Distributed strategies for generating weight-balanced and doubly stochastic digraphs*

Bahman Gharesifard and Jorge Cortés †

Abstract

This paper deals with the design and analysis of dynamical systems on directed graphs (digraphs) that achieve weight-balanced and doubly stochastic assignments. Weight-balanced and doubly stochastic digraphs are two classes of digraphs that play an essential role in a variety of coordination problems, including formation control, agreement, and distributed optimization. We refer to a digraph as doubly stochasticable (weight-balanceable) if it admits a doubly stochastic (weight-balanced) adjacency matrix. This paper studies the characterization of both classes of digraphs, and introduces distributed dynamical systems to compute the appropriate set of weights in each case. It is known that semiconnectedness is a necessary and sufficient condition for a digraph to be weight-balanceable. The first main contribution is a characterization of doubly-stochasticable digraphs. As a by-product, we unveil the connection of this class of digraphs with weight-balanceable digraphs. The second main contribution is the synthesis of a distributed strategy running synchronously on a directed communication network that allows individual agents to balance their in- and out-degrees. We show that a variation of our distributed procedure over the mirror graph has a much smaller time complexity than the currently available centralized algorithm based on the computation of the graph cycles. The final main contribution is the design of two cooperative strategies for finding a doubly stochastic weight assignment. One algorithm works under the assumption that individual agents are allowed to add self-loops. For the case when this assumption does not hold, we introduce an algorithm distributed over the mirror digraph which allows the agents to compute a doubly stochastic weight assignment if the digraph is doubly stochasticable and announce otherwise if it is not. Various examples illustrate the results.

Keywords. Distributed dynamical systems, set-valued stability analysis, cooperative control, network design and communication, weight-balanced digraphs, doubly stochastic digraphs

*This work was supported in part by NSF Awards CCF-0917166 and CMMI-0908508. This manuscript substantially improves and extends the preliminary conference versions presented as [12] and [13].

†B. Gharesifard and Jorge Cortés are with the Department of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093-0411, United States, Tel.: +1 858-822-7930, Fax. +1 858-822-3107, {bgharesifard,cortes}@ucsd.edu

1 Introduction

In the last years there has been considerable interest in understanding the underpinnings of collective behavior from a dynamical systems perspective. A variety of phenomena from biology and physics have been carefully studied, and more are coming into light. A few examples of a vast literature include oscillator synchronization [35, 36, 37], self-organization in biological systems [7, 16, 30], and animal grouping and aggregation [17, 27]. In the study of collective behavior, a major concern is the modeling of the interactions between individual agents and the understanding on how local interconnections give rise to global emergent behavior. Such coordination problems have also become a major research area in engineering because of the connections with distributed robotics, networked systems, and autonomy, see e.g., [6, 25, 33] and references therein.

This paper is a contribution to this buoying field. From a systems and controls perspective, one of the main objectives is the design of dynamical systems that are distributed over a given interaction topology and whose performance guarantees can be formally established with regards to the task at hand. In both analysis and design, the interaction topology (or graph) of the underlying network is a key element as it determines the information available to each individual. Typically, directed interaction topologies (where interactions among agents are unidirectional) pose greater technical challenges.

In this paper, we study dynamical systems on two important classes of directed graphs, weight-balanced and doubly stochastic digraphs. A digraph is weight-balanced if, at each vertex, the sum of the weights of the incoming edges is equal to the sum of the weights of the outgoing edges. A digraph is doubly stochastic if it is weight-balanced and these sums at each vertex are equal to one. The notion of weight-balanced digraph is key in establishing convergence results of distributed algorithms for average-consensus [28, 29] and consensus on general functions [9] via Lyapunov stability analysis. Weight-balanced digraphs also appear in the design of leader-follower strategies under time delays [19], virtual leader strategies under asymmetric interactions [34] and stable flocking algorithms for agents with significant inertial effects [21]. In [18], a traffic-flow problem is introduced with n junction and m one-way streets with the goal of ensuring a smooth traffic flow. It is shown that the problem can be reduced to computing weights on the edges of the associated digraph so that it is weight-balanced. Furthermore, necessary and sufficient conditions are given for a digraph to be weight-balanced and a centralized algorithm is presented for computing the weight on each edge. Doubly stochastic digraphs also play a key role in networked control problems. Examples include distributed averaging [6, 29, 33, 40] and distributed convex optimization [20, 26, 41]. Convergence in gossip algorithms also relies on the structure of doubly stochastic digraphs, see [5, 22].

Because of the numerous algorithms available in the literature that use weight-balanced and doubly stochastic interaction topologies, it is important to develop distributed strategies that allow individual agents to find the appropriate weight assignments, i.e., to balance their in- and out-degrees, so that the overall interaction digraph is weight-balanced or doubly stochastic. In particular, we are interested in designing discrete-time dynamical systems that can be run on the directed communication network which, in finite time, converge to a weight-

balanced/doubly stochastic assignment. As a necessary step towards this goal, it is an important research question to characterize when a digraph can be given an edge weight assignment that makes it belong to either category. Our focus in this paper is on nonzero weight assignments. In addition to its theoretical interest, the consideration of nonzero weight assignments is also relevant from a practical perspective, as the use of the maximum number of edges leads to higher algebraic connectivity [39], which in turn affects positively the rate of convergence [5, 8, 14, 22] of the algorithms typically executed over doubly stochastic digraphs. Alternative versions of the problem, where some edge weights are allowed to be zero, are also worth exploring, although we do not consider them here. From a distributed perspective, such problems pose nontrivial challenges, as individual agents would need to concurrently implement a procedure, see e.g., [10], to determine if severing a particular edge (i.e., setting its weight to zero) or set of them disconnects the overall digraph. Such procedures would necessarily require information beyond the immediate neighborhood of the agents.

Our contributions are threefold. First, we obtain a characterization of doubly stochasticable digraphs (a characterization of weight-balanceable digraphs is already available in the literature, cf. [18]). As a by-product of our study, we demonstrate that the set of doubly stochasticable digraphs can be generated by a special subset of weight-balanced digraphs. Second, we develop two discrete-time set-valued dynamical systems on the directed communication network that converge to a weight-balanced digraph in finite time. The `imbalance-correcting algorithm` is a synchronous distributed strategy on a digraph in which each agent provably balances its in- and out-degrees in finite time. In this algorithm, each individual agent can send a message to one of its out-neighbors and receive a message from its in-neighbors. The `mirror imbalance-correcting algorithm` is a provably correct distributed strategy over the mirror digraph whose time complexity is much smaller than that of the existing centralized strategy of [18] based on the computation of all cycles of the digraph. Third, we synthesize discrete-time dynamical systems to construct doubly stochastic adjacency matrices. The `imbalance-correcting algorithm with self-loop addition` achieves this task under the assumption that individual agents can add self-loops to the structure of the digraph. If this is not allowed, we introduce the `load-pushing algorithm`, which is a strategy distributed over the mirror digraph that allows agents to: (i) identify if their digraph is doubly stochasticable and, if this is the case, (ii) find a set of weights that makes the digraph doubly stochastic. The algorithm relies on the notion of Doubly-Stochastic-index (DS-index for short) of a strongly connected doubly stochasticable digraph and its design and correctness analysis draw substantially on distributed solutions to the maximum flow problem [1, 32]. Our results are applicable to any of the scenarios described above (distributed formation, consensus, flocking, gossip, or optimization). Our strategies can be executed by the networked system prior to any other coordination algorithm that requires the interaction topology to be weight-balanced or doubly stochastic.

This paper is organized as follows. Section 2 presents some mathematical preliminaries. Section 3 gives the necessary and sufficient conditions for the existence of a doubly stochastic adjacency matrix assignment for a given digraph. Section 4 introduces two weight-balancing algorithms that allow each agent to balance its in- and

out-degrees. We characterize their distributed character as well as the convergence and complexity properties. In Section 5, we discuss the problem of designing cooperative strategies that allow agents to find a doubly stochastic edge weight assignment. Finally, Section 6 contains our conclusions and ideas for future work.

2 Mathematical preliminaries

We adopt some basic notions from [2, 6, 11]. A *directed graph*, or simply *digraph*, is a pair $G = (V, E)$, where V is a finite set called the vertex set and $E \subseteq V \times V$ is the edge set. If $|V| = n$, i.e., the cardinality of V is $n \in \mathbb{Z}_{>0}$, we say that G is of order n which, unless otherwise noted, is the standard assumption throughout the paper. A digraph $G_1 = (V_1, E_1)$ is a subdigraph of $G_2 = (V_2, E_2)$ if $V_1 \subset V_2$ and $E_1 \subset E_2$. An edge $(u, v) \in E$ is *incident away from* u (or an *out-edge* of u) and *incident toward* v (or an *in-edge* of v), and we call u an *in-neighbor* of v and v an *out-neighbor* of u. We denote the set of in-neighbors and out-neighbors of v, respectively, with $\mathscr{N}^{\text{in}}(v)$ and $\mathscr{N}^{\text{out}}(v)$. The *in-degree* and *out-degree* of v, denoted $d_{\text{in}}(v)$ and $d_{\text{out}}(v)$, are the number of in-neighbors and out-neighbors of v, respectively. We call a vertex v *isolated* if it has zero in- and out-degrees. An *undirected graph*, or simply *graph*, is a pair $G = (V, E)$, where V is a finite set called the vertex set and the edge set E consists of unordered pairs of vertices. In a graph, neighboring relationships are always bidirectional, and hence we simply use the terms neighbor, degree, etc. for the notions introduced above. A graph $G_1 = (V_1, E_1)$ is a subgraph of $G_2 = (V_2, E_2)$ if $V_1 \subset V_2$ and $E_1 \subset E_2$. A graph is *regular* if each vertex has the same number of neighbors. The *union* $G_1 \cup G_2$ of digraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined by $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$. The intersection of two digraphs can be defined similarly. A digraph G is *generated* by a set of digraphs $G_1, \ldots, G_m$ if $G = G_1 \cup \cdots \cup G_m$. We let $E^- \subseteq V \times V$ denote the set obtained by changing the order of the elements of E, i.e., $(v, u) \in E^-$ iff $(u, v) \in E$. The digraph $\overline{G} = (V, E \cup E^-)$ is the *mirror* of G.

