), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2020) (B), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2021) (C), the quantized gossip algorithm in Kashyap et al. (2007) (D), the deterministic algorithm in Aysal et al. (2008) (undirected graph) (E), the distributed algorithm in Frasca et al. (2009) (F), the distributed algorithm in Carli et al. (2010) (G), the quantized asymmetric averaging algorithm in Cai and Ishii (2011) (H), the distributed averaging algorithm with quantized communication in Chamie et al. (2016) (I), and the distributed algorithm in Dibaji et al. (2017) (undirected graph) (J), averaged over 1000 randomly generated strongly connected digraphs of 20 nodes each.

where

$$A(j) = \max(q_j[k] - \lceil q \rceil, 0) + \max(\lfloor q \rfloor - q_j[k], 0).$$

Note here that in $e[k]$ we use the floor operation because we assume that if the difference of the state of a node with the true average q is less than 1, then this node has converged to the average. In Fig. 5, we can see that Algorithm 1 requires the same number of time steps as Chamie et al. (2016). Also, Algorithm 1 requires a few more steps than Aysal et al. (2008), Frasca et al. (2009), Rikos and Hadjicostis (2020). However, as mentioned previously, Algorithm 1 does not need a set of weights over the digraph links that form a doubly stochastic matrix (Aysal et al., 2008; Chamie et al., 2016; Frasca et al., 2009), and does not oscillate between the ceiling and the floor of the real average (Chamie et al., 2016; Rikos & Hadjicostis, 2020).

Table 2 shows the average number of transmissions required for convergence of the algorithms presented in Fig. 5. Specifically, the number of transmissions are averaged over 1000 randomly generated static digraphs of 20 nodes each, in which the average of the nodes initial states is equal to $q = \frac{510}{20} = 25.5$. Note here that in Table 2 we count the number of nodes performing transmissions until the state of every node is equal either to the ceiling or the floor of the quantized average of the initial values. In Table 2, we can see that Rikos and Hadjicostis (2020) require the least number of transmissions for convergence. However, the operation of Algorithm 1 over static digraphs, requires almost the same number of transmissions as Chamie et al. (2016) and Kashyap et al. (2007), and a few more transmissions than Carli et al. (2010), Dibaji et al. (2017) and Rikos and Hadjicostis (2021).

Table 2
Average Number of Transmissions During Operation of Algorithm 1 (A), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2020) (B), the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2021) (C), the quantized gossip algorithm in Kashyap et al. (2007) (D), the deterministic algorithm in Aysal et al. (2008) (undirected graph) (E), the distributed algorithm in Frasca et al. (2009) (F), the distributed algorithm in Carli et al. (2010) (G), the quantized asymmetric averaging algorithm in Cai and Ishii (2011) (H), the distributed averaging algorithm with quantized communication in Chamie et al. (2016) (I), and the distributed algorithm in Dibaji et al. (2017) (undirected graph) (J), averaged over 1000 randomly generated strongly connected digraphs of 20 nodes each.

Algorithm	N. Tr.
(A)	211.88
(B)	89.52
(C)	131.28
(D)	218.48
(E)	713.11
(F)	525.52
(G)	147.14
(H)	383.41
(I)	221.34
(J)	180.85

Remark 4. Algorithm 1 possesses attractive features for consensus-based distributed optimization. Apart from operating with quantized states which reduces the communication bottleneck (Reisizadeh et al., 2020; Shlezinger et al., 2020), it also allows for fast distributed averaging, which makes it suitable as an intermediate step between optimization operations (Grammenos et al., 2020; Khatana & Salapaka, 2020). In the latter case, the convergence speed of the averaging algorithm plays a significant role for the overall convergence speed of the optimization procedure (as we saw in this section, the convergence speed of Algorithm 1 significantly outperforms other algorithms in the available literature for the case where the communication topology is static). Specifically, the convergence speed of Algorithm 1 over static networks plays an important role in Rikos et al. (2021). In this paper a set of server nodes operates over a large scale network, i.e., a data center. Server nodes aim to balance their CPU utilization by deciding how to allocate CPU resources to workloads in a distributed fashion. The optimal allocation algorithm in Rikos et al. (2021) relies on Algorithm 1 to achieve fast convergence and also to calculate the optimal scheduling in finite time with quantized (i.e., efficient) communication. In this case, Algorithm 1 achieves finite time convergence after 15 − 20 time steps for large scale networks consisting of 10 000 nodes (see Fig. 2 in Rikos et al., 2021). The fast convergence speed and finite time convergence of Algorithm 1 is also significant for the operation of the test allocation algorithm in Nylof et al. (2022). In this paper, a set of cities (or separate entities) in a country aim to optimally allocate tests according to infections for monitoring the spread of a pandemic. The optimal allocation is also done in a privacy-preserving manner (since the number of infections and stored test kits contain sensitive health data). The test allocation algorithm in Nylof et al. (2022) is based on a variation of Algorithm 1 in which a privacy preserving mechanism is incorporated. This variation of Algorithm 1 is able to calculate the optimal allocation in finite time with quantized (i.e., efficient) communication and processing while it preserves the privacy of the state of every node from other possibly curious nodes. In this case, the operation of the variation of Algorithm 1 achieves finite time convergence after 10 − 50 time steps for large scale networks consisting of 100 nodes (see Fig. 3 in Nylof et al., 2022). Finally, note that Algorithm 1 can also find various applications on load balancing and voting schemes. In these cases

