

# Decentralised Minimum-Time Average Consensus in Digraphs

Themistoklis Charalambous, Ye Yuan, Tao Yang, Wei Pan, Christoforos N. Hadjicostis and Michael Johansson

**Abstract**—Distributed algorithms for average consensus in directed graphs are typically asymptotic in the literature. In this work, we propose a protocol to distributively reach average consensus in a finite number of steps on interconnection topologies that form strongly connected directed graphs (digraphs). The average consensus value can be computed, based exclusively on local observations at each component, by running a protocol that requires each component to observe and store its own value over a finite and minimal number of steps, and to have knowledge of the number of its out-going links (i.e., the number of components to which it sends information). The proposed algorithm is demonstrated via illustrative examples.

## I. INTRODUCTION

A distributed multi-component system consists of a set of components (nodes) that can generally share information via connection links (edges), forming a directed communication topology (directed graph or digraph). The successful operation of a distributed system depends on the structure of the digraph which typically proves to be of vital importance for our ability to apply distributed strategies and perform various tasks. In general, the objective of a consensus problem is to have the agents belonging to a group agree upon a certain (*a priori* unknown) quantity of interest. When the agents have reached agreement, we say that the distributed system has reached consensus.

One of the most well known consensus problems is the so-called *average consensus* problem in which agents aim to reach agreement to the average of their initial values (see, e.g., [1], [2]). The initial value associated with each agent might be, for instance, a sensor measurement of some signal [3], [4], Bayesian belief of a decision to be taken [5], [6], or the capacity of distributed energy resources for the provisioning of ancillary services [7]. It has been shown in [8] that, under a fixed interconnection topology, average consensus is achieved by performing a linear iteration in a distributed fashion if the interconnection topology is both strongly connected and balanced, while gossip algorithms [9], [10] and convex optimization [11]–[13] require update matrices to be doubly stochastic. In undirected interconnection topologies, weight balanced and doubly stochastic update matrices can

easily be obtained, but this is significantly more challenging in directed interconnection topologies, which arise frequently in reality, due to the heterogeneity of communication systems and ranges. There exists a limited number of results (see, for example, [14]–[16]) that enable each agent to converge to the exact average of the initial values in an unbalanced (directed) graph. However, all of the average consensus algorithms currently present in the literature that are applicable to directed graphs only produce asymptotic convergence (i.e., exact average consensus is not reached in a finite number of steps). In addition, few of them (e.g., [17]) have addressed delays and topology changes.

Finite-time consensus was proposed by [18], [19], while [20] proposed a method to compute the asymptotic final consensus value in finite-time. Also, [21], [22] proposed a distributed algorithm with which any arbitrarily chosen agent can compute its asymptotic final consensus value in minimal-time. However, none of the aforementioned works is able to distributively compute – using only local information at each iteration – the average consensus value in a finite number of steps in a digraph.

In this work, we address the problem of designing a discrete-time coordination algorithm for exact average consensus in a finite number of steps. More specifically, we propose a distributed algorithm with which the exact average of the initial values is distributively computed in a finite number of steps. The algorithm developed is based on the following two approaches:

(a) An algorithm suggested in [14] that solves the average consensus problem in a directed graph using a linear iteration strategy. More specifically, each node  $v_j$  distributively sets the weights on its self-link and outgoing-links such that the collective set of weights forms a column stochastic weight matrix that is not necessarily row stochastic. Using the weights in this matrix, average consensus is reached via *ratio consensus*, i.e., two iterations (with appropriately chosen initial conditions) that run simultaneously so that the average can be obtained at each node by taking the ratio of the two values (maintained at each node for each of the two iterations).

(b) A decentralized minimum-time consensus (not average consensus) algorithm suggested in [21], [22] that allows any node to compute the consensus value of the whole network in finite time using minimum number of successive values of its own state history. More specifically, an algorithm is developed in which the minimum number of steps is obtained by checking a rank condition on a Hankel matrix of local observations.

Themistoklis Charalambous, Tao Yang and Mikael Johansson are with ACCESS Linnaeus Center, School of Electrical Engineering, KTH-Royal Institute of Technology, SE-100 44 Stockholm, Sweden. Emails: {themisc, taoyang, mikaelj}@kth.se.

Ye Yuan is with the Control Group, Department of Engineering, University of Cambridge, Cambridge, UK. Email: yy311@cam.ac.uk.

Wei Pan is with Department of Bioengineering, Imperial College London, London, UK. Emails: w.pan11@imperial.ac.uk.

Christoforos N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. E-mail: chadjic@ucy.ac.cy.