A *weighted digraph* is a triplet $G = (V, E, A)$, where (V, E) is a digraph and $A \in \mathbb{R}^{n \times n}_{\geq 0}$ is the *adjacency matrix*. We denote the entries of A by a_{ij}, where $i, j \in \{1, \ldots, n\}$. The adjacency matrix has the property that the entry $a_{ij} > 0$ if $(v_i, v_j) \in E$ and $a_{ij} = 0$, otherwise. If a matrix A satisfies this property, we say that A is a *weight assignment* of the digraph $G = (V, E)$. Note that any digraph can be trivially seen as a weighted digraph by assigning weight 1 to each one of its edges. We refer to this as the *trivial weight assignment*. We will find it useful to extend the definition of union of digraphs to weighted digraphs. The union $G_1 \cup G_2$ of the weighted digraphs $G_1 = (V_1, E_1, A_1)$ and $G_2 = (V_2, E_2, A_2)$ is defined by $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2, A)$, where

$$a_{ij} = \begin{cases} (a_1)_{ij} & (v_i, v_j) \in E_1 \setminus E_2, \\ (a_2)_{ij} & (v_i, v_j) \in E_2 \setminus E_1, \\ (a_1)_{ij} + (a_1)_{ij} & (v_i, v_j) \in E_1 \cap E_2. \end{cases}$$

For a weighted digraph, the weighted out-degree, weighted in-degree and imbalance of v_i, $i \in \{1, \ldots, n\}$, are

respectively,

$$d^{\mathrm{w}}_{\mathrm{out}}(v_i) = \sum_{j=1}^{n} a_{ij}, \qquad d^{\mathrm{w}}_{\mathrm{in}}(v_i) = \sum_{j=1}^{n} a_{ji}, \qquad \omega(v_i) = d^{\mathrm{w}}_{\mathrm{in}}(v_i) - d^{\mathrm{w}}_{\mathrm{out}}(v_i). \tag{1}$$

It is worth noticing that the imbalances across the graph always satisfy

$$\sum_{i=1}^{n} \omega(v_i) = 0. \tag{2}$$

2.1 Graph connectivity notions

A *directed path* in a digraph, or in short path, is an ordered sequence of vertices so that any two consecutive vertices in the sequence are an edge of the digraph. A *cycle* in a digraph is a directed path that starts and ends at the same vertex and has no other repeated vertex. Hence, a self-loop is a cycle while an isolated vertex is not. Two cycles are *disjoint* if they do not have any vertex in common. We denote by G_{cyc} a union of some disjoint cycles of G (note that G_{cyc} can be just one cycle).

A digraph is *strongly connected* if there is a path between each pair of distinct vertices and is *strongly semiconnected* if the existence of a path from v to w implies the existence of a path from w to v, for all $v, w \in V$. Clearly, strong connectedness implies strong semiconnectedness, but the converse is not true. The *strongly connected components* of a directed graph G are its maximal strongly connected subdigraphs.

2.2 Basic notions from linear algebra

A matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ is *weight-balanced* if $\sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji}$, for all $i \in \{1,\dots,n\}$. A matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ is *row-stochastic* if each of its rows sums 1. One can similarly define a column-stochastic matrix. We denote the set of all row-stochastic matrices on $\mathbb{R}^{n\times n}_{\geq 0}$ by $\mathrm{RStoc}(\mathbb{R}^{n\times n}_{\geq 0})$. A non-zero matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ is *doubly stochastic* if it is both row-stochastic and column-stochastic. A matrix $A \in \{0,1\}^{n\times n}$ is a *permutation matrix*, where $n \in \mathbb{Z}_{\geq 1}$, if A has exactly one entry 1 in each row and each column. A matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ is *irreducible* if, for any nontrivial partition $J \cup K$ of the index set $\{1,\dots,n\}$, there exist $j \in J$ and $k \in K$ such that $a_{jk} \neq 0$. We denote by $\mathrm{Irr}(\mathbb{R}^{n\times n}_{\geq 0})$ the set all irreducible matrices on $\mathbb{R}^{n\times n}_{\geq 0}$. Note that a weighted digraph G is strongly connected if and only if its adjacency matrix is irreducible [2].

One can extend the adjacency matrix associated to a disjoint union of cycles G_{cyc} of G to a matrix $A_{\mathrm{cyc}} \in \mathbb{R}^{n\times n}$ by adding zero rows and columns for the vertices of G that are not included in G_{cyc}. Note that the matrix A_{cyc} is the adjacency matrix for a subdigraph of G. We call A_{cyc} the *extended adjacency matrix* associated to G_{cyc}. The following result establishes the relationship between cycles of G and permutation matrices.

Lemma 2.1 (Cycles and permutation matrices)**.** *The extended adjacency matrix associated to a union of disjoint cycles G_{cyc} is a permutation matrix if and only if G_{cyc} contains all the vertices of G.*

Proof. It is clear that if G_{cyc} is a union of some disjoint cycles and contains all the vertices of G, then the adjacency matrix associated to G_{cyc} is a permutation matrix. Conversely, suppose that G_{cyc} does not contain one of the vertices of G. Then the adjacency matrix associated to G_{cyc} has a zero row and thus is not a permutation matrix. □

2.3 Weight-balanced and doubly stochastic digraphs

A weighted digraph G is *weight-balanced* (resp. *doubly stochastic*) if its adjacency matrix is weight-balanced (resp. doubly stochastic). Note that G is weight-balanced if and only if $d^{\text{w}}_{\text{out}}(v) = d^{\text{w}}_{\text{in}}(v)$, for all $v \in V$. A digraph is called *weight-balanceable* (resp. *doubly stochasticable*) if it admits a weight-balanced (resp. doubly stochastic) adjacency matrix. The following two results characterize when a digraph is weight-balanceable.

Theorem 2.2 ([18]). *A digraph $G = (V,E)$ is weight-balanceable if and only if the edge set E can be decomposed into k subsets $E_1, \ldots, E_k$ such that*

(i) $E = E_1 \cup E_2 \cup \ldots \cup E_k$ *and*

(ii) each $G = (V,E_i)$, $i = \{1, \ldots, k\}$, is weight-balanceable.

Theorem 2.3 ([18]). *Let $G = (V,E)$ be a digraph. The following statements are equivalent:*

(i) G is weight-balanceable,

(ii) Every element of E lies in a cycle,

(iii) G is strongly semiconnected.

The proofs of these theorems are constructive but rely on the computation of all the cycles of the digraph. The basic idea is that if one can find all the cycles for a given strongly semiconnected digraph, then, by taking the weighted union of these cycles (i.e., by assigning to each edge the number of times it appears in the cycles), one arrives at a weight-balanced assignment, see [18].

Note that Theorem 2.3 implies that any strongly semiconnected digraph can be generated by the cycles contained in it. Therefore, it makes sense to define a minimal set of cycles with this property. This motivates the introduction of the following concept.

Definition 2.4 (Principal cycle set). *Let G be a strongly semiconnected digraph and let $\mathscr{C}(G)$ denote the set of all subdigraphs of G that are either isolated vertices, cycles of G, or a union of disjoint cycles of G. A set $P \subseteq \mathscr{C}(G)$ is a* principal cycle set *of G if its elements generate G, and there is no subset of $\mathscr{C}(G)$ with strictly smaller cardinality that satisfies this property. We denote by $p(G)$ the cardinality of P.*

Note that there might exist more than one principal cycle set. However, by definition, the cardinalities of all principal cycle sets are the same, hence the notation $p(G)$. Note that a cycle, or a union of disjoint cycles, that

contains all the vertices has the maximum number of edges that an element of $\mathscr{C}(G)$ can have. Thus these elements are the obvious candidates for constructing a principal cycle set. Principal cycle sets give rise to weight-balanced assignments, as we state next.

Lemma 2.5 (Weight-balancing via principal cycle sets)**.** *Let G be a strongly semiconnected digraph. Then, the union of the elements of a principal cycle set P of G, considered as subdigraphs with trivial weight assignment, gives a set of positive integer weights which make the digraph weight-balanced.*

Proof. Since each element of a principal cycle P is either an isolated vertex, a cycle, or union of disjoint cycles, it is weight-balanced. By definition, G can be written as the union of the elements of P. Thus by Theorem 2.2, the weighted union of the elements of P gives a set of weights makes the digraph G weight-balanced. □

In general, the assignment in Lemma 2.5 uses fewer number of cycles than the ones used in Theorem 2.3.

