

Distributed Formation of Balanced and Bistochastic Weighted Digraphs in Multi-Agent Systems

Themistoklis Charalambous and Christoforos N. Hadjicostis

Abstract—We propose two distributed algorithms, one for solving the weight-balance problem and another for solving the bistochastic matrix formation problem, in a distributed system whose components (nodes) can exchange information via interconnection links (edges) that form an arbitrary, possibly directed, strongly connected communication topology (digraph). Both distributed algorithms achieve their goal asymptotically and operate iteratively by having each node adapt the (non-negative) weights on its outgoing edges based on the weights of its incoming links. The weight-balancing algorithm is shown to admit geometric convergence rate, whereas the second algorithm, which is a modification of the weight-balancing algorithm, leads asymptotically to a bistochastic digraph with geometric convergence rate for a certain set of initial values. The two algorithms perform better than existing approaches, as illustrated by the examples we provide.

I. INTRODUCTION

A distributed system or network consists of a set of subsystems (nodes) that can share information via connection links (edges), which form a generally directed communication topology (digraph). This digraph typically proves to be of vital importance for our ability to apply distributed strategies and perform various tasks. Cooperative distributed control algorithms and protocols have received tremendous attention over the last decade by several diverse research communities (e.g., biology, physics, control, communication, and computer science), resulting in many recent advances in consensus-based approaches (see, for example, [1]–[7] and references therein).

In general, the objective of a consensus problem is to have the components of a distributed system agree upon a certain (*a priori* unknown) quantity of interest. When the components reach agreement, we say that the distributed system reaches consensus. Typical tasks include network coordination problems involving self-organization, formation patterns, parallel processing, and distributed optimization. One of the most well known consensus problems is the so-called average consensus problem in which components aim to reach the average of their initial values. The initial value associated with each component might be, for instance, a sensor measurement of some signal [8], the Bayesian belief of a decision to be taken [9], or the capacity of distributed energy resources for the provisioning of ancillary services [10]. Average consensus is closely related to two classes of

graphs: weight-balanced (for continuous-time systems) and bistochastic graphs (for discrete-time systems). A weighted digraph is called balanced if, for each node v_j , the sum of the weights of the edges outgoing from that node is equal to the sum of the weights of the edges incoming to that node. A weight-balanced digraph is bistochastic if all weights are nonnegative and, for each node v_j , the sum of weights of edges incoming to that node and the sum of the weights of edges out-going from that node is unity; this implies that the corresponding weight matrix is column and row stochastic (i.e., doubly stochastic or bistochastic).

It is shown in [2] that average consensus is achieved if the information exchange topology is both strongly connected and balanced, while gossip algorithms [11], [12] and convex optimization [13] admit update matrices which need to be doubly stochastic. These methods have applicability to a variety of topics, such as multi-component systems, cooperative control, and modeling the behaviour of various phenomena in biology and physics, such as flocking. Since their operation requires weight-balanced and bistochastic digraphs, it is important to be able to distributively transform a weighted digraph to a weight-balanced or bistochastic one, provided that each node is allowed to adjust the weight of its outgoing links accordingly.

In this paper, we address the problem of designing discrete-time coordination algorithms that allow a networked system to distributively obtain a set of weights that make it weight-balanced or bistochastic. This task is relatively straightforward when the underlying communication topology forms an undirected graph (i.e., when communication links are bi-directional) but more challenging when dealing with a digraph (i.e., when some communication links might be uni-directional). The paper proposes two algorithms that can be used in distributed networks with directed interconnection topologies; the first algorithm leads to a weight-balanced digraph whereas the second algorithm leads to a bistochastic digraph. Even though there exist some earlier approaches in the literature for weight balancing (e.g., [14] presents a finite-time algorithm but does not characterize the number of steps required in the worst case, whereas [15] presents an asymptotic algorithm whose rate is bounded explicitly based on the structure of the graph), the algorithm for weight-balancing is, to the best of our knowledge, the first asymptotic algorithm of this nature shown to admit geometric convergence rate. Work in the literature appears for bistochastic matrix formation as well (e.g., [16] proposes an asymptotic algorithm with an unspecified rate of convergence). Our proposed algorithm for obtaining a

Themistoklis Charalambous is with the School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. E-mail: themisc@kth.se.

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

bistochastic digraph is a modification of the weight-balancing algorithm that leads to a bistochastic digraph with asymptotic convergence. Under some minor additional assumptions, this second algorithm can also be shown to admit a geometric rate.