The remainder of the paper is organized as follows. In Section II, we provide necessary notation and background on linear algebra and graph theory. In Section III, the problem to be solved is formulated, and Section IV presents our main results. Illustrative examples are presented in Section V. Finally, Section VI presents concluding remarks and future directions.

## II. NOTATION AND PRELIMINARIES

### A. Notation

The set of real (integer) numbers is denoted by  $\mathbb{R}$  ( $\mathbb{Z}$ ) and the set of positive numbers (integers) is denoted by  $\mathbb{R}_+$  ( $\mathbb{Z}_+$ ).  $\mathbb{R}_+^n$  denotes the non-negative orthant of the  $n$ -dimensional real space  $\mathbb{R}^n$ . Vectors are denoted by small letters whereas matrices are denoted by capital letters.  $A^\top$  denotes the transpose of matrix  $A$ . The  $i^{\text{th}}$  component of a vector  $x$  is denoted by  $x_i$ , and the notation  $x \geq y$  implies that  $x_i \geq y_i$  for all components  $i$ . For  $A \in \mathbb{R}^{n \times n}$ ,  $a_{ij}$  denotes the entry in row  $i$  and column  $j$ . By  $\mathbb{1}$  we denote the all-ones vector and by  $I$  we denote the identity matrix (of appropriate dimensions). We also denote by  $e_j^\top = [0, \dots, 0, 1_{j^{\text{th}}}, 0, \dots, 0] \in \mathbb{R}^{1 \times n}$ , where the single “1” entry is at the  $j^{\text{th}}$  position.  $|A|$  is the element-wise absolute value of matrix  $A$  (i.e.,  $|A| \triangleq [|A_{ij}|]$ ),  $A \leq B$  ( $A < B$ ) is the (strict) element-wise inequality between matrices  $A$  and  $B$ . A matrix whose elements are nonnegative, called nonnegative matrix, is denoted by  $A \geq 0$  and a matrix whose elements are positive, called positive matrix, is denoted by  $A > 0$ .

In multi-component systems with fixed communication links (edges), the exchange of information between components (nodes) can be conveniently captured by a directed graph (digraph)  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  of order  $n$  ( $n \geq 2$ ), where  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  is the set of nodes and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges. A directed edge from node  $v_i$  to node  $v_j$  is denoted by  $\varepsilon_{ji} = (v_j, v_i) \in \mathcal{E}$  and represents a communication link that allows node  $v_j$  to receive information from node  $v_i$ . A graph is said to be undirected if and only if  $\varepsilon_{ji} \in \mathcal{E}$  implies  $\varepsilon_{ij} \in \mathcal{E}$ . In this paper, links are not required to be bidirectional, i.e. we deal with digraphs; for this reason, we use the terms “graph” and “digraph” interchangeably. Note that by convention and for notational purposes, we assume that the given graph does not include any self-loops (i.e.,  $\varepsilon_{jj} \notin \mathcal{E}$  for all  $v_j \in \mathcal{V}$ ) although each node  $v_j$  obviously has a link (access) to its own information. A digraph is called *strongly* connected if there exists a path from each vertex  $v_i$  of the graph to each vertex  $v_j$  ( $v_j \neq v_i$ ). In other words, for any  $v_j, v_i \in \mathcal{V}$ ,  $v_j \neq v_i$ , one can find a sequence of nodes  $v_i = v_{l_1}, v_{l_2}, v_{l_3}, \dots, v_{l_t} = v_j$  such that link  $(v_{l_{k+1}}, v_{l_k}) \in \mathcal{E}$  for all  $k = 1, 2, \dots, t - 1$ .

All nodes that can transmit information to node  $v_j$  directly are said to be in-neighbors of node  $v_j$  and belong to the set  $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid \varepsilon_{ji} \in \mathcal{E}\}$ . The cardinality of  $\mathcal{N}_j^-$ , is called the *in-degree* of  $v_j$  and is denoted by  $\mathcal{D}_j^- = |\mathcal{N}_j^-|$ . The nodes that receive information from node  $v_j$  belong to the set of out-neighbors of node  $v_j$ , denoted by  $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid \varepsilon_{lj} \in \mathcal{E}\}$ . The cardinality of  $\mathcal{N}_j^+$ , is called the *out-degree* of  $v_j$  and is denoted by  $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$ .