2.4 Discrete set-valued analysis

Here, we provide a brief exposition of useful concepts from discrete-time set-valued dynamical systems following [6, 23]. For $X \subseteq \mathbb{R}^n$, let $F : X \rightrightarrows X$ denote a set-valued map that takes a point in X to a subset $F(x)$ of X. The map F is non-empty if $F(x) \neq \emptyset$, for all $x \in X$. A point $x^* \in X$ is a *fixed point* of F if $x^* \in F(x^*)$. An *evolution* of F on X is any trajectory $\gamma : \mathbb{Z}_{\geq 0} \to X$ such that

$$\gamma(k+1) \in F(\gamma(k)), \quad \text{for all } k \in \mathbb{Z}_{\geq 0}.$$

The map F is *closed* at $x \in X$ if, for any two convergent sequences $\{x_k\}_{k=0}^{\infty}$ and $\{y_k\}_{k=0}^{\infty}$, with $\lim_{k\to\infty} x_k = x$, $\lim_{k\to\infty} y_k = y$, and $y_k \in F(x_k)$, for all $k \in \mathbb{Z}_{\geq 0}$, we have $y \in F(x)$. The map F is closed on X if it is closed at x, for all $x \in X$. A set $W \subset X$ is *weakly positively invariant* with respect to F if for any $x \in W$ there exists $y \in W$ such that $y \in F(x)$ and *strongly positively invariant* with respect to F if $F(x) \subset W$, for all $x \in W$. Finally, a continuous function $V : X \to \mathbb{R}$ is called *non-increasing* along F in $W \subset X$ if $V(y) \leq V(x)$, for all $x \in W$ and $y \in F(x)$. Equipped with these tools, one can formulate the following set-valued version of the LaSalle invariance principle, which will be most useful in the developments later.

Theorem 2.6 (LaSalle invariance principle for discrete-time set-valued dynamical systems)**.** *Let $F : X \rightrightarrows X$ be a set-valued map on $X \subset \mathbb{R}^n$ and let $W \subset X$ be a closed and strongly positively invariant with respect to F. Suppose F is non-empty and closed on W and all evolutions of F with initial condition in W are bounded. Let $V : X \to \mathbb{R}$ be continuous and non-increasing function along F on W. Then, any evolution of F with initial condition in W approaches a set of the form $S \cap V^{-1}(c)$, where $c \in \mathbb{R}$ and S is the largest weakly positively invariant set contained in $\{x \in W \mid \text{there exists } y \in F(x) \text{ such that } V(x) = V(y)\}$.*

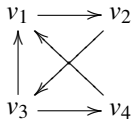

Figure 1: A weight-balanceable digraph for which their exists no doubly stochastic adjacency assignment.

3 When does a digraph admit a doubly stochastic weight assignment?

Our main goal in this section is to obtain necessary and sufficient conditions that characterize when a digraph is doubly stochasticable. This characterization is a necessary step before addressing in later sections the design of distributed dynamical systems that find doubly stochastic weight assignments. As an intermediate step of this characterization, we will also find it useful to study the relationship between weight-balanced and doubly stochastic digraphs.

Note that strong semiconnectedness is a necessary, and sufficient, condition for a digraph to be weight-balanceable. All doubly stochastic digraphs are weight-balanced; thus a necessary condition for a digraph to be doubly stochasticable is strong semiconnectedness. Moreover, weight-balanceable digraphs that are doubly stochasticable do not have any isolated vertices. However, none of these conditions are sufficient. A simple example illustrates this. Consider the digraph shown in Figure 1. Note that this digraph is strongly connected; thus there exists a set of positive weights which makes the digraph weight-balanced. However, there exists no set of nonzero weights that makes this digraph doubly stochastic. Suppose

$$A = \begin{pmatrix} 0 & \alpha_1 & 0 & 0 \\ 0 & 0 & \alpha_2 & 0 \\ \alpha_3 & 0 & 0 & \alpha_4 \\ \alpha_5 & 0 & 0 & 0 \end{pmatrix},$$

where $\alpha_i \in \mathbb{R}_{>0}$, for all $i \in \{1,\dots,5\}$, is a doubly stochastic adjacency assignment for this digraph. Then a simple computation shows that the only solution that makes the digraph doubly stochastic is by choosing $\alpha_3 = 0$, which is not possible by assumption. Thus this digraph is not doubly stochasticable.

The following result will simplify our analysis by allowing us to restrict our attention to strongly connected digraphs.

Lemma 3.1 (Strongly connected components of a doubly stochasticable digraph)**.** *A strongly semiconnected digraph is doubly stochasticable if and only if all of its strongly connected components are doubly stochasticable.*

Proof. Let G_1 and G_2 be two strongly connected components of the digraph. Note that there can be no edges from $v_1 \in G_1$ to $v_2 \in G_2$ (or vice versa). If this were the case, then the strong semiconnectedness of the digraph would imply that there is a path from v_2 to v_1 in the digraph, and hence $G_1 \cup G_2$ would be strongly connected, contradicting the fact that G_1 and G_2 are maximal. Therefore, the adjacency matrix of the digraph is a block-

diagonal matrix, where each block corresponds to the adjacency matrix of a strongly connected component, and the result follows. □

As a result of Lemma 3.1, we are interested in characterizing the class of strongly connected digraphs which are doubly stochasticable.

3.1 The relationship between weight-balanced and doubly stochastic adjacency matrices

As an intermediate step of the characterization of doubly stochasticable digraphs, we will find it useful to study the relationship between weight-balanced and doubly stochastic digraphs. The example in Figure 1 underscores the importance of characterizing the set of weight-balanceable digraphs that are also doubly-stochasticable.

We start by introducing the *row-stochastic normalization map* $\phi : \mathrm{Irr}(\mathbb{R}^{n\times n}_{\geq 0}) \to \mathrm{RStoc}(\mathbb{R}^{n\times n}_{\geq 0})$ defined by

$$\phi : (a_{ij}) \mapsto \left(\frac{a_{ij}}{\sum_{l=1}^{n} a_{il}} \right).$$

Note that, for $A \in \mathrm{Irr}(\mathbb{R}^{n\times n}_{\geq 0})$, $\phi(A)$ is doubly stochastic if and only if

$$\sum_{i=1}^{n} \frac{a_{ij}}{\sum_{l=1}^{n} a_{il}} = 1,$$

for all $j \in \{1,\dots,n\}$. The following result characterizes when the digraph associated with an irreducible weight-balanced adjacency matrix is doubly stochasticable.

Theorem 3.2 (Weight-balanced and doubly stochasticable digraphs)**.** *Let $A \in \mathrm{Irr}(\mathbb{R}^{n\times n}_{\geq 0})$ be an adjacency matrix associated to a strongly connected weight-balanced digraph. Then $\phi(A)$ is doubly stochastic if and only if $\sum_{l=1}^{n} a_{il} = C$, for all $i \in \{1,\dots,n\}$, for some $C \in \mathbb{R}_{>0}$.*

Proof. The implication from right to left is immediate. Suppose then that A is associated to a strongly connected weight-balanced digraph. Then we need to show that if A satisfies the following set of equations

$$\sum_{l=1}^{n} a_{jl} = \sum_{l=1}^{n} a_{lj}, \qquad \sum_{i=1}^{n} \frac{a_{ij}}{\sum_{l=1}^{n} a_{il}} = 1,$$

for all $j \in \{1,\dots,n\}$, there exists $C \in \mathbb{R}_{>0}$ such that $\sum_{l=1}^{n} a_{il} = C$, for all $i \in \{1,\dots,n\}$. Let $C_k = \sum_{l=1}^{n} a_{kl}$, $k \in \{1,\dots,n\}$. Then the doubly stochastic conditions can be written as

$$\frac{a_{1j}}{C_1} + \frac{a_{2j}}{C_2} + \cdots + \frac{a_{nj}}{C_n} = 1, \tag{3}$$

for all $j \in \{1,\dots,n\}$. Note that $C_k \neq 0$, for all $k \in \{1,\dots,n\}$, since A is irreducible. By the weight-balanced assumption, we have

$$a_{1j} + a_{2j} + \cdots + a_{nj} = C_j,$$

for all $j \in \{1,\dots,n\}$. Thus

$$\frac{a_{1j}}{C_j} + \frac{a_{2j}}{C_j} + \cdots + \frac{a_{nj}}{C_j} = 1, \tag{4}$$

From (3) and (4), we have

$$a_{1j}\left(\frac{1}{C_1} - \frac{1}{C_j}\right) + \cdots + a_{nj}\left(\frac{1}{C_n} - \frac{1}{C_j}\right) = 0, \tag{5}$$

for all $j \in \{1,\dots,n\}$. Suppose that, up to rearranging,

$$C_1 = \min_k \{C_k \mid k \in \{1,\dots,n\}\},$$

and, $0 < C_1 < C_i$, for all $i \in \{2,\dots,n\}$. Then (5) gives

$$a_{21}\left(\frac{1}{C_2} - \frac{1}{C_1}\right) + \cdots + a_{n1}\left(\frac{1}{C_n} - \frac{1}{C_1}\right) = 0;$$

thus $a_{j1} = 0$, for all $j \in \{2,\dots,n\}$, which contradicts the irreducibility assumption. If the set $\{C_k\}_{k=1}^n$ has more than one element giving the minimum, the proof follows a similar argument. Suppose

$$C_1 = C_2 = \min_k \{C_k \mid k \in \{1,\dots,n\}\},$$

and suppose that $0 < C_1 = C_2 < C_i$, for all $i \in \{3,\dots,n\}$. Then we have

$$a_{31}\left(\frac{1}{C_3} - \frac{1}{C_1}\right) + \cdots + a_{n1}\left(\frac{1}{C_n} - \frac{1}{C_1}\right) = 0,$$

and

$$a_{32}\left(\frac{1}{C_3} - \frac{1}{C_2}\right) + \cdots + a_{n2}\left(\frac{1}{C_n} - \frac{1}{C_2}\right) = 0,$$

and thus $a_{j1} = 0 = a_{j2}$, for all $j \in \{3,\dots,n\}$, which contradicts the irreducibility assumption. The same argument holds for an arbitrary number of minima. □

Corollary 3.3 (Self-loop addition makes a digraph doubly stochasticable)**.** *Any strongly connected digraph is doubly stochasticable after adding enough number of self-loops.*

Proof. Any strongly connected digraph is weight-balanceable. The result follows from noting that, for any weight-balanced matrix, it is enough to add self-loops with appropriate weights to the vertices of the digraph to make the conditions of Theorem 3.2 hold. □

Regular undirected graphs trivially satisfy the conditions of Theorem 3.2 and hence the following result.