each node needs to calculate a specific state and the oscillation avoiding characteristic of Algorithm 1 facilitates convergence to a specific state rather than leading to oscillations between two different states/decisions.

6. Conclusions

We have considered the quantized average consensus problem over dynamically changing communication networks. We presented a randomized distributed averaging algorithm in which processing, storing and exchange of information between neighboring agents is subject to uniform quantization. We analyzed its operation, established its correctness and showed that it allows every agent to reach a consensus state equal to either the ceiling or the floor of the real average (thus avoiding oscillating behavior) without any specific requirements regarding the network that describes the underlying communication topology, apart from jointly strong connectivity. We analyzed the convergence of our algorithm and showed that it almost surely converges to the average of the initial states in finite time. Furthermore, we presented simulation results and argued that its convergence speed is among the fastest in the available literature.

In the future we plan to explore the performance of Algorithm 1 in the presence of network unreliability (e.g., delays and packet drops). Furthermore, we plan to utilize our algorithm for designing consensus-based distributed optimization algorithms.

References

Aysal, T. C., Coates, M., & Rabbat, M. (2007). Distributed average consensus using probabilistic quantization. In *IEEE/SP workshop on statistical signal processing* (pp. 640–644).

Aysal, T. C., Coates, M. J., & Rabbat, M. G. (2008). Distributed average consensus with dithered quantization. *IEEE Transactions on Signal Processing*, *56*(10), 4905–4918.

Bambos, N., Chen, S. C., & Pottie, G. J. (2000). Channel access algorithms with active link protection for wireless communication networks with power control. *IEEE/ACM Transactions on Networking*, *8*(5), 583–597.

Basar, T., Etesami, S., & Olshevsky, A. (2016). Convergence time of quantized Metropolis consensus over time-varying networks. *IEEE Transactions on Automatic Control*, *61*(12), 4048–4054.

Benezit, F., Thiran, P., & Vetterli, M. (2011). The distributed multiple voting problem. *IEEE Journal of Selected Topics in Signal Processing*, *5*(4), 791–804.

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., & Tsitsiklis, J. N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the IEEE conference on decision and control* (pp. 2996–3000).

Cai, K., & Ishii, H. (2011). Quantized consensus and averaging on gossip digraphs. *IEEE Transactions on Automatic Control*, *56*(9), 2087–2100.

Carli, R., Fagnani, F., Frasca, P., & Zampieri, S. (2010). Gossip consensus algorithms via quantized communication. *Automatica*, *46*(1), 70–80.

Carli, R., Fagnani, F., Speranzon, A., & Zampieri, S. (2008). Communication constraints in the average consensus problem. *Automatica*, *44*(3), 671–684.

Chamie, M. E., Liu, J., & Basar, T. (2016). Design and analysis of distributed averaging with quantized communication. *IEEE Transactions on Automatic Control*, *61*(12), 3870–3884.

Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C. N., & Johansson, M. (2013). Decentralised minimum-time average consensus in digraphs. In *Proceedings of the IEEE conference on decision and control* (pp. 2617–2622).

Dibaji, S. M., Ishii, H., & Tempo, R. (2017). Resilient randomized quantized consensus. *IEEE Transactions on Automatic Control*, *63*(8), 2508–2522.

Elgabli, A., Park, J., Bedi, A. S., Issaid, C. B., Bennis, M., & Aggarwal, V. (2021). Q-GADMM: Quantized group ADMM for communication efficient decentralized machine learning. *IEEE Transactions on Communications*, *69*(1), 164–181.

Etesami, S., & Basar, T. (2016). Convergence time for unbiased quantized consensus over static and dynamic networks. *IEEE Transactions on Automatic Control*, *61*(2), 443–455.