In the type of algorithms we consider, we associate a positive weight  $p_{ji}$  for each edge  $\varepsilon_{ji} \in \mathcal{E} \cup \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ . The nonnegative matrix  $P = [p_{ji}] \in \mathbb{R}_+^{n \times n}$  (with  $p_{ji}$  as the entry at its  $j^{\text{th}}$  row,  $i^{\text{th}}$  column position) is a weighted adjacency matrix (also referred to as weight matrix) that has zero entries at locations that do not correspond to directed edges (or self-edges) in the graph. In other words, apart from the main diagonal, the zero-nonzero structure of the adjacency matrix  $P$  matches exactly the given set of links in the graph. We use  $w_j[k] \in \mathbb{R}$  to denote the information state of node  $j$  at time  $t_k$ . In a synchronous setting, each node  $v_j$  updates and sends its information to its neighbors at discrete times  $t_0, t_1, t_2, \dots$ . We index nodes’ information states and any other information at time  $t_k$  by  $k$ . Hence, we use  $w_j[k] \in \mathbb{R}$  to denote the information state of node  $j$  at time  $t_k$ .

Each node updates its information state  $w_j[k]$  by combining the available information received by its neighbors  $w_i[k]$  ( $v_i \in \mathcal{N}_j^-$ ) using the positive weights  $p_{ji}[k]$ , that capture the weight of the information inflow from agent  $v_i$  to agent  $v_j$  at time  $k$ . In this work, we assume that each node  $v_j$  can choose its self-weight and the weights on its out-going links  $\mathcal{N}_j^+$  only. Hence, in its general form, each node updates its information state according to the following relation:

$$w_j[k+1] = p_{jj}w_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}w_i[k], \quad k \geq 0, \quad (1)$$

where  $w_j[0] \in \mathbb{R}$  is the initial state of node  $v_j$ . If we let  $w[k] = (w_1[k] \ w_2[k] \ \dots \ w_n[k])^\top$  and  $P = [p_{ji}] \in \mathbb{R}_+^{n \times n}$ , then (1) can be written in matrix form as

$$w[k+1] = Pw[k], \quad (2)$$

where  $w[0] = (w_1[0] \ w_2[0] \ \dots \ w_n[0])^\top \triangleq w_0$ . We say that the nodes asymptotically reach average consensus if

$$\lim_{k \rightarrow \infty} w_j[k] = \frac{\sum_{v_i \in \mathcal{V}} w_i[0]}{n}, \quad \forall v_j \in \mathcal{V}.$$

The necessary and sufficient conditions for (2) to reach average consensus are the following: (a)  $P$  has a simple eigenvalue at one with left eigenvector  $\mathbb{1}^\top$  and right eigenvector  $\mathbb{1}$ , and (b) all other eigenvalues of  $P$  have magnitude less than 1. If  $P \geq 0$  (as in our case), the necessary and sufficient condition is that  $P$  is a primitive doubly stochastic matrix. In an undirected graph, assuming each node knows  $n$  (or an upper bound  $n'$ ) and the graph is connected, each node  $v_j$  can distributively choose the weights on its outgoing links to be  $\frac{1}{n'}$  and set its diagonal to be  $1 - \frac{\mathcal{D}_j^+}{n'}$  (where  $D_j^+ = D_j^- \triangleq D_j$ ), so that the resulting  $P$  is primitive doubly stochastic. However, in a digraph, this choice does not necessarily lead to a doubly stochastic weight matrix.

## III. PROBLEM STATEMENT

In this work, we consider a digraph in which each node has knowledge of the number of its out-going links. We assume that each node  $v_j$  has access to local information only via its communication with the in-neighboring nodes

and does not have access to global information, such as the number of nodes in the network. As aforementioned, there exists a number of results that establish that agents can distributively choose the weights to ensure convergence to the exact average of the initial values in an undirected graph. However, all of the consensus algorithms currently present in the literature that successfully reach the average in a digraph only produce asymptotic convergence (i.e., exact average consensus is not reached in a finite number of steps).

The problem is to design a discrete-time coordination algorithm that allows every node  $v_j \in \mathcal{V}$  in a *directed graph* to distributively compute the exact average consensus in a finite number of steps.

#### IV. MAIN RESULTS

In this section, we show that if each node in a directed graph can observe and store the evolution of its own values over a finite number of steps, then the average consensus value can be computed immediately from the successive observations of two iterations (with appropriately chosen initial conditions) that run simultaneously so that the average can be obtained at each node by taking the ratio of the two *final values* (computed at each node for each of the two iterations). Hence, we present a distributed protocol with which each node can compute, based on its own local observations and after a minimal number of steps, the exact average.

In [14], an algorithm is suggested that solves the average consensus problem in a directed graph in which each node  $v_j$  distributively sets the weights on its self-link and outgoing-links to be  $\frac{1}{1+\mathcal{D}_j^+}$ , so that the resulting weight matrix  $P$  is column stochastic, but not necessarily row stochastic. Average consensus is reached by using this weight matrix to run two iterations with appropriately chosen initial conditions. The algorithm is stated below for a specific choice of weights on each link that assumes that each node knows its out-degree; note, however, that the algorithm works for any set of weights that adhere to the graph structure and form a primitive column stochastic weight matrix.