Corollary 3.4 (Undirected regular graphs)**.** *All undirected regular graphs are doubly stochasticable.*

We capture the essence of Theorem 3.2 with the following definition.

Definition 3.5 (C-regularity)**.** *Let $G = (V,E)$ be a strongly connected digraph and let A be a weight-balanced adjacency matrix which satisfies the conditions of Theorem 3.2 with $C \in \mathbb{R}_{>0}$. Then we refer to $G = (V,E,A)$ as a* C-regular digraph.

3.2 Necessary and sufficient conditions for doubly stochasticability

In this section, we provide a characterization of the structure of digraphs that are doubly stochasticable. We start by giving a sufficient condition for doubly stochasticability.

Proposition 3.6 (Sufficient condition for doubly stochasticability)**.** *A strongly connected digraph G is doubly stochasticable if there exists a set $\{G^i_{\text{cyc}}\}_{i=1}^{\xi} \subseteq \mathscr{C}(G)$, where $\xi \geq p(G)$, that generates G and such that, for each $i \in \{1,\dots,\xi\}$, G^i_{cyc} contains all the vertices of G.*

Proof. Suppose $G = \cup_{i=1}^{\xi} G^i_{\text{cyc}}$, where $G^i_{\text{cyc}} \in \mathscr{C}(G)$ contain all the vertices, for all $i \in \{1,\dots,\xi\}$. Consider the adjacency matrix

$$A = \sum_{i=1}^{\xi} A^i_{\text{cyc}},$$

where A^i_{cyc} is the extended adjacency matrix associated to G^i_{cyc}. Note that A is weight-balanced and satisfies the condition of Theorem 3.2 (the sum of each row is equal to ξ). Thus G is doubly stochasticable. □

Proposition 3.6 suggests the definition of the following notion. Given a strongly connected digraph G that satisfies the sufficient condition of Proposition 3.6, $DS(G) \subseteq \mathscr{C}(G)$ is a *DS-cycle set* of G if all its elements contain all the vertices of G, they generate G, and there is no subset of $\mathscr{C}(G)$ with strictly smaller cardinality that satisfies these properties. The cardinality of any DS-cycle set of G is the *DS-index* of G, denoted $\text{ds}(G)$. By definition, $\text{ds}(G) \geq p(G)$, where recall that $p(G)$ denotes the cardinality of any principal cycle set.

The following result states that the condition of Proposition 3.6 is actually necessary for a digraph to be doubly stochasticable.

Proposition 3.7 (Necessary condition for doubly stochasticability)**.** *Let G be a strongly connected digraph. Suppose that one can assign a doubly stochastic adjacency matrix A to G. Then G satisfies the condition of Proposition 3.6 and*

$$A = \sum_{i=1}^{\xi} \lambda_i A^i_{\text{cyc}},$$

where

- $\{\lambda_i\}_{i=1}^{\xi} \subset \mathbb{R}_{>0}$, $\sum_{i=1}^{\xi} \lambda_i = 1$, *and* $\xi \geq \text{ds}(G)$.
- A^i_{cyc}, $i \in \{1,\dots,\xi\}$, *is the extended adjacency matrix associated to an element of $\mathscr{C}(G)$ that contains all the vertices.*

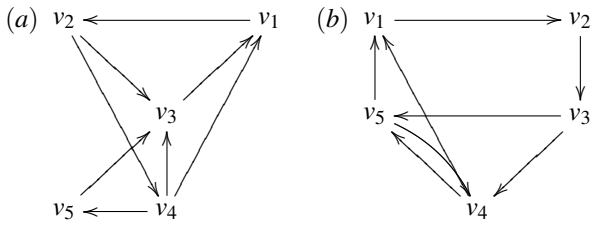

Figure 2: The digraph of Examples 3.9 and 3.10 are shown in plots (a) and (b), respectively.

Proof. Let A be a doubly stochastic matrix associated to G. By the Birkhoff–von Neumann theorem [3], a square matrix is doubly stochastic if and only if it is a convex combination of permutation matrices. Therefore,

$$A = \sum_{i=1}^{n!} \bar{\lambda}_i A^i_{\text{perm}},$$

where $\bar{\lambda}_i \in \mathbb{R}_{\geq 0}$, $\sum_{i=1}^{n!} \bar{\lambda}_i = 1$, and A^i_{perm} is a permutation matrix for each $i \in \{1,\dots,n!\}$. By Lemma 2.1, for all $\bar{\lambda}_i > 0$, one can associate to the corresponding A^i_{perm} a union of disjoint cycles that contains all the vertices. Thus each A^i_{perm} is an extended adjacency matrix associated to an element of $\mathscr{C}(G)$ that contains all the vertices. This proves that there exists a set $\{G^i_{\text{cyc}}\}_{i=1}^{\xi} \subseteq \mathscr{C}(G)$, where $\xi \geq p(G)$, that generates G; thus G satisfies the condition of Proposition 3.6. Let us rename all the nonzero coefficients $\bar{\lambda}_i > 0$ to λ_i. In order to complete the proof, we need to show that at least $\text{ds}(G)$ of the λ_i's are nonzero. Suppose otherwise. Since each A^i_{perm} with nonzero coefficient is associated to an element of $\mathscr{C}(G)$, this means that the digraph G can be generated by fewer elements than $\text{ds}(G)$, which would contradict the definition of DS-index. □

Note that Propositions 3.6 and 3.7 fully characterize the set of doubly stochasticable strongly connected digraphs. We gather this result in the following corollary.

Corollary 3.8 (Necessary and sufficient condition for doubly stochasticability)**.** *A strongly connected digraph G is doubly stochasticable if and only if there exists a set $\{G^i_{\text{cyc}}\}_{i=1}^{\xi} \subseteq \mathscr{C}(G)$, where $\xi \geq \text{ds}(G)$, that generates G and such that G^i_{cyc} contains all the vertices of G, for each $i \in \{1,\dots,\xi\}$.*

If a doubly stochasticable digraph is not strongly connected, one can use these results for Corollary 3.8 on each strongly connected component.

Example 3.9 (Weight-balanceable, not doubly stochasticable digraph)**.** Consider the digraph G shown in Figure 2(a). It is shown in [12] that there exists a set of weights which makes this digraph weight-balanced. We show that the digraph is not doubly stochasticable. Suppose there exists a union of disjoint cycles containing all the vertices. Since the edge (v_2, v_3) only appears in the cycle $G_{\text{cyc}} = \{v_1, v_2, v_3\}$, one element in such a union must contain this cycle. But then it is impossible for this element to also contain v_4 and v_5, as every cycle containing v_4 and v_5 also contains at least one of the vertices $\{v_1, v_2, v_3\}$. Thus by Corollary 3.8, there exists no doubly stochastic adjacency assignment for this digraph. One can verify this by trying to find such assignment explicitly,

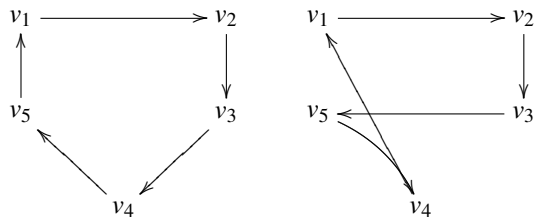

Figure 3: The only principal cycle set for the digraph of Example 3.10 contains the above cycles.