The remainder of the paper is organised as follows. In Section II, we provide necessary notation and background on graph properties. In Section III, the problem to be solved is formulated, and Sections IV and V present our main results in which we propose two algorithms, one for weight-balancing and one for bistochastic matrix formation. In Section VI, the derived algorithms are demonstrated via illustrative examples which compare their performance against existing approaches in the literature. Finally, Section VII presents concluding remarks and future directions.

II. NOTATION AND PRELIMINARIES

The sets of real, integer and natural numbers are denoted by \mathbb{R} , \mathbb{Z} and \mathbb{N} , respectively; their positive orthant is denoted by the subscript $+$ (e.g. \mathbb{R}_+). The symbol \mathbb{N}_0 denotes the set of nonnegative integers. Vectors are denoted by small letters whereas matrices are denoted by capital letters; A^{-1} denotes the inverse of matrix A . By I we denote the identity matrix (of appropriate dimensions), whereas by $\mathbb{1}$ we denote a column vector (of appropriate dimension) whose elements are all equal to one. 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 $A > 0$. Notation $\sigma(A)$ denotes the spectrum of matrix A , $\lambda(A)$ denotes an element of the spectrum of matrix A , and $\rho(A)$ denotes its spectral radius. Notation $\text{diag}(x_i)$ is used to denote the matrix with elements in the finite set $\{x_1, x_2, \dots, x_i, \dots\}$ on the leading diagonal and zeros elsewhere.

Let the exchange of information between nodes be modeled by a weighted digraph (directed graph) $\mathcal{G}(\mathcal{V}, \mathcal{E}, W)$ of order n ($n \geq 2$), where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges, and $W = [w_{ji}] \in \mathbb{R}_+^{n \times n}$ is a weighted $n \times n$ adjacency matrix where w_{ji} are nonnegative elements. A directed edge from node v_i to node v_j is denoted by $\varepsilon_{ji} = (v_j, v_i) \in \mathcal{E}$, which represents a directed information exchange link from node v_i to node v_j , i.e., it denotes that node v_j can receive information from node v_i . A directed edge $\varepsilon_{ji} \in \mathcal{E}$ if and only if $w_{ji} > 0$. Note that the definition of \mathcal{G} excludes self-edges (though self-weights are added when we consider bistochastic digraphs). The graph is undirected if and only if $\varepsilon_{ji} \in \mathcal{E}$ implies $\varepsilon_{ij} \in \mathcal{E}$ and $w_{ji} = w_{ij}$. Note that a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be thought of as a weighted digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, W)$ by defining the weighted adjacency matrix W with $w_{ji} = 1$ if $\varepsilon_{ji} \in \mathcal{E}$, and $w_{ji} = 0$ otherwise.

A digraph is called *strongly connected* if for each pair of vertices $v_j, v_i \in \mathcal{V}$, $v_j \neq v_i$, there exists a directed *path* from v_i to v_j , i.e., we can find a sequence of vertices $v_i \equiv v_{l_0}, v_{l_1}, \dots, v_{l_t} \equiv v_j$ such that $(v_{l_{\tau+1}}, v_{l_\tau}) \in \mathcal{E}$ for $\tau = 0, 1, \dots, t-1$. 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 j and it is denoted by \mathcal{D}_j^- . The nodes that receive information from node j comprise its out-neighbors and are 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 it is denoted by \mathcal{D}_j^+ . Given a weighted digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, W)$ of order n , the total *in-weight* of node v_j is denoted as S_j^- and is defined by $S_j^- = \sum_{v_i \in \mathcal{N}_j^-} w_{ji}$, whereas the total *out-weight* of node v_j is denoted by S_j^+ and is defined as $S_j^+ = \sum_{v_l \in \mathcal{N}_j^+} w_{lj}$.

Definition 1: A weighted digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, W)$ is called weight-balanced if the total in-weight equals the total out-weight, i.e., $S_j^- = S_j^+$ for every node $v_j \in \mathcal{V}$. A weight-balanced digraph is also called doubly stochastic (bistochastic) if its weights are nonnegative and each of its weighted adjacency matrix rows and columns sums to 1.

For the discrete-time setup we investigate, we conveniently define the time coordinate so that unity is the time between consecutive iterations. For example $S_j^+[k]$ will denote the value of the out-weight of node v_j at time instant k , $k \in \mathbb{N}_0$.

III. PROBLEM FORMULATION

Given a strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we want distributed algorithms that allow the nodes to adjust the weight on their outgoing edges to obtain a weight matrix $W = [w_{ji}]$ such that the following are achieved.