**Proposition 1:** [14] Consider a strongly connected digraph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ . Let  $y_j[k]$  and  $x_j[k]$  (for all  $v_j \in \mathcal{V}$  and  $k = 0, 1, 2, \dots$ ) be the result of the iterations

$$y_j[k+1] = p_{jj}y_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}y_i[k], \quad (3a)$$

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{v_i \in \mathcal{N}_j^-} p_{ji}x_i[k], \quad (3b)$$

where  $p_{lj} = \frac{1}{1+\mathcal{D}_j^+}$  for  $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$  (zeros otherwise), and the initial conditions are  $y[0] = y_0$  and  $x[0] = 1$ . Then, the solution to the average consensus problem can be asymptotically obtained as

$$\lim_{k \rightarrow \infty} \mu_j[k] = \frac{\sum_{v_i \in \mathcal{V}} y_i[0]}{|\mathcal{V}|}, \quad \forall v_j \in \mathcal{V},$$

$$\text{where } \mu_j[k] = \frac{y_j[k]}{x_j[k]}.$$

**Remark 1:** Proposition 1 proposes a decentralised algorithm with which the exact average is *asymptotically* reached, even if the directed graph is not balanced.

In what follows, we propose an algorithm (that is based on the algorithm in Proposition 1), with which every node can compute  $\mu_j \triangleq \lim_{k \rightarrow \infty} \mu_j[k]$  in a *minimum* number of steps.

**Definition 1:** (Minimal polynomial of a matrix pair) The minimal polynomial associated with the matrix pair  $[P, e_j^T]$  denoted by  $q_j(t) = t^{M_j+1} + \sum_{i=0}^{M_j} \alpha_i^{(j)} t^i$  is the monic polynomial of minimum degree  $M_j+1$  that satisfies  $e_j^T q_j(P) = 0$ .

Considering the iteration in (2) with weight matrix  $P$ , it is easy to show (e.g., using the techniques in [21]) that

$$\sum_{i=0}^{M_j+1} \alpha_i^{(j)} w_j[k+i] = 0, \quad \forall k \in \mathbb{Z}_+, \quad (4)$$

where  $\alpha_{M_j+1}^{(j)} = 1$ .

Let us now denote the  $z$ -transform of  $w_j[k]$  as  $W_j(z) \triangleq \mathbb{Z}(w_j[k])$ . From (4) and the time-shift property of the  $z$ -transform, it is easy to show (see [21], [22])

$$W_j(z) = \frac{\sum_{i=1}^{M_j+1} \alpha_i^{(j)} \sum_{\ell=0}^{i-1} w_j[\ell] z^{i-\ell}}{q_j(z)}, \quad (5)$$

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

$$p_j(z) \triangleq \frac{q_j(z)}{z-1} \triangleq \sum_{i=0}^{M_j} \beta_i^{(j)} z^i. \quad (6)$$

The application of the final value theorem [21], [22] yields:

$$\phi_y(j) = \lim_{k \rightarrow \infty} y_j[k] = \lim_{z \rightarrow 1} (z-1)Y_j(z) = \frac{y_{M_j}^T \beta_j}{\mathbf{1}^T \beta_j}, \quad (7a)$$

$$\phi_x(j) = \lim_{k \rightarrow \infty} x_j[k] = \lim_{z \rightarrow 1} (z-1)X_j(z) = \frac{x_{M_j}^T \beta_j}{\mathbf{1}^T \beta_j}, \quad (7b)$$

where  $y_{M_j}^T = (y_j[0], y_j[1], \dots, y_j[M_j])$ ,  $x_{M_j}^T = (x_j[0], x_j[1], \dots, x_j[M_j])$  and  $\beta_j$  is the vector of coefficients of the polynomial  $p_j(z)$ .

The next question is how one can obtain the coefficient vector  $\beta_j$  in the computation of final values, e.g., Eq. (7a) and (7b). Consider the vectors of  $2k+1$  successive discrete-time values at node  $v_j$ , given by

$$y_{2k}^T = (y_j[0], y_j[1], \dots, y_j[2k]),$$

$$x_{2k}^T = (x_j[0], x_j[1], \dots, x_j[2k]),$$

for the two iterations  $y_j[k]$  and  $x_j[k]$  at node  $v_j$  (as given in iterations (3a) and (3b)), respectively. Let us define their

associated Hankel matrices:

$$\Gamma\{y_{2k}^\top\} \triangleq \begin{bmatrix} y_j[0] & y_j[1] & \dots & y_j[k] \\ y_j[1] & y_j[2] & \dots & y_j[k+1] \\ \vdots & \vdots & \ddots & \vdots \\ y_j[k] & y_j[k+1] & \dots & y_j[2k] \end{bmatrix},$$

$$\Gamma\{x_{2k}^\top\} \triangleq \begin{bmatrix} x_j[0] & x_j[1] & \dots & x_j[k] \\ x_j[1] & x_j[2] & \dots & x_j[k+1] \\ \vdots & \vdots & \ddots & \vdots \\ x_j[k] & x_j[k+1] & \dots & x_j[2k] \end{bmatrix}.$$