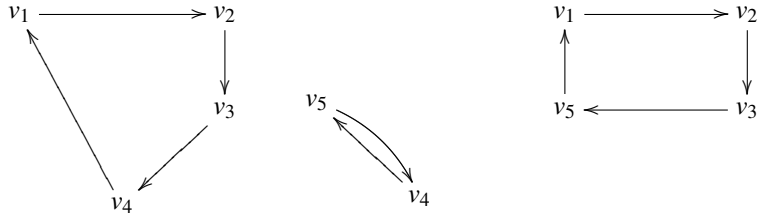

Figure 4: Cycles of the digraph G of Example 3.10 which are not in the principal cycle set.

i.e., by seeking $\alpha_i \in \mathbb{R}_{>0}$, where $i \in \{1,\dots,8\}$, such that

$$A = \begin{pmatrix} 0 & \alpha_1 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & \alpha_3 & 0 \\ \alpha_4 & 0 & 0 & 0 & 0 \\ \alpha_5 & 0 & \alpha_6 & 0 & \alpha_7 \\ 0 & 0 & \alpha_8 & 0 & 0 \end{pmatrix}$$

is doubly stochastic. A simple computation shows that such an assignment is not possible unless $\alpha_2 = \alpha_5 = \alpha_6 = 0$, which is a contradiction. •

Example 3.10 (Doubly stochasticable digraph)**.** Consider the digraph G shown in Figure 2(b). One can observe that the only principal cycle set of G contains the two cycles shown in Figure 3. Both of these cycles pass through all the vertices of the digraph and thus, using Corollary 3.8, this digraph is doubly stochasticable. Note that this digraph has another three cycles, shown in Figure 4, none of which are in the principal cycle set. The adjacency matrix assignment

$$A = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

obtained by the sum of the elements of the principal cycle set, is weight-balanced and satisfies the conditions of Theorem 3.2 and thus is doubly stochasticable. Also note that not all the weight-balanced adjacency assignments

become doubly stochastic under the row-stochastic normalization map. An example is given by

$$A = \begin{pmatrix} 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 2 & 0 \end{pmatrix}. \qquad \bullet$$

An alternative question to the one considered above would be to find a set of edge weights (some possibly zero) that make the digraph doubly stochastic. Such assignments exist for the digraph in Figure 1. However, such weight assignments are not guaranteed, in general, to preserve the connectivity of the digraph. The following result gives a sufficient condition for the existence of such a weight assignment.

Proposition 3.11 (Doubly stochasticable digraphs via weight assignments with some zero entries)**.** *A strongly connected digraph G admits an edge weight assignment (where some entries might be zero) such that the resulting weighted digraph is strongly connected and doubly stochastic if there exists a cycle containing all the vertices of G.*

It is an interesting research question to determine sufficient and necessary conditions that characterize digraphs that are doubly stochasticable via weight assignments that can have some zero entries and still preserve the graph connectivity. Regarding Proposition 3.11, note that even if connectivity is preserved, fewer edges lead to smaller algebraic connectivity [39], which in turn affects negatively the rate of convergence of the consensus, optimization, and gossip algorithms executed over doubly stochastic digraphs, see e.g., [5, 8, 14, 22].

3.3 Properties of the topological character of doubly stochasticable digraphs

In this section, we investigate the properties of DS-cycle sets and of their cardinality $\mathrm{ds}(G)$. First, we show that rational weight assignments are sufficient to make an adjacency matrix doubly stochastic.

Lemma 3.12 (Rational weight assignments)**.** *Assume there exists a real weight assignment that makes a digraph G doubly stochastic. Then, there also exists a rational weight assignment that makes G doubly stochastic.*

Proof. By assumption, G is doubly stochasticable. Thus, by Corollary 3.8, it has a DS-cycle set. Using elements of this DS-cycle set, one can construct an integer weight assignment A_{wb} that makes G weight balanced and satisfies the conditions of Theorem 3.2 with some $C \in \mathbb{Z}_{>0}$. Therefore, $\frac{1}{C}A_{\mathrm{wb}}$ gives rise to a rational weight assignment that makes G doubly stochastic. □

Lemma 3.12 allows us to focus the attention, without loss of generality, on doubly stochastic adjacency matrices with rational entries or, alternatively, on weight-balanced adjacency matrices with integer entries whose rows and columns all sum up to the same integer. This is what we do in the rest of the paper.

Proposition 3.13 (Properties of the DS-index)**.** *Let G be a strongly connected doubly stochasticable digraph of order $n \in \mathbb{Z}_{>0}$ with DS-index* $\mathrm{ds}(G)$*. Then for $C \geq \mathrm{ds}(G)$, there exists a weight assignment $A_{\mathrm{wb}} \in \mathbb{Z}_{\geq 0}^{n\times n}$ that makes G a C-regular digraph.*

Proof. By Corollary 3.8, it is clear that one can generate $A_{\mathrm{wb}} \in \mathbb{Z}_{\geq 0}^{n\times n}$ that makes G weight-balanced and also satisfies the conditions of Theorem 3.2 for $C = \mathrm{ds}(G)$, just by taking the weighted union of the members of a DS-cycle set. Let $C > \mathrm{ds}(G)$. Choose a set of integer numbers $\lambda_i \in \mathbb{Z}_{>0}$, for $i \in \{1,\dots,\mathrm{ds}(G)\}$, such that $\sum_{i=1}^{\mathrm{ds}(G)} \lambda_i = C$. Consider the adjacency matrix

$$A = \sum_{i=1}^{\mathrm{ds}(G)} \lambda_i A_{\mathrm{cyc}}^i,$$

where A_{cyc}^i is the extended adjacency matrix associated to the ith element of the DS-cycle set. The matrix A is weight-balanced adjacency and satisfies the conditions of Theorem 3.2. □

We finish this section by bounding the DS-index of a digraph.

Lemma 3.14 (Bounds for the DS-index)**.** *Let $G = (V,E)$ be a doubly stochasticable strongly connected digraph. Then*

$$\max\{\max_{v\in V} d_{\mathrm{out}}(v), \max_{v\in V} d_{\mathrm{in}}(v)\} \leq \mathrm{ds}(G) \leq |E| - |V| + 1.$$

Proof. The first inequality follows from the fact that none of the out-edges of the vertex v (similarly, none of the in-edges of this vertex) are contained in the same element of any DS-cycle set $DS(G)$. To show the second inequality, take any element of $DS(G)$. This element must contain $|V|$ edges. The rest of the edges of the digraph can be represented by at most $|E| - |V|$ elements of $\mathscr{C}(G)$, and hence the bound follows. □

4 Strategies for making a digraph weight-balanced

The existing centralized algorithm for constructing a weight-balanced digraph, proposed in [18], relies on computing all the cycles of the digraph and thus it is computationally complex. In this section, we instead introduce two distributed strategies that are guaranteed to find a weight-balanced adjacency matrix for a weight-balanceable digraph and compare their convergence properties. Given the characterization of Theorem 2.3, we focus on strongly semiconnected digraphs. Since each strongly connected component of the digraph is completely independent from the others and can be balanced separately, without loss of generality we deal with strongly connected digraphs throughout the section.

4.1 The `imbalance-correcting algorithm`

Given a strongly connected digraph G, we introduce an algorithm, distributed over G, which allows the agents to balance their in- and out-degrees. An informal description of the `imbalance-correcting algorithm` is given in Table 1. Note that this algorithm updates the weights synchronously. A formal description of the

Setup

1: the communication topology is given by a strongly connected digraph
2: each agent can send messages to its out-neighbors and receive messages from its in-neighbors

At initialization, each agent

1: assigns a weight to each of its out-edges and sends this information to its corresponding out-neighbor
2: computes in-degree with messages received

At each round, each agent synchronously

1: **if** in-degree is greater than out-degree **then**
2: selects one of the out-edges with minimum weight
3: changes selected out-edge weight to correct the imbalance
4: sends a message to the corresponding out-neighbor informing of the change
5: recomputes out-degree
6: **end if**
7: updates in-degree with messages received

Table 1: The `imbalance-correcting algorithm`.

`imbalance-correcting algorithm` is as follows. Let $G = (V, E)$ denote the strongly connected digraph describing the communication topology. Let $\mathrm{Adj}(G) \subset \mathbb{R}^{n\times n}_{\geq 0}$ be the subset of all possible adjacency matrices with nonnegative entries associated to G. For $A \in \mathrm{Adj}(G)$ and $i \in \{1,\dots,n\}$, let

$$a_i^* = \min_{k\in\{1,\dots,n\}\setminus\{i\}} \{a_{ik} \mid a_{ik} \neq 0\},$$
$$J_i^* = \{j \in \{1,\dots,n\} \setminus \{i\} \mid a_{ij} = a_i^*\}.$$

The set-valued evolution map $f_{\mathrm{imcor}} : \mathrm{Adj}(G) \rightrightarrows \mathrm{Adj}(G)$ assigns to $A = (a_{ij}) \in \mathrm{Adj}(G)$ the set

$$f_{\mathrm{imcor}}(A) = \Bigg\{ B \in \mathrm{Adj}(G) \mid \text{for } i \in \{1,\dots,n\} \text{ with } \omega(v_i) \leq 0,\ b_{ij} = a_{ij} \text{ for } j \in \{1,\dots,n\} \text{ and} \tag{6}$$
$$\text{for } i \in \{1,\dots,n\} \text{ with } \omega(v_i) > 0, \text{ there is } j_i^* \in J_i^* \text{ such that } b_{ij} = \begin{cases} a_{ij} + \omega(v_i), & \text{if } j = j_i^*, \\ a_{ij}, & \text{if } j \in \{1,\dots,n\} \setminus \{j_i^*\}. \end{cases} \Bigg\},$$

where recall that $\omega(v_i)$ denotes the imbalance (1) of vertex v_i. Note that the map f_{imcor} defines a discrete-time set-valued dynamical system on $\mathrm{Adj}(G)$ given by

$$A(k+1) \in f_{\mathrm{imcor}}(A(k)), \quad \text{for all } k \in \mathbb{Z}_{\geq 0},$$

which corresponds to the `imbalance-correcting algorithm` presented in Table 1. Our strategy to establish the correctness of the algorithm is then based on characterizing the properties of the set-valued map f_{imcor} and then applying the LaSalle invariance principle, cf. Theorem 2.6. We first establish that the map is closed.