- (i) The weighted digraph becomes balanced in a distributed fashion; i.e., a weight matrix W is found such that $w_{ji} > 0$ for each edge $(v_j, v_i) \in \mathcal{E}$, $w_{ji} = 0$ if $(v_j, v_i) \notin \mathcal{E}$, and $S_j^+ = S_j^-$ for every $v_j \in \mathcal{V}$.
- (ii) The weighted digraph becomes bistochastic in a distributed fashion; i.e., a weight matrix W is found with nonnegative diagonal elements $w_{jj} \geq 0$, such that $w_{ji} > 0$ if $(v_j, v_i) \in \mathcal{E}$, $w_{ji} = 0$ if $(v_j, v_i) \notin \mathcal{E}$, $v_j \neq v_i$, and $w_{jj} + S_j^+ = w_{jj} + S_j^- = 1$ for every $v_j \in \mathcal{V}$.

IV. DISTRIBUTED ALGORITHM FOR WEIGHT-BALANCING

A. Description of the algorithm

Balancing a weighted digraph can be accomplished via a variety of algorithms. We introduce and analyse a distributed cooperative algorithm that exhibits asymptotic convergence and outperforms existing algorithms suggested in the literature [17]. The algorithm achieves weight-balance as long as the underlying digraph is strongly connected (or is a collection of strongly connected digraphs, a necessary and sufficient condition for balancing to be possible [18]). The rate of convergence of the algorithm is geometric and depends exclusively on the structure of the given digraph and some constant parameters chosen by the nodes.

Algorithm 1 is an iterative algorithm in which the nodes distributively adjust the weights of their outgoing links such that the digraph becomes asymptotically weight-balanced. We assume that each node observes but cannot set the weights of its incoming links. Given a strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the distributed algorithm has each node v_j

initialize the weights of all of its outgoing links to unity, i.e., $w_{lj}[0] = 1, \forall v_l \in \mathcal{N}_j^+$. Then, node v_j enters an iterative stage where it performs the following steps:

- 1) It computes its weight imbalance defined by

$$x_j[k] \triangleq S_j^-[k] - S_j^+[k]. \quad (1)$$

- 2) If $x_j[k]$ is positive (resp. negative), all the weights of its outgoing links are increased (resp. decreased) by an equal amount and proportionally to $x_j[k]$.

We discuss why the above distributed algorithm asymptotically leads to weights that balance the graph (and also characterize its rate of convergence) after we describe the algorithm in more detail. For simplicity, we assume that during the execution of the distributed algorithm, the nodes update the weights on their outgoing links in a synchronous¹ fashion. Also note that 2) above implies that w_{lj} for $v_l \in \mathcal{N}_j^+$ will always have the same value.

Algorithm 1 Weight balancing algorithm

Input: A strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges (and no self-loops).

Initialization: Each node $v_j \in \mathcal{V}$

- 1) Sets $w_{lj}[0] = 1, \forall v_l \in \mathcal{N}_j^+$.
- 2) Chooses $\beta_j \in (0, 1)$.

Iteration: For $k = 0, 1, 2, \dots$, each node $v_j \in \mathcal{V}$ updates the weights of each of its outgoing links $w_{lj}, \forall v_l \in \mathcal{N}_j^+$, as

$$w_{lj}[k+1] = w_{lj}[k] + \beta_j \left(\frac{S_j^-[k]}{\mathcal{D}_j^+} - w_{lj}[k] \right). \quad (2)$$

The intuition behind the proposed algorithm is that we compare $S_j^-[k]$ with $S_j^+[k] = \mathcal{D}_j^+ w_{lj}[k]$. If $S_j^+[k] > S_j^-[k]$ (resp. $S_j^+[k] < S_j^-[k]$), then the algorithm reduces (resp. increases) the weights on the outgoing links.

Proposition 1: If a digraph is strongly connected or is a collection of strongly connected digraphs, *Algorithm 1* reaches a steady state weight matrix W^* that forms a weight-balanced digraph, with geometric convergence rate equal to $R_\infty(P) = -\ln \delta(P)$, where

$$P_{ji} \triangleq \begin{cases} 1 - \beta_j, & \text{if } i = j, \\ \beta_j / \mathcal{D}_j^+, & \text{if } v_i \in \mathcal{N}_j^-, \end{cases} \quad (3)$$

and $\delta(P) \triangleq \max\{|\lambda| : \lambda \in \sigma(P), \lambda \neq 1\}$.