We also consider the vector of differences between successive values of  $y_j[k]$  and  $z_j[k]$ :

$$\bar{y}_{2k}^\top = (y_j[1] - y_j[0], \dots, y_j[2k+1] - y_j[2k]),$$

$$\bar{x}_{2k}^\top = (x_j[1] - x_j[0], \dots, x_j[2k+1] - x_j[2k]).$$

It has been shown in [22] that  $\beta_j$  can be computed as the kernel of the first defective Hankel matrices  $\Gamma\{\bar{y}_{2k}^\top\}$  and  $\Gamma\{\bar{x}_{2k}^\top\}$  for arbitrary initial conditions  $y_0$  and  $x_0$ , except a set of initial conditions with Lebesgue measure zero.

Next, we provide our first main result, in which it is stated that the exact average  $\mu$  can be distributively obtained in finite-time in strongly connected digraphs.

**Theorem 1:** Consider a strongly connected graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ . Let  $y_j[k]$  and  $x_j[k]$  (for all  $v_j \in \mathcal{V}$  and  $k = 0, 1, 2, \dots$ ) be the result of the iterations (3a) and (3b), where  $P = [p_{ji}] \in \mathbb{R}_+^{n \times n}$  is any set of weights that adhere to the graph structure and form a primitive column stochastic weight matrix. Then, the solution to the average consensus can be distributively obtained in finite-time at each node  $v_j$ , by computing

$$\mu_j \triangleq \lim_{k \rightarrow \infty} \frac{y_j[k]}{x_j[k]} = \frac{\phi_y(j)}{\phi_x(j)} = \frac{y_{M_j}^\top \beta_j}{x_{M_j}^\top \beta_j}, \quad (8)$$

where  $\phi_y(j)$  and  $\phi_x(j)$  are given by Eqs. (7a) and (7b), respectively and  $\beta_j$  is the vector of coefficients, as defined in (6).

*Proof:* The consensus value of node  $v_j$  for each of the iterations (3a) (with initial condition  $y[0] = y_0$ ) and (3b) (with initial condition  $x[0] = \mathbf{1}$ ) is found by (7a)-(7b). Note that the vector  $\beta_j$  does not depend on the initial conditions and hence it is the same for each node for every initial condition not in the Lebesgue measure zero set. Hence,

$$\lim_{k \rightarrow \infty} \frac{y_j[k]}{x_j[k]} = \frac{\phi_y(j)}{\phi_x(j)} = \frac{y_{M_j}^\top \beta_j}{x_{M_j}^\top \beta_j}.$$

But, from ratio consensus and Proposition 1, we already know that

$$\lim_{k \rightarrow \infty} \mu_j[k] = \lim_{k \rightarrow \infty} \frac{y_j[k]}{x_j[k]} = \frac{\sum_{v_i \in \mathcal{V}} y_0(i)}{|\mathcal{V}|}.$$

Hence,

$$\lim_{k \rightarrow \infty} \mu_j[k] = \frac{y_{M_j}^\top \beta_j}{x_{M_j}^\top \beta_j}.$$

**Remark 2:** Theorem 1 states that the average consensus in a strongly connected digraph can be computed by the ratio of the final values computed for each of the iterations (3a) with initial condition  $y[0] = y_0$  and iteration (3b) with initial condition  $x[0] = \mathbf{1}$ . Note that  $x[0] = \mathbf{1}$  does not belong into the Lebesgue measure zero set of matrix  $P$  as defined in Proposition 1  $P\mathbf{1} = \mathbf{1}$ . This occurs only when  $P$  is row stochastic. By construction (see Proposition 1),  $P$  is column stochastic and  $p_{lj} = 1/(1 + \mathcal{D}_j^+)$  for  $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$  (zeros otherwise). Thus, having a second iteration does not impose any extra limitations in the computation of the average consensus value.

**Remark 3:** Each node does not need to know a priori how many values it needs to store. However, as soon as the square Hankel matrices lose rank, the nodes can stop storing information. Throughout the work we assume that nodes have enough storage ability to store at least as many values as necessary and the resulting defective matrix. Nevertheless, it has been shown in [21] that the number of steps required for predicting  $y$  and  $x$  are less than  $2n$  where  $n$  is the number of nodes in the network.