Lemma 4.1 (Closedness of f_{imcor})**.** *The map f_{imcor} is closed on* $\mathrm{Adj}(G)$.

Proof. Let $D \in \mathrm{Adj}(G)$ and consider two sequences $\{D_k\}_{k=1}^{\infty} \subset \mathrm{Adj}(G)$ and $\{C_k\}_{k=1}^{\infty} \subset \mathrm{Adj}(G)$ such that $C_k \in f_{\mathrm{imcor}}(D_k)$, for all $k \in \mathbb{Z}_{>0}$, $\lim_{k\to\infty} D_k = D$ and $\lim_{k\to\infty} C_k = C$. Since $J_i^*(D_k) \subseteq \{1,\dots,n\}$ and n is finite, there

must exist a set $J_1 \times \cdots \times J_n$ in $\{1,\ldots,n\}^n$ that appears infinitely often in the sequence $\{J_1^*(D_k) \times \cdots \times J_n^*(D_k)\}_{k=1}^{\infty}$. Therefore, there exists a subsequence of $\{D_k\}_{k=1}^{\infty}$ which, with a slight abuse of notation and for simplicity, we denote in the same way, such that $J_i^*(D_k) = J_i$, for all $k \in \mathbb{Z}_{\geq 1}$. Now, because $C_k \in f_{\text{imcor}}(D_k)$, there exist $(j_1(k),\ldots,j_n(k)) \in J_1 \times \ldots \times J_n$ for each $k \in \mathbb{Z}_{>0}$ such that one can write

$$(C_k)_{ij} = \begin{cases} (D_k)_{ij} + \omega_k(v_i) & \text{if} \quad \omega_k(v_i) > 0, \text{ and } j = j_i(k), \\ (D_k)_{ij}, & \text{otherwise,} \end{cases} \tag{7}$$

where $(D_k)_{ij}$ and $(C_k)_{ij}$ denote, respectively, the entries of D_k and C_k, and $\omega_k(v_i)$ is the imbalance of vertex v_i in the weight assignment D_k. Reasoning as before, since $J_1 \times \ldots \times J_n$ is finite, there exists an element $(j_1,\ldots,j_n)$ of this set that appears infinitely often in the sequence $\{(j_1(k),\ldots,j_n(k))\}_{k=1}^{\infty}$. Therefore, there exists a subsequence, which with a slight abuse of notation we denote in the same way, such that $(j_1(k),\ldots,j_n(k)) = (j_1,\ldots,j_n)$ for all $k \in \mathbb{Z}_{>0}$. Note that, for each $i \in \{1,\ldots,n\}$ such that $\omega(v_i) > 0$, the element j_i is unique since otherwise the sequence $\{C_k\}_{k=1}^{\infty}$ would not be convergent. Combining these facts with (7) and taking the limit as $k \to \infty$, we conclude that $C \in f_{\text{imcor}}(D)$, and hence f_{imcor} is closed at D, as claimed. □

Next, we characterize the fixed points of f_{imcor}.

Lemma 4.2 (Fixed points of f_{imcor})**.** *f_{imcor} has at least one fixed point. Furthermore, $A^* \in \text{Adj}(G)$ is a fixed point if and only if A^* is weight-balanced.*

Proof. Given that the digraph is strongly connected, Theorem 2.3 guarantees that it is weight-balanceable, and therefore at least one fixed point exists (if $A \in \text{Adj}(G)$ is weight-balanced, then $f_{\text{imcor}}(A) = \{A\}$, and hence A is a fixed point of f_{imcor}). On the other hand, suppose $A^* \in \text{Adj}(G)$ satisfies $A^* \in f_{\text{imcor}}(A^*)$. We reason by contradiction. If A^* is not weight-balanced, then (2) would imply that there exists at least a vertex $v \in V$ with $\omega(v) > 0$. From (6), this would imply $A^* \notin f_{\text{imcor}}(A^*)$, which is a contradiction. □

The logic used by the `imbalance-correcting algorithm` to update edge weights consists of using edges with the minimum weight. The next result shows that such logic is powerful in terms of propagating a token (in our case, an imbalance) across the network.

Lemma 4.3 (Propagation of tokens via out-edges with minimum weight)**.** *Let G be a strongly connected weighted digraph and consider a finite number of tokens initially located at some nodes. If each node that possesses a token repeatedly passes it to one of its out-neighbors via an out-edge with the minimum weight and adds a positive constant to this weight, all the nodes will be visited by at least one token after a finite number of iterations.*

Proof. Let V denote the vertex set of G. Let t_0 denote an arbitrary initial time and $\text{Vis}(t,t_0)$ be the set of nodes visited by any of the tokens up to time t. Since V is finite and $\text{Vis}(t,t_0) \subset \text{Vis}(t+1,t_0)$, for $t \in \mathbb{Z}_{\geq 0}$, we deduce that there exists T such that $\text{Vis}(t,t_0) = Y(t_0) \subseteq V$, for all $t \geq T$. Let us show that $Y(t_0) = V$ for all t_0. We do this by

discarding any other possibility. Clearly, $|Y(t_0)|$ cannot be 1 because if a node has a token, it will pass it to some other node. Let $m < |V|$ and assume that we have shown that $|Y(t_0)| \neq 1, \ldots, m-1$ for all t_0. Choose any t_0 and let us show that $|Y(t_0)| \neq m$. Suppose otherwise, i.e., $|Y(t_0)| = m$. Since the digraph is strongly connected, there exists at least one edge from a member of $Y(t_0)$, say v_k, to a member of $V \setminus Y(t_0)$, say v_{k+1}. Since $v_k \in Y(t_0)$, v_k has one of the tokens at some point in time, say t_1. Suppose v_k sends this token via an out-edge to a member of $Y(t_0)$ and increases the weight on this edge; otherwise, $|Y(t_0)| \geq m+1$, which is a contradiction. By hypothesis, the tokens never reach $V \setminus Y(t_0)$ and get passed in $Y(t_0)$ while increasing the weight of edges among nodes in $Y(t_0)$. We claim that, after a finite time, at least one of the tokens comes back to v_k. Suppose otherwise, i.e., for $t > t_1$, no token will ever visit v_k. Since $Y(t_1+1) \subseteq Y(t_0)$ and $v_k \notin Y(t_1+1)$, we conclude that $|Y(t_1+1)| \leq m-1$, which is a contradiction. Therefore, a token must visit v_k after t_1. When this occurs, node v_k will choose the out-edge with minimum weight for sending the token. The whole process gets repeated and thus, after a finite time, the weight on the out-edge to v_{k+1} has the minimum weight, and thus v_k sends a token to v_{k+1}, implying $|Y(t_0)| \geq m+1$, which is a contradiction. Therefore, $|Y(t_0)| = |V|$, as claimed. □

The last ingredient we need in order to characterize the convergence of the `imbalance-correcting algorithm` is the Lyapunov function $V_{\mathrm{wb}} : \mathrm{Adj}(G) \to \mathbb{R}_{\geq 0}$ defined by

$$V_{\mathrm{wb}}(A) = \sum_{i=1}^{n} \Big| \sum_{j=1}^{n} a_{ij} - \sum_{j=1}^{n} a_{ji} \Big| = \sum_{i=1}^{n} |\omega(v_i)|. \tag{8}$$

This function is continuous on $\mathrm{Adj}(G)$. Note that $V(A) = 0$ if and only if A is weight-balanced. We are now ready to state our convergence result.

Theorem 4.4 (Convergence of the `imbalance-correcting algorithm`)**.** *For a strongly connected digraph G, any evolution under the* `imbalance-correcting algorithm` *converges in finite time to a weight-balanced adjacency matrix.*

Proof. Recall that the evolutions under the `imbalance-correcting algorithm` correspond to trajectories of the discrete-time dynamical system defined by f_{imcor}. For an initial condition $A \in \mathrm{Adj}(G)$, consider the sublevel set $V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A)) = \{B \in \mathrm{Adj}(G) \mid 0 \leq V_{\mathrm{wb}}(B) \leq V_{\mathrm{wb}}(A)\}$. Since V_{wb} is continuous, this set is closed. Let us show that the continuous function V_{wb} is non-increasing along f_{imcor}, and consequently $V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A))$ is strongly positively invariant with respect to f_{imcor}. Note that the weight of any given edge is only modified by the agent who has it as an out-edge. Consider a weight change done by the algorithm on an arbitrary edge (v_i, v_j), $v_i, v_j \in V$, and let us see the effect it has on the imbalances of the two vertices it affects. According to f_{imcor}, v_i increases the weight on its out-edge to v_j by $\varepsilon \in \mathbb{R}_{>0}$ in order to balance itself. Consequently, the imbalance of agent v_j increases by at most ε. Considered together, the weight change performed in the edge decrease the imbalance of v_i by ε while increasing the imbalance of v_j by at most ε; thus the value of that V_{wb} does not increase. Since the edge (v_i, v_j) is arbitrary, from (8), this argument implies that V_{wb} is non-increasing along f_{imcor}.