Proof: First, we observe from equation (2) that all the outgoing links have the same weight, i.e., $w_{vj} = w_{lj}, \forall v_l, v_l \in \mathcal{N}_j^+$ (because they are equal at initialization and they are updated in the same fashion). The fact that the outgoing links have the same weight facilitates broadcasting of the weights, avoiding separate transmissions or multiple broadcastings. Hence, from hereafter, we will denote the weight on any outgoing link of node v_j as w_j . In order to study the system with update formula (2) for each node

¹Even though we do not discuss this issue in the paper, asynchronous operation is not a problem.

in the graph, we define $w = (w_1 \ w_2 \ \dots \ w_n)^T$ with $w_j = w_{lj} (v_l \in \mathcal{N}_j^+)$. We can write the evolution of the weights in matrix form as follows.

$$w[k+1] = Pw[k], w[0] = w_0, \quad (4)$$

where $w_0 = \mathbf{1}$. It should be clear from the above update equation that the weights remain nonnegative during the execution of the algorithm.

Matrix P can be written as $P = I - B + BD^{-1}A$, where I is the identity matrix, $B = \text{diag}(\beta_j)$, $D = \text{diag}(\mathcal{D}_j^+)$ and A is the adjacency matrix with $a_{ji} = 1$ if $\varepsilon_{ji} \in \mathcal{E}$, and $a_{ji} = 0$ otherwise. Since $\sigma(D^{-1}A) = \sigma(AD^{-1})$, then $\rho(D^{-1}A) = \rho(AD^{-1})$. In addition, $\rho(AD^{-1}) = 1$ because matrix AD^{-1} is column stochastic. As a result, $\rho(D^{-1}A) = 1$. Also, note that $\bar{P} \triangleq I - B + AD^{-1}B$ is column stochastic and therefore $\rho(\bar{P}) = 1$. Furthermore,

$$\begin{aligned} \rho(\bar{P}) &= \rho(\bar{P}B^{-1}DD^{-1}B) = \rho(D^{-1}B\bar{P}B^{-1}D) \\ &= \rho(I - B + BD^{-1}A) = \rho(P) = 1. \end{aligned}$$

Since P is a nonnegative matrix, we can ask whether P is primitive,² i.e., whether $P^m > 0$ for some $m \geq 1$. Since the digraph is strongly connected for $0 < \beta_j < 1, \forall v_j \in \mathcal{V}$, and all the main diagonal entries are positive, we easily conclude that $m \leq n - 1$ [19, Lemma 8.5.5] and P is primitive. Hence, there is no other eigenvalue with modulus equal to the spectral radius. A sufficient condition for primitivity is that a nonnegative irreducible matrix A has at least one positive diagonal element [20, Example 8.3.3], which means that some β_j can also be set at unity (as long as at least one is strictly smaller than unity). Hence, iteration (4) has a geometric convergence rate $R_\infty(P) = -\ln \delta(P)$, where $\delta(P) \triangleq \max\{|\lambda| : \lambda \in \sigma(P), \lambda \neq 1\}$. In other words, $\delta(P)$ is the second largest of the moduli of the eigenvalues of P (see also [1], [2], [4]). ■

B. Illustrative Example

In this illustrative example (borrowed from [18]), we demonstrate the validity of the proposed algorithm in the network depicted in Figure 1. In our plots we are typically concerned with the absolute balance defined as $\varepsilon[k] = \sum_{j=1}^n |x_j[k]|$, where $x_j[k]$ is given in (1). If weight balance is achieved, then $\varepsilon[k] = 0$ and $x_j[k] = 0, \forall v_j \in \mathcal{V}$.

Figure 2 plots the absolute balance of *Algorithm 1* when $\beta_j = 0.1$ or 0.5 or 0.9 for all $v_j \in \mathcal{V}$, as it evolves during the execution of the algorithm. These plots agree with the claims in Proposition 1 and validate that the algorithm converges to a weight-balanced digraph with geometric convergence rate. In this particular example, with the choice of $\beta_j = 0.1$ or 0.5 or 0.9 , the rate discussed in the proof of Proposition 1 is given by $R_\infty(P(\beta_j)) = 0.1204, 0.5180$ and 0.2524 , respectively.

The final weight-balanced digraph W^* is the same for all three cases mentioned above (because all the β_j are identical — but equal to a different constant each time) and is given by

²A nonnegative matrix is said to be primitive if it is irreducible and has only one eigenvalue of maximum modulus [19].

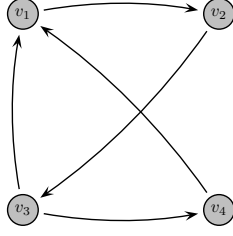


Fig. 1. A simple digraph which is weight-balanceable, but not bistoichiastic due to the absence of self-loops. This example is given in [18] in order to indicate that not all strongly connected digraphs are bistoichiastic.