Frasca, P., Carli, R., Fagnani, F., & Zampieri, S. (2009). Average consensus on networks with quantized communication. *International Journal on Robust and Nonlinear Control*, *19*(16), 1787–1816.

Garcia, E., Cao, Y., Yu, H., Antsaklis, P., & Casbeer, D. (2013). Decentralised event-triggered cooperative control with limited communication. *International Journal of Control*, *86*(9), 1479–1488.

Gharesifard, B., & Cortés, J. (2012). Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, *18*(6), 539–557.

Grammenos, A., Charalambous, T., & Kalyvianaki, E. (2020). CPU scheduling in data centers using asynchronous finite-time distributed coordination mechanisms. arXiv preprint arXiv:2101.06139.

Hadjicostis, C. N., Domínguez-García, A. D., & Charalambous, T. (2018). Distributed averaging and balancing in network systems, with applications to coordination and control. *Found. Trends® Syst. Control*, *5*(3–4).

Jiang, P., & Agrawal, G. (2018). A linear speedup analysis of distributed deep learning with sparse and quantized communication. *Advances in Neural Information Processing Systems*, 2525–2536.

Kar, S., & Moura, J. M. F. (2010). Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *IEEE Transactions on Signal Processing*, *58*(3), 1383–1400.

Kashyap, A., Basar, T., & Srikant, R. (2007). Quantized consensus. *Automatica*, *43*(7), 1192–1203.

Khatana, V., & Salapaka, M. V. (2020). DC-DistADMM: ADMM algorithm for constrained distributed optimization over directed graphs. arXiv preprint arXiv:2003.13742.

Lavaei, J., & Murray, R. M. (2012). Quantized consensus by means of gossip algorithm. *IEEE Transactions on Automatic Control*, *57*(1), 19–32.

Li, T., Fu, M., Xie, L., & Zhang, J. F. (2011). Distributed consensus with limited communication data rate. *IEEE Transactions on Automatic Control*, *56*(2), 279–292.

Liu, Z., Chen, Z., & Yuan, Z. (2012). Event-triggered average-consensus of multi-agent systems with weighted and direct topology. *Journal of Systems Science and Complexity*, *25*(5), 845–855.

Liu, J., Mou, S., Morse, A. S., Anderson, B. D. O., & Yu, C. (2011). Deterministic gossiping. *Proceedings of the IEEE*, *99*(9), 1505–1524.

Lynch, N. (1996). *Distributed algorithms*. San Mateo, CA: Morgan Kaufmann Publishers.

Mou, S., Garcia, E., & Casbeer, D. W. (2017). Distributed algorithms for the average bridge consensus. In *Proceedings of the IEEE conference on control technology and applications* (pp. 1710–1715).

Nedic, A., Olshevsky, A., Ozdaglar, A., & Tsitsiklis, J. N. (2008). Distributed subgradient methods and quantization effects. In *Proceedings of the IEEE conference on decision and control* (pp. 4177–4184).

Nedic, A., Olshevsky, A., Ozdaglar, A., & Tsitsiklis, J. N. (2009). On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, *54*(11), 2506–2517.

Nowzari, C., & Cortés, J. (2016). Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica*, *68*, 237–244.

Nylof, J., Rikos, A. I., Gracy, S., & Johansson, K. H. (2022). Distributed optimal allocation with quantized communication and privacy-preserving guarantees. arXiv preprint arXiv:2109.14481.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*(9), 1520–1533.

Rabbat, M. G., & Nowak, R. D. (2005). Quantized incremental algorithms for distributed optimization. *IEEE Journal on Selected Areas in Communications*, *23*(4), 798–808.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., & Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the 23rd international conference on artificial intelligence and statistics* (pp. 2021–2031).

Rikos, A. I., Charalambous, T., & Hadjicostis, C. N. (2014). Distributed weight balancing over digraphs. *IEEE Transactions on Control of Network Systems*, *1*(2), 190–201.

Rikos, A. I., Charalambous, T., Johansson, K. H., & Hadjicostis, C. N. (2020). Privacy-preserving event-triggered quantized average consensus. In *Proceedings of the IEEE conference on decision and control* (pp. 6246–6253).

Rikos, A. I., Grammenos, A., Kalyvianaki, E., Hadjicostis, C. N., Charalambous, T., & Johansson, K. H. (2021). Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism. In *IEEE conference on decision and control* (pp. 6276–6281).

Rikos, A. I., & Hadjicostis, C. N. (2018). Distributed average consensus under quantized communication via event-triggered mass summation. In *Proceedings of the IEEE conference on decision and control* (pp. 894–899).