We now formally describe an algorithm, herein called *Algorithm 1*, in which the nodes distributively compute the exact average of the initial values in a finite number of steps.

---

**Algorithm 1** Decentralised minimum-time average consensus in digraphs

---

**Input:** A strongly connected digraph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  with  $n = |\mathcal{V}|$  nodes and  $m = |\mathcal{E}|$  edges.

**Data:** Successive observations for  $y_j[k]$  and  $x_j[k]$ ,  $\forall v_j \in \mathcal{V}$ ,  $k = 0, 1, 2, \dots$ , using iterations (3a) and (3b), with initial conditions  $y[0] = y_0$  and  $x[0] = \mathbf{1}$ , respectively.

**Step 1:** For  $k = 0, 1, 2, \dots$ , each node  $v_j \in \mathcal{V}$  runs the ratio consensus algorithm (3) and stores the vectors of differences  $\bar{y}_{M_j}^\top$  and  $\bar{x}_{M_j}^\top$  between successive values of  $y_j[k]$  and  $x_j[k]$ , respectively.

**Step 2:** Increase the dimension  $k$  of the square Hankel matrices  $\Gamma\{\bar{y}_{M_j}^\top\}$  and  $\Gamma\{\bar{x}_{M_j}^\top\}$  for each of the iterations, until they lose rank; store their first defective matrix.

**Step 3:** The kernel  $\beta_j = (\beta_0, \dots, \beta_{M_j-1}, 1)^\top$  of the first defective matrix gives the consensus values  $\phi_y$  and  $\phi_x$ , via (7a) and (7b), respectively.

**Step 4:** The average is computed by

$$\mu_j = \frac{\phi_y(j)}{\phi_x(j)} = \frac{y_{M_j}^\top \beta_j}{x_{M_j}^\top \beta_j}.$$


---

## V. EXAMPLES

In this section, we demonstrate the validity of the proposed methodology on simple strongly connected networks.

**Example 1:** Consider the directed network in Fig. 1 where each node  $v_j$  chooses its weight and the weight of its outgoing links to be  $(1 + \mathcal{D}_j^+)^{-1}$  (such that the sum

of all weights assigned by each node  $v_j$  is equal to 1). When each node updates its information state  $w_j[k]$  using

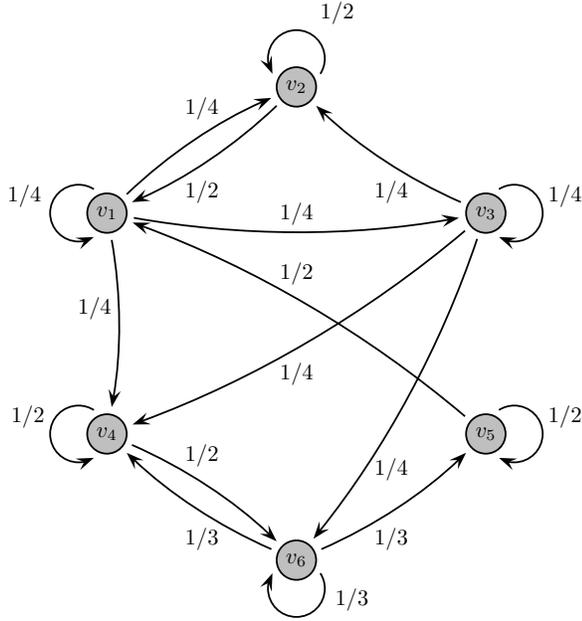


Fig. 1. A digraph consisting of six nodes.

Eq. (1), the information state for the whole network is given by  $w[k+1] = Pw[k]$  (as in Eq. (2)), where

$$P = \begin{pmatrix} 1/4 & 1/2 & 0 & 0 & 1/2 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/2 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1/2 & 1/3 \\ 0 & 0 & 1/4 & 1/2 & 0 & 1/3 \end{pmatrix}.$$

Each node  $v_j \in \mathcal{V}$  has an initial value  $y_0(j)$  and runs ratio consensus, i.e., the iteration in (3a) with initial value  $y_0(j)$ , and iteration (3b) with initial value  $x_0(j) = 1$ . In other words, we use the update formula (3) to run simultaneously two iterations with initial conditions  $y[0] = [-1 \ 0 \ 1 \ 2 \ 3 \ 4]^T$  and  $z[0] = \mathbf{1}$ . Since the update matrix is not doubly stochastic, the iteration (3) for this network does converge, but not necessarily to the average (as shown in Fig. 2 for the case when the initial condition is  $y[0]$ ). The final consensus vectors  $\phi_y$  and  $\phi_x$ , for initial conditions  $y[0]$  and  $x[0]$ , respectively, are given by

$$\begin{aligned} \phi_y &= [1.6119 \ 1.0746 \ 0.5373 \ 2.4179 \ 1.3433 \ 2.0149]^T, \\ \phi_x &= [1.0746 \ 0.7164 \ 0.3582 \ 1.6119 \ 0.8955 \ 1.3433]^T. \end{aligned}$$