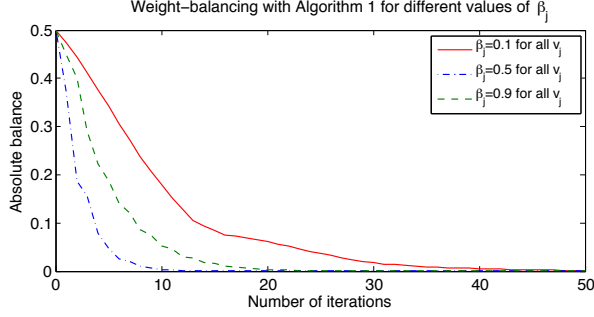


Fig. 2. Absolute balance for weight-balancing algorithm (*Algorithm 1*) for the digraph depicted in Figure 1.

$$W^* = \begin{bmatrix} 0 & 0 & 0.7143 & 0.7143 \\ 1.4286 & 0 & 0 & 0 \\ 0 & 1.4286 & 0 & 0 \\ 0 & 0 & 0.7143 & 0 \end{bmatrix}.$$

Remark 1: If $\beta_j = 1, \forall v_j \in \mathcal{V}$, then the weighted adjacency matrix P is not necessarily primitive and hence the algorithm does not converge to weights that form a weight-balanced digraph.

V. DISTRIBUTED ALGORITHM FOR BISTOCHASTIC MATRIX FORMATION

A. Description of the algorithm

An algorithm is proposed, herein called *Algorithm 2*, with which a bistoichiastic adjacency matrix is formed in a distributed fashion. One extra requirement for *Algorithm 2*, however, is to maintain column stochasticity of the weighted adjacency matrix $W[k]$ for all times k , so that it can be used for consensus problems without the need to first form the bistoichiastic matrix (the details can be found in [18]). More specifically, we obtain a sequence of column stochastic matrices $W[0], W[1], W[2], \dots, W[k]$ such that $\lim_{k \rightarrow \infty} W[k] = W$ is bistoichiastic and thus the iteration $x[k+1] = W[k]x[k]$, $x[0] = x_0$, reaches average consensus asymptotically [16].

Digraphs that are weight-balanceable do not necessarily admit a doubly stochastic assignment [18, *Theorem 4.1*]. However, if self-weights are added then any strongly connected graph is bistoichiastic after adding enough self-

loops [18, *Corollary 4.2*]. *Algorithm 2* overcomes this problem with the introduction of nonzero self weights w_{jj} at each node $v_j \in \mathcal{V}$, and by appropriately adjusting them in a distributed manner. *Algorithm 2* is described in detail below.

Algorithm 2 Bistoichiastic matrix formation algorithm

Input: A strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges (and no self-loops).

Initialization: Set $w_{lj}[0] = 1/(1 + \mathcal{D}_j^+)$, $\forall v_l \in \mathcal{N}_j^+ \cup \{j\}$.

Iterations: For $k = 0, 1, 2, \dots$, each node $v_j \in \mathcal{V}$ updates the weights w_{lj} , $v_l \in \mathcal{N}_j^+$, by performing the following steps:

1) It chooses $\beta_j[k]$ as follows:

$$\beta_j[k] = \begin{cases} \alpha_j \frac{1 - S_j^+[k]}{S_j^-[k] - S_j^+[k]}, & S_j^-[k] > S_j^+[k], \\ \alpha_j, & \text{otherwise,} \end{cases} \quad (5)$$

where $\alpha_j \in (0, 1)$.

2) It updates

$$w_{lj}[k+1] = w_{lj}[k] + \beta_j[k] \left(\frac{S_j^-[k]}{\mathcal{D}_j^+} - w_{lj}[k] \right), \quad (6)$$

for all $v_l \in \mathcal{N}_j^+$. [This is the same update as *Algorithm 1*, with the difference that the proportionality constant β_j can be adapted at each time step k , and is chosen so that $S_j^+[k+1] \leq 1$ (so as to ensure that w_{jj} can be chosen in Step 3 to be nonnegative and satisfy $w_{jj} + S_j^+[k+1] = 1$).]

3) Then, $w_{jj} \geq 0$ is assigned so that the weighted adjacency matrix retains its column stochasticity; i.e.,

$$w_{jj}[k+1] = 1 - \sum_{l \in \mathcal{N}_j^+} w_{lj}[k+1], \quad \forall v_j \in \mathcal{V}. \quad (7)$$

Proposition 2: If a digraph is strongly connected or is a collection of strongly connected digraphs, then *Algorithm 2* reaches a steady state weight matrix W^* that forms a bistoichiastic digraph. Furthermore, the weights of all edges in the graph are nonzero.