Rikos, A. I., & Hadjicostis, C. N. (2020). Distributed average consensus under quantized communication via event-triggered mass splitting. In *Proceedings of the 20th IFAC world congress* (pp. 3019–3024).

Rikos, A. I., & Hadjicostis, C. N. (2021). Event-triggered quantized average consensus via ratios of accumulated values. *IEEE Transactions on Automatic Control*, *66*(3), 1293–1300.

Ruan, M., Gao, H., & Wang, Y. (2019). Secure and privacy-preserving consensus. *IEEE Transactions on Automatic Control*, *64*(10), 4035–4049.

Schenato, L., & Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of the IEEE conference on decision and control* (pp. 2289–2294).

Seyboth, G. S., Dimarogonas, D. V., & Johansson, K. H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, *49*(1), 245–252.

Shlezinger, N., Chen, M., Eldar, Y. C., Poor, H. V., & Cui, S. (2020). Federated learning with quantization constraints. In *IEEE international conference on acoustics, speech and signal processing* (pp. 8851–8855).

Sun, J., Chen, T., Giannakis, G., & Yang, Z. (2019). Communication-efficient distributed learning via lazily aggregated quantized gradients. In *Advances in neural information processing system* (pp. 3370–3380).

Sun, J., Chen, T., Giannakis, G. B., Yang, Q., & Yang, Z. (2020). Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1, (Early Access).

Sundaram, S., & Hadjicostis, C. N. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, *26*(4), 650–660.

Taheri, H., Mokhtari, A., Hassani, H., & Pedarsani, R. (2020). Quantized decentralized stochastic learning over directed graphs. In *Proceedings of the 37th international conference on machine learning* (pp. 9324–9333).

Thanou, D., Kokiopoulou, E., Pu, Y., & Frossard, P. (2013). Distributed average consensus with quantization refinement. *IEEE Transactions on Signal Processing*, *61*(1), 194–295.

Xiao, L., Boyd, S., & Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the international symposium on information processing in sensor networks* (pp. 63–70).

Apostolos I. Rikos received the B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the Department of Electrical and Computer Engineering, University of Cyprus in 2010, 2012 and 2018 respectively. In 2018 he joined the KIOS Research and Innovation Center of Excellence in Cyprus, where he was a Special Scientist and Associate Lecturer. Since February 2020, he has been a Postdoctoral Researcher at the Division of Decision and Control Systems of KTH Royal Institute of Technology. His research interests are in the area of distributed systems, privacy and security, distributed optimization, and algorithmic analysis.

Christoforos N. Hadjicostis received the B.S. degrees in electrical engineering, computer science and engineering, and in mathematics, the M.Eng. degree in electrical engineering and computer science in 1995, and the Ph.D. degree in electrical engineering and computer science in 1999, all from the Massachusetts Institute of Technology, Cambridge. In 1999, he joined the Faculty at the University of Illinois at Urbana-Champaign, where he served as Assistant and then Associate Professor with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Information Trust Institute. Since 2007, he has been with the Department of Electrical and Computer Engineering, University of Cyprus, where he is currently Professor. His research focuses on fault diagnosis and tolerance in distributed dynamic systems, error control coding, monitoring, diagnosis and control of large-scale discrete-event systems, and applications to network security, energy distribution systems, anomaly detection, and medical diagnosis. Dr. Hadjicostis serves as Editor in Chief of the Journal of Discrete Event Dynamic Systems. In the past, he served as Associate Editor of Automatica, IEEE Transactions on Automatic Control, Nonlinear systems: Hybrid Systems, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Control Systems Technology, and IEEE Transactions on Circuits and Systems I.

Karl H. Johansson is Professor with the School of Electrical Engineering and Computer Science at KTH Royal Institute of Technology in Sweden and Director of Digital Futures. He received M.Sc. degree in Electrical Engineering and Ph.D. in Automatic Control from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. His research interests are in networked control systems and cyber–physical systems with applications in transportation, energy, and automation networks. He is President of the European Control Association and member of the IFAC Council, and has served on the IEEE Control Systems Society Board of Governors and the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He has received several best paper awards and other distinctions from IEEE, IFAC, and ACM. He has been awarded Swedish Research Council Distinguished Professor, Wallenberg Scholar with the Knut and Alice Wallenberg Foundation, Future Research Leader Award from the Swedish Foundation for Strategic Research, the triennial IFAC Young Author Prize, and IEEE Control Systems Society Distinguished Lecturer. He is Fellow of the IEEE and the Royal Swedish Academy of Engineering Sciences.