Then, each node can compute the exact average  $\mu_j = \phi_y(j)/\phi_x(j)$ . For example, for node  $v_1$  he have

$$\mu_1 = \frac{\phi_y(1)}{\phi_x(1)} = \frac{1.6119}{1.0746} = 1.5.$$

The exact average for  $v_1$  is computed in 12 steps (i.e.,  $2(M_1 + 1) = 2 \times 6 = 12$  steps). See [22] for more details on how to compute the minimum number of steps required.

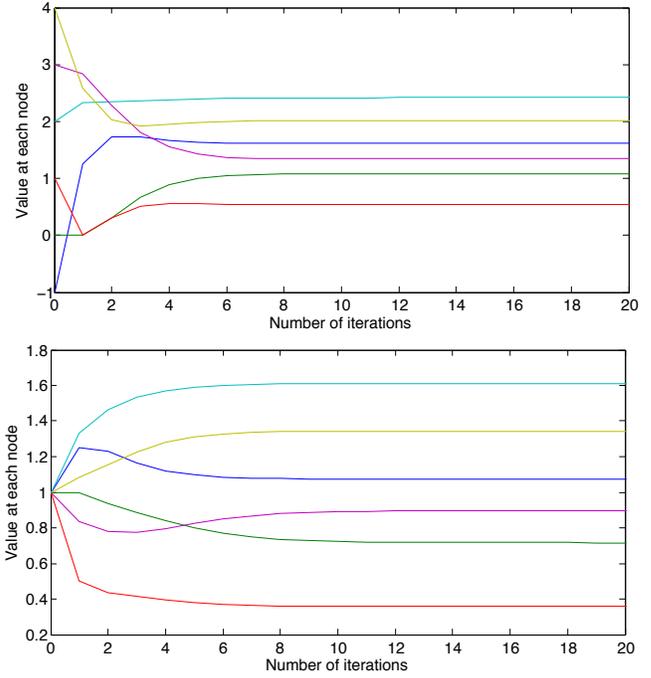


Fig. 2. Iteration (3a) with initial condition  $y[0] = [-1 \ 0 \ 1 \ 2 \ 3 \ 4]^T$  (above) and iteration (3b) with initial condition  $x[0] = \mathbf{1}$  (below), for the network in Fig. 1 do not converge to the average (the average in this case is 1.5) for the digraph, since  $P$  is not a doubly stochastic matrix.

The exact average is also justified by running the ratio consensus algorithm showing asymptotic convergence to the exact average. From the simulation we observe that the ratio consensus algorithm needs more or less 10–12 steps for the error to be small enough, illustrating that the minimum-time needed for decentralized computation of the exact average is fast enough.

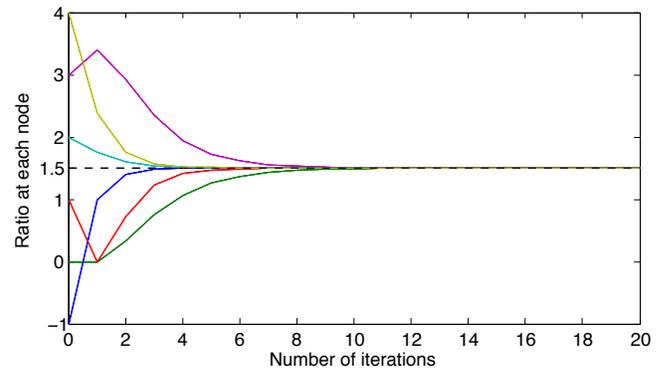


Fig. 3. By running two iterations  $y[k]$  and  $x[k]$  as in (3a)-(3b) (using the weight matrix  $P$  and initial conditions  $y[0] \triangleq y_0$  and  $x[0] = \mathbf{1}$ , respectively), average consensus is asymptotically reached for the ratio  $y_j[k]/x_j[k]$ .