Proof: As before, all the outgoing links have the same weight, i.e., $w_{lv} = w_{lj}, \forall v_l \in \mathcal{N}_j^+$. Thus, the evolution of the weight $w_j \triangleq w_{lj}, \forall v_l \in \mathcal{N}_j^+$, can be written in matrix form as follows.

$$w[k+1] = P[k]w[k], w[0] = w_0, \quad (8)$$

where

$$P_{ji}[k] \triangleq \begin{cases} 1 - \beta_j[k] & \text{if } i = j, \\ \beta_j[k]/\mathcal{D}_j^+ & \text{if } i \in \mathcal{N}_j^-. \end{cases} \quad (9)$$

In order to make sure that the sum of each column can be made one by choosing a nonnegative self-weight w_{jj} , we need to establish (for all k) that $S_j^+[k+1] \leq 1$ or $w_{lj}[k+1] \leq 1/\mathcal{D}_j^+$, for all $v_l \in \mathcal{N}_j^+$. In our updates, there are two cases: (a) $S_j^-[k] \leq S_j^+[k]$, and (b) $S_j^-[k] > S_j^+[k]$; we analyze both cases below.

(a) When $S_j^-[k] \leq S_j^+[k]$, then $\beta_j[k]$ can be chosen to be

any value $\alpha_j, \alpha_j \in (0, 1)$. Then, the algorithm is equivalent to Algorithm 1.

(b) When $S_j^-[k] > S_j^+[k]$, then $w_{lj}[k+1], \forall v_l \in \mathcal{N}_j^+$, are increased. In order to avoid having $S_j^+[k+1] > 1$ or $w_{lj}[k+1] > 1/\mathcal{D}_j^+$ for any $v_l \in \mathcal{N}_j^+$, we choose $\beta_j[k]$ as shown in equation (5). In this case, for any initial $S_j^+[0]$ for which $S_j^-[k] > S_j^+[k]$, we obtain from (6) that

$$S_j^+[k+1] = S_j^+[k] + \alpha_j(1 - S_j^+[k]) = (1 - \alpha_j)S_j^+[k] + \alpha_j.$$

Hence, it is guaranteed that $S_j^+[k+1] < 1$, and as a result, $\beta_j[k] \neq 0$ for all k . It can be easily shown by perfect induction that after n steps (for which $S_j^-[k+r] > S_j^+[k+r]$ holds for all $r \in \mathbb{Z}_+, r \leq n$) we have

$$S_j^+[k+n] = 1 - (1 - \alpha_j)^n(1 - S_j^+[k]),$$

which approaches 1 as $n \rightarrow \infty$. Note that someone can choose $\alpha_j \in (0, 1)$ closer to 1 and guarantee that $\beta_j[k] > \epsilon, \epsilon > 0$, for all the iterations k . ■

Unlike the case of Algorithm 1, in Algorithm 2 we have a discrete-time switching dynamical system whose stability and convergence rate are not as easily characterized. As indicated by the various simulations we have tried (as well as the special case identified by the remark below), it is very likely that the rate of convergence of Algorithm 2 is geometric, however, this has not been formally established thus far.

Proposition 3: If a digraph is strongly connected or is a collection of strongly connected digraphs, Algorithm 2 with initial condition $w_{Lj}[0] = \frac{1}{m(1+\mathcal{D}_j^+)}$, $\forall v_L \in \mathcal{N}_j^+, m \geq |\mathcal{V}|$, reaches a steady state weight matrix W^* that forms a bistochastic digraph, with geometric convergence rate equal to $R_\infty(P) = -\ln \delta(P)$, where

$$P_{ji}[k] \triangleq \begin{cases} 1 - \alpha_j & \text{if } i = j, \\ \alpha_j/\mathcal{D}_j^+ & \text{if } i \in \mathcal{N}_j^-. \end{cases}$$

Furthermore, the weights of all edges in the graph are nonzero.

Proof: If it is possible for each node to know the number of nodes in the graph, or at least an upper bound, then we can set $w_{Lj}[0] = \frac{1}{m(1+\mathcal{D}_j^+)}$ for all $v_L \in \mathcal{N}_j^+$, where $m \geq |\mathcal{V}|$. Hence, we establish that $S_j^-[0] < 1$ and $S_j^+[0] < 1$ for all $v_j \in \mathcal{V}$. By equation (6) we have that

$$S_j^+[k+1] = (1 - \beta_j[k])S_j^+[k] + \beta_j[k]S_j^-[k],$$

which guarantees that $S_j^+[k] < 1$ and that $\beta_j[k] = \alpha_j$, for all k . The self-weights are chosen so that we make each column sum up to one (and consequently the row sums, once the algorithm converges). Therefore, with these initial conditions we can choose $\beta_j[k]$ to be constant (and equal to $\alpha_j \in (0, 1)$) throughout the execution of the algorithm, and hence, admit geometric convergence rate (because we are essentially back to Algorithm 1 with the only difference that we also calculate the value of the self-weight (equal to $1 - \mathcal{D}_j^+ w_j[k]$) each time). Therefore, Algorithm 2 has a geometric convergence rate $R_\infty(P) = -\ln \delta(P)$, as in Algorithm 1. ■