Next, let us show that all evolutions of $(\mathrm{Adj}(G), f_{\mathrm{imcor}})$ with initial condition in $V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A))$ are bounded. Suppose otherwise, and take an initial condition that gives rise to an unbounded trajectory. This implies that there exists, infinitely often, an agent $v \in V$ with some positive imbalance. Let us justify that the infimum θ of all these positive imbalances is positive too. Since, by definition of f_{imcor}, any imbalance in the timesteps after the initial one is a linear combination of the initial imbalances with coefficients 0 or 1, the following inequality holds

$$\theta \geq \nu = \min_{k\in\{1,\ldots,n\}} \min_{(i_1,\ldots,i_k)\in C(n,k)} \{|\omega_0(v_{i_1}) + \omega_0(v_{i_2}) + \ldots + \omega_0(v_{i_k})| \neq 0\},$$

where $C(n,k)$ denotes the set of combinations of k elements out of n, and $\omega_0(v)$ refers to the initial imbalance of $v \in V$. Note that $\nu > 0$. Since $\sum_{v\in V} \omega(v) = 0$, there exists a set of agents $Y = \{v_1, \ldots, v_k\} \subset V$, $k \in \mathbb{Z}_{>0}$, with negative imbalance, such that $\sum_{v_i\in Y} \omega(v_i)$ is at most $-\theta$ infinitely often. We show that this actually must happen always. Suppose that at some point in the execution of the algorithm $\sum_{v_i\in Y} \omega(v_i) > -\theta$. After that, it will never happen that $\sum_{v_i\in Y} \omega(v_i) = -\theta$, otherwise, at least one of the agents has decreased its imbalance from a negative value, which is not possible by the definition of f_{imcor}. But this contradicts $\sum_{v_i\in Y} \omega(v_i)$ being at most $-\theta$ infinitely often. The above argument shows that there exists a positive imbalance of at least θ that does not visit, even once, any edge going into the set Y. Finally, since G is strongly connected, this contradicts Lemma 4.3.

Finally, recall that by Lemma 4.1, the map f_{imcor} is closed on $V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A))$. With all the above hypotheses satisfied, the application of the LaSalle Invariance Principle for set-valued dynamical systems, cf. Theorem 2.6, implies that the algorithm evolution approaches a set of the form $V_{\mathrm{wb}}^{-1}(c) \cap S$, where $c \in \mathbb{R}$ and S is the largest weakly positively invariant set contained in $\{B \in V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A)) \,|\, \text{there exists } \ C \in f_{\mathrm{imcor}}(B) \text{ such that } V_{\mathrm{wb}}(C) = V_{\mathrm{wb}}(B)\}$. In order to complete the proof, we need to show that $c = 0$. We reason by contradiction. Suppose $c > 0$ and let $A \in S \cap V_{\mathrm{wb}}^{-1}(c)$. Since S is weakly positively invariant, there exists an evolution starting from A which is contained in S, and hence stays in the level set $V_{\mathrm{wb}}^{-1}(c)$. Let us show that after a finite time, this evolution will leave $V_{\mathrm{wb}}^{-1}(c)$, hence reaching a contradiction. Since $c \in \mathbb{R}_{>0}$, there exists at least one agent v_i with positive imbalance $\omega(v_i) > 0$. Therefore, since $\sum_{v\in V} \omega(v) = 0$, there exists a set of agents whose imbalances are negative and sum at least $-\omega(v_i)$. Since the digraph is strongly connected, by Lemma 4.3, after a finite time the positive imbalance $\omega(v_i)$ will visit this set, which would strictly decrease the value of V_{wb}, reaching a contradiction. Finally, the finite-time convergence to the set of weight-balanced adjacency matrices follows from noting that the value of V_{wb} is initially finite and, by Lemma 4.3, there exists a finite number of iterations after which V_{wb} is guaranteed to have decreased at least an amount 2ν. The convergence to a weight-balanced adjacency matrix is a consequence of the finite-time convergence to the set. □

Remark 4.5. (Time complexity of the `imbalance-correcting algorithm`): Roughly speaking, the time complexity of an algorithm corresponds to the number of rounds required to achieve its objective. More formal definitions can be found in [6, 24, 31, 38]. Therefore, characterizing the worst-case time complexity provides a termination time for the algorithm. The characterization of the time complexity of the `imbalance-correcting`

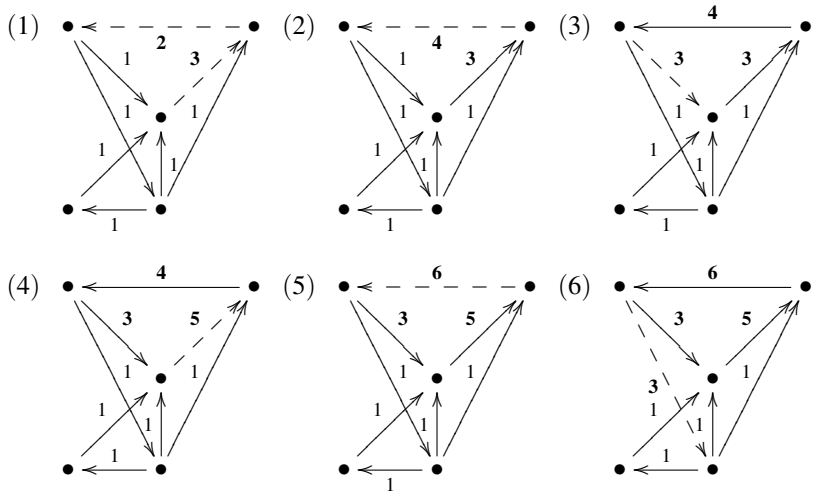

Figure 5: Execution of the `imbalance-correcting algorithm` for the digraph of Figure 2(a). The initial condition is the adjacency matrix where all edges are assigned weight 1. In each round, the edges which are used for sending messages are shown dashed. One can observe that the Lyapunov function $V_{\mathrm{wb}}(A_0)$ takes the following values in subsequent time steps: 6, 4, 4, 4, 4, 4, 0.

`algorithm` is an open problem (however, we characterize in Section 4.2 the time complexity of a modified version of this algorithm). In the absence of this characterization, agents could alternatively implement a version of the convergecast algorithm [31] to determine whether every agent in the digraph has achieved zero imbalance.•

Example 4.6. (Execution of the `imbalance-correcting algorithm`): Consider the digraph of Figure 2(a). The `imbalance-correcting algorithm` converges to a weight-balanced digraph in 6 rounds, as illustrated in Figure 5. •

4.2 The `mirror imbalance-correcting algorithm`

In this section, we modify the `imbalance-correcting algorithm` to synthesize a strategy, distributed over the mirror of the digraph, which converges quickly to a weight-balanced digraph. This procedure can be employed to construct a weight-balanced digraph without computing any cycle. An informal description of the `mirror imbalance-correcting algorithm` is given in Table 2.

A formal description of the `mirror imbalance-correcting algorithm` is as follows. Let $G = (V,E)$ denote the strongly connected digraph describing the communication topology. For all $i \in \{1,\ldots,n\}$, let

$$J_i^{\#} = \{j \in \{1,\ldots,n\} \setminus \{i\} \mid v_j \in \mathcal{N}^{\mathrm{out}}(v_i) \text{ and } \omega(v_j) = \min_{v_l \in \mathcal{N}^{\mathrm{out}}(v_i)} \omega(v_l)\}.$$

Setup

1: the communication topology is given by a strongly connected digraph
2: each agent can send messages to its out-neighbors and receive messages from its in-neighbors
3: each agent has a memory of the number of messages sent to each of its neighbors

At initialization, each agent

1: assigns a weight to each of its out-edges and sends this information to its corresponding out-neighbor
2: computes in-degree with messages received

At each round, each agent synchronously

1: **if** in-degree is greater than out-degree **then**
2: selects out-neighbors with minimum imbalance
3: among these, selects randomly among the ones with minimum messages sent to (*fair-decision rule*)
4: changes corresponding out-edge weight to correct the imbalance
5: sends a message to the corresponding out-neighbor informing of the change
6: updates memory of number of messages sent to out-neighbor
7: recomputes out-degree
8: **end if**
9: updates in-degree with messages received

Table 2: The `mirror imbalance-correcting algorithm`.

We define $g_{\text{imcor}} : \text{Adj}(G) \rightrightarrows \text{Adj}(G)$ by

$$g_{\text{imcor}}(A) = \Big\{ B \in \text{Adj}(G) \mid \text{for } i \in \{1,\dots,n\} \text{ with } \omega(v_i) \leq 0,\ b_{ij} = a_{ij} \text{ for } j \in \{1,\dots,n\} \text{ and} \tag{9}$$

$$\text{for } i \in \{1,\dots,n\} \text{ with } \omega(v_i) > 0, \text{ there is } j_i^{\#} \in J_i^{\#} \text{ such that } b_{ij} = \left\{ \begin{array}{ll} a_{ij} + \omega(v_i), & \text{if } j = j_i^{\#}, \\ a_{ij}, & \text{if } j \in \{1,\dots,n\} \setminus \{j_i^{\#}\}. \end{array} \right\}.$$

This map defines a discrete-time set-valued dynamical system on $\text{Adj}(G)$ given by

$$A(k+1) \in g_{\text{imcor}}(A(k)), \quad \text{for all } k \in \mathbb{Z}_{\geq 0}.$$

Note that the evolutions of the `mirror imbalance-correcting algorithm` are a subset of all the evolutions of this dynamical system (because of the fair-decision rule). The next result shows that this algorithm converges in finite time to a weight-balanced digraph.