**Example 2:** Next, we consider an example of a larger network, for which asymptotic convergence to the average takes a considerable amount of time. More specifically, we consider a Leslie (a discrete, age-structured model of population growth) matrix consisting of a 100 nodes. The

adjacency matrix  $P \in \mathbb{R}_+^{100 \times 100}$  is as follows.

$$P = \begin{pmatrix} 1/2 & 1/3 & 1/3 & 1/3 & \cdots & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1/3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1/3 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1/3 & 1/2 \end{pmatrix}.$$

Due to the structure of the network, the asymptotic convergence time is considerable (see Fig. 4). In our algorithm,

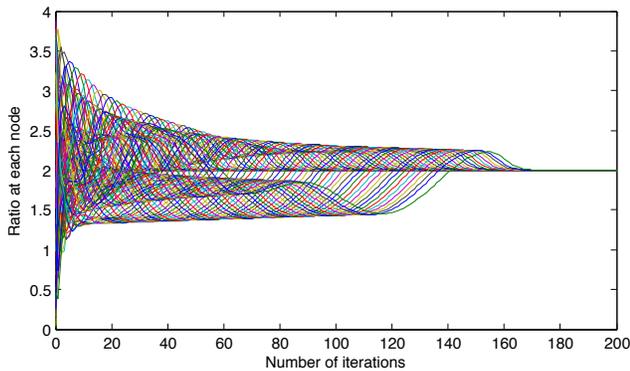


Fig. 4. Average consensus is asymptotically reached for the ratio  $y_j[k]/x_j[k]$  of each node  $v_j$  for a Leslie matrix of size 100.

a node can compute the exact average value quite fast; in this case, node  $v_{100}$  computes the average in 34 steps (i.e.,  $2(M_{100} + 1) = 2 \times 17 = 34$  steps), while the (asymptotic) ratio-consensus algorithm seems to need more than 160 steps for the error to be small enough (see Fig. 4).

## VI. CONCLUSIONS AND FUTURE DIRECTIONS

This paper proposes a discrete-time coordination algorithm that allows a networked system to distributively compute the exact average consensus in a finite number of steps using only output observations at each component of a multi-component system. The average consensus value can be obtained using exclusively local available information and by running a protocol that requires each component to have knowledge of the number of its out-going links (i.e., the number of components to which it sends information) and its own history of values. To the best of the authors' knowledge this is the first algorithm that can reach average consensus in a digraph in a finite number of steps.

Asymptotic convergence to the average for digraphs in the present of delays has been shown in [7]. Towards this end, the finite-time average consensus problem over digraphs in the presence of bounded delays in the communication links should be considered. In addition, an interesting problem to consider is the case when the interconnection topology is also switching.

## REFERENCES

- [1] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.
- [2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [3] A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, "Gossip algorithms for simultaneous distributed estimation and classification in sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 691–706, 2011.
- [4] L. Schenato and F. Fiorentin, "Average TimeSynch: a consensus-based protocol for time synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.
- [5] K. Greene, J. Kniss, G. Luger, and C. Stern, "Satisficing the masses: Applying game theory to large-scale, democratic decision problems," in *International Conference on Computational Science and Engineering (CSE)*, vol. 4, Aug. 2009, pp. 1156–1162.
- [6] G. Vogiatzis, I. MacGillivray, and M. Chli, "A probabilistic model for trust and reputation," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010, pp. 225–232.
- [7] C. N. Hadjicostis and T. Charalambous, "Asynchronous coordination of distributed energy resources for the provisioning of ancillary services," in *Proceedings of 49th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2011, pp. 1500–1507.
- [8] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [9] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, March 2005, pp. 1653–1664.
- [10] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, Nov. 2009.
- [11] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. on Optimization*, vol. 20, no. 3, pp. 1157–1170, Aug. 2009.
- [12] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, pp. 5917–5922.
- [13] —, "Asynchronous Newton-Raphson consensus for distributed convex optimization," in *Proceedings of 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12)*, Sep. 2012.
- [14] A. D. Domínguez-García and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *Proceedings of the First IEEE International Conference on Smart Grid Communications*, Oct. 2010, pp. 537–542.
- [15] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed averaging in sensor networks based on broadcast gossip algorithms," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 808–817, March 2011.
- [16] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, Nov. 2012.
- [17] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graphs," *IEEE Transactions on Automatic Control*, 2014 (to appear).
- [18] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.
- [19] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 950–955, April 2010.
- [20] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proceedings of the American Control Conference (ACC)*, July 2007, pp. 711–716.
- [21] Y. Yuan, G.-B. Stan, L. Shi, and J. Gonçalves, "Decentralised final value theorem for discrete-time LTI systems with application to minimal-time distributed consensus," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC)*, Dec. 2009, pp. 2664–2669.
- [22] Y. Yuan, G.-B. Stan, M. Barahona, L. Shi, and J. Gonçalves, "Decentralised minimal-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227–1235, May 2013.