B. Illustrative Example

We consider a random strongly connected graph consisting of 50 nodes. The quantity of interest in this case is the absolute balance of the graph at time step k ; the absolute balance $Ab[k]$ is defined as $Ab[k] = \sum_{v_j \in \mathcal{V}} \left| 1 - \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k] \right|$. Since the weight matrix at each time step is column stochastic by construction, $Ab[k]$ effectively measures the distance of the weight matrix at time step k from a bistochastic matrix. As it can be seen from Figure 3, the algorithm asymptotically converges to a bistochastic adjacency matrix for different values of α_j (in the simulations α_j is chosen to be 0.9 or 0.5 or 0.1 for all $v_j \in \mathcal{V}$).

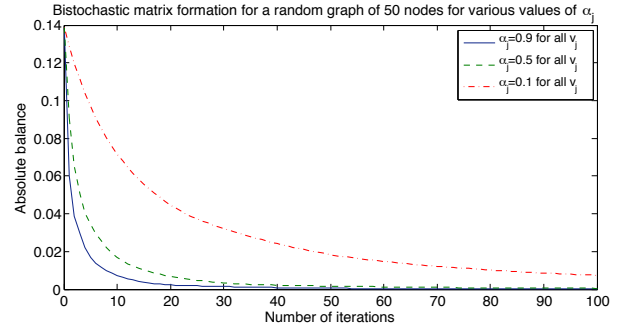


Fig. 3. Absolute balance for bistochastic formation algorithm (Algorithm 2) for different values of $\alpha_j \forall v_j \in \mathcal{V}$. It can be observed that Algorithm 2 converges to a bistochastic adjacency matrix asymptotically.

VI. COMPARISONS

A. Weight-Balanced Matrix Formation

Here we run the proposed algorithm in larger graphs (of size $n = 50$) and we compare the performance of our algorithm against two current state-of-the-art approaches: (a) the *imbalance-correcting algorithm* in [14] in which every node v_j adds all of its weight imbalance x_j to the outgoing node with the lowest weight w , and (b) the *weight balancing algorithm* in [15] in which each node v_j with positive imbalance $x_j > 0$, increases the weights of its outgoing links by an equal amount so that it becomes weight-balanced.

The proposed weight-balancing algorithm (Algorithm 1) shows geometric convergence and outperforms the algorithms suggested in [14], [15], as shown in Figure 4.

B. Bistochastic Matrix Formation

Here we run the algorithm for larger graphs (of size $n = 50$) and we compare the performance of our algorithm against a distributed algorithm suggested in [16] in which every node $v_j \in \mathcal{V}$ first chooses a particular weight $w_{jj}[k] \in (0, 1)$ for its self weight and it sets the weights of its outgoing links to be $w_{lj}[k] = c_{lj}(1 - w_{jj}[k])$, where $c_{lj} > 0, \forall v_l \in \mathcal{N}_j^+$, and $\sum_{v_l \in \mathcal{N}_j^+} c_{lj} = 1$. The suggested bistochastic formation algorithm (Algorithm 2) shows asymptotic convergence and slightly outperforms the algorithm suggested in [16] for $\alpha_j = 0.95, \forall v_j \in \mathcal{V}$, as shown in Figure 5.

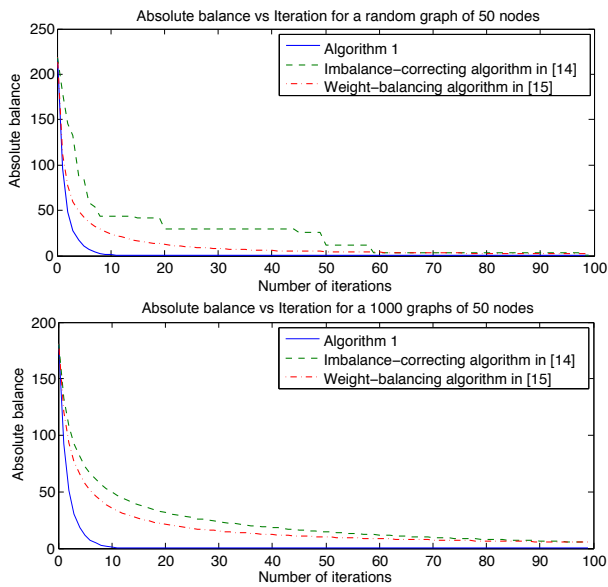


Fig. 4. *Top figure*: Absolute balance plotted against the number of iterations for a random graph of 50 nodes for the distributed weight-balancing (*Algorithm 1*) algorithm, the imbalance-correcting algorithm [14], and the weight-balancing algorithm proposed in [15]. *Bottom figure*: Absolute balance plotted against the number of iterations for a 1000 random graphs of 50 nodes for *Algorithm 1*, the imbalance-correcting algorithm [14], and the weight-balancing algorithm proposed in [15].

VII. CONCLUSIONS

In this paper we have developed two iterative algorithms: one for balancing a weighted digraph and one for forming a bistochastic adjacency matrix in a digraph. Both algorithms are distributed and asymptotic. The weight-balancing algorithm is shown to admit geometric convergence rate, while the second algorithm, a modification of the weight-balancing algorithm, leads to a bistochastic digraph with asymptotic convergence, and can be shown to admit geometric convergence rate for a certain set of initial values. The two algorithms are illustrated via examples and are shown to perform very well compared to existing algorithms.

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] 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, September 2004.
- [3] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.
- [4] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.
- [5] D. Angeli and P. Bliman, "Stability of leaderless discrete-time multi-agent systems," *Mathematics of Control, Signals, and Systems*, vol. 18, no. 4, pp. 293–322, 2006.
- [6] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.
- [7] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468–2481, October 2012.

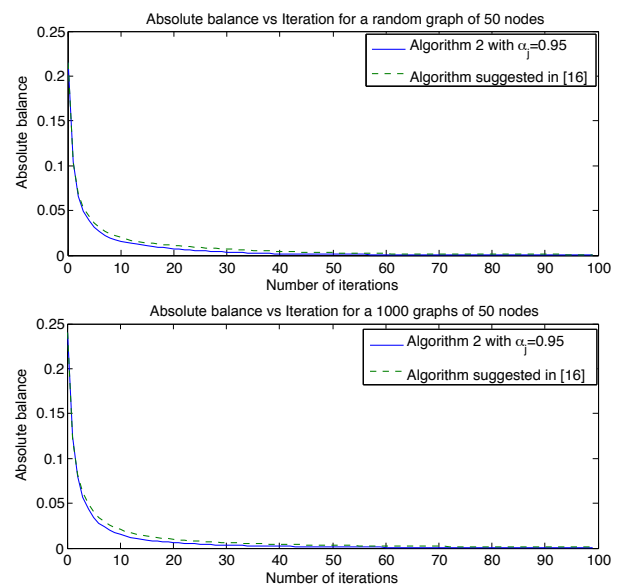


Fig. 5. *Top figure*: Absolute balance for bistochastic formation algorithms (*Algorithm 2*) and a distributed algorithm suggested in [16] for a random strongly connected graph consisting of 50 nodes. *Bottom figure*: Absolute balance plotted against the number of iterations for a 100 random strongly connected graphs of 50 nodes for the distributed bistochastic formation algorithm (*Algorithm 2*) and the distributed algorithm suggested in [16].

- [8] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Third International Symposium on Information Processing in Sensor Networks (IPSN)*, April 2004, pp. 20–27.
- [9] 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, August 2009, pp. 1156–1162.
- [10] C. N. Hadjicostis and T. Charalambous, "Asynchronous coordination of distributed energy resources for the provisioning of ancillary services," in *47th Annual Allerton Conference on Communication, Control, and Computing*, September 2011, pp. 1500–1507.
- [11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, March 2005, pp. 1653–1664.
- [12] 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, November 2009.
- [13] 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, August 2009.
- [14] B. Gharesifard and J. Cortés, "Distributed strategies for making a digraph weight-balanced," in *47th Annual Allerton Conference on Communication, Control, and Computing*, September 2009, pp. 771–777.
- [15] C. N. Hadjicostis and A. Rikos, "Distributed strategies for balancing a weighted digraph," in *20th Mediterranean Conference on Control Automation (MED)*, July 2012, pp. 1141–1146.
- [16] A. Dominguez-Garcia and C. N. Hadjicostis, "Distributed matrix scaling and application to average consensus in directed graphs," *IEEE Trans. on Automatic Control*, vol. 58, no. 3, pp. 667–681, March 2013.
- [17] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, March 2008.
- [18] B. Gharesifard and J. Cortés, "When does a digraph admit a doubly stochastic adjacency matrix?" in *American Control Conference (ACC)*, July 2010, pp. 2440–2445.
- [19] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, CB2 2RU, UK: Cambridge University Press, 1985.
- [20] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.