Theorem 4.7 (Convergence of the `mirror imbalance-correcting algorithm`)**.** *For a strongly connected digraph G, any evolution under the* `mirror imbalance-correcting algorithm` *converges in finite time to a weight-balanced adjacency matrix.*

Proof. For an initial condition $A \in \text{Adj}(G)$, consider the sublevel set $V_{\text{wb}}^{-1}(\leq V_{\text{wb}}(A))$. Similarly to the proof of Theorem 4.4, observe that the continuous function V_{wb} is non-increasing along g_{imcor} and thus $V_{\text{wb}}^{-1}(\leq V_{\text{wb}}(A))$ is strongly positively invariant with respect to g_{imcor}. An argument similar to the one in Lemma 4.1 shows that the map g_{imcor} is closed at $V_{\text{wb}}^{-1}(\leq V_{\text{wb}}(A))$. Now, pick an evolution of $(\text{Adj}(G), g_{\text{imcor}})$ with initial condition in $V_{\text{wb}}^{-1}(\leq V_{\text{wb}}(A))$ that satisfies the fair-decision rule. Then a similar version of Lemma 4.3 can be used to establish that this evolution is bounded. Thus by the LaSalle Invariance Principle for set-valued dynamical systems,

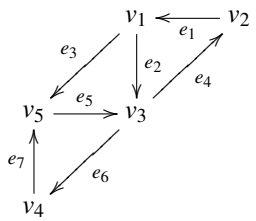

Figure 6: A strongly connected digraph.

Theorem 2.6, this evolution approaches a set of the form $V_{\mathrm{wb}}^{-1}(c) \cap S$, where $c \in \mathbb{R}_{\geq 0}$ and S is the largest weakly positively invariant set contained in $\{B \in V_{\mathrm{wb}}^{-1}(\leq V_{\mathrm{wb}}(A)) \,|\, \text{there exists } C \in g_{\mathrm{imcor}}(B) \text{ such that } V_{\mathrm{wb}}(C) = V_{\mathrm{wb}}(B)\}$. The fact that c must be zero follows from the observation that, under the `mirror imbalance-correcting algorithm`, the value of V_{wb} decreases by a finite amount, bounded away by a positive constant, after a finite number of iterations. The convergence in finite time can also be justified in a similar way to Theorem 4.4. □

Example 4.8 (Execution of the `mirror imbalance-correcting algorithm`)**.** Figure 6 shows a strongly connected digraph. Figure 7 shows an execution of the `mirror imbalance-correcting algorithm` for this digraph that converges in three rounds to a weight-balanced digraph. Note that agent v_4 has two out-neighbors

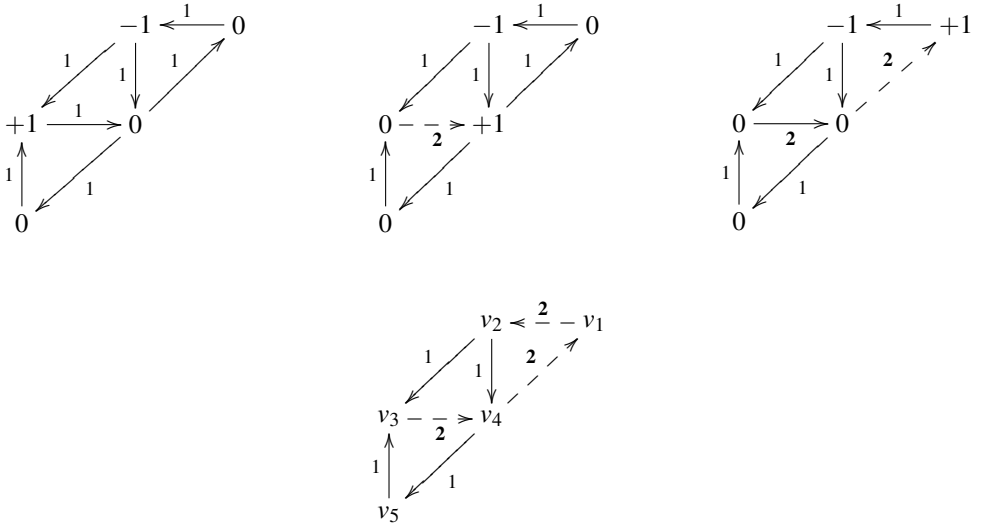

Figure 7: An execution of `mirror imbalance-correcting algorithm` for the digraph of Figure 6. The dashed lines show the edges which have been used to send messages.

with the same imbalance, namely v_1 and v_5. If v_4 decides to update the weight on the edge (v_4, v_5) to correct its imbalance, then the fair-decision rule will make it choose the edge (v_4, v_1) the next time. Figure 8 shows the result of the execution of the `mirror imbalance-correcting algorithm` in such a case. If after the 4th round of the execution the agent v_4 would keep updating the weight on the edge to agent v_5, then the algorithm would never converge to a weight-balanced digraph. •

Next, we investigate the rate of convergence of the `mirror imbalance-correcting algorithm`.

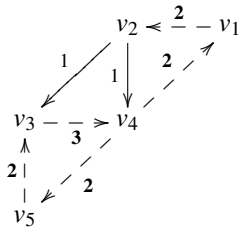

Figure 8: Another possible execution of `mirror imbalance-correcting algorithm` for Example 4.8. The dashed lines show the edges which have been used to send messages.

Proposition 4.9. (Time complexity of the `mirror imbalance-correcting algorithm`): *The time complexity of the `mirror imbalance-correcting algorithm` is in* $O(n^4)$.

Proof. Let $G = (V,E)$ be a strongly connected digraph and consider the agent $v_i \in V$ which initially has the maximum positive imbalance $\omega(v_i) = r \in \mathbb{R}_{>0}$. In the worst-case scenario, this imbalance should reach r agents with negative imbalance of -1 and it has to go through the longest path in order to reach the first agent with negative imbalance -1 and, after that, through the longest path to get to the second agent with negative imbalance and so on. According to the execution of the `mirror imbalance-correcting algorithm`, agent $v_i \in V$ updates the weight on the edge to an out-neighbor $v_{i+1} \in V$. Then, agent v_{i+1} might pass the imbalance to the next agent in the longest path directly, or after sending it to a cycle which does not include any agent with negative imbalance. Then, after some finite time, the agent v_{i+1} will receive the positive imbalance r again and by the fair-decision rule, this time it will pick a different out-neighbor. Now, suppose that all the agents in this longest path have the maximum number of out-neighbors and furthermore, suppose that all the agents try all their other out-neighbors (via cycles) before finding the correct out-neighbor. Since the longest path, the maximum length of a cycle, and the maximum out-degree are all in $O(n)$, the time complexity of the positive imbalance r reaching the first agent with negative imbalance -1 is in $O(n^3)$. Since the imbalance r is in $O(n)$, the time complexity of `mirror imbalance-correcting algorithm` is in $O(n^4)$. □

The result stated in Proposition 4.9 is to be contrasted with the time complexity of the centralized algorithm proposed in [18] to construct a weight-balanced digraph, which can be shown to be in $O(2^{n^2})$.

5 Strategies for making a digraph doubly stochastic

In this section, we investigate the design of cooperative strategies running on strongly connected communication networks that allow agents to determine a set of edge weights that make the digraph doubly stochastic. The key difference with respect to Section 4 is that weight-balancing strategies aim at achieving zero imbalance for all agents, whereas the doubly-stochastic strategies must additionally ensure that all agents share the same in-degree.

5.1 The `imbalance-correcting algorithm with self-loop addition`

From Corollary 3.3, we know that all strongly connected digraphs can be made doubly stochastic by allowing the addition of self-loops to the digraph. Combining this observation with the `imbalance-correcting algorithm`, we synthesize a distributed strategy that allows the agents to make any strongly connected communication network doubly stochastic. This strategy, named the `imbalance-correcting algorithm with self-loop addition`, is described in Table 3. Note that there are a number of distributed ways to compute the maximum out-degree (cf. Step `2:` of the initialization stage in Table 3), see e.g., [24, 31].

Setup
1: the communication topology is given by a strongly connected digraph
2: each agent can send messages to its out-neighbors and receive messages from its in-neighbors
At initialization, group of agents
1: executes the `imbalance-correcting algorithm`
2: computes maximum out-degree
At each round, each agent synchronously
1: **if** out-degree is smaller than maximum out-degree **then**
2: adds a self-loop with appropriate weight to make out-degree equal to maximum out-degree
3: divides weights on each out-edge by maximum out-degree
4: **end if**

Table 3: The `imbalance-correcting algorithm with self-loop addition`.

Theorem 5.1. (`imbalance-correcting algorithm with self-loop addition`): *The execution of* `imbalance-correcting algorithm with self-loop addition` *over a strongly connected communication topology gives rise to a doubly stochastic weighted digraph.*

Example 5.2. (Execution of the `imbalance-correcting algorithm with self-loop addition`): Consider the digraph of Figure 2(a). As we showed, there exists no doubly stochastic adjacency matrix associated to this digraph. However, if we allow the agents to modify the digraph by adding self-loops, a doubly stochastic adjacency matrix can be associated to this digraph. Using the `imbalance-correcting algorithm`, the agents can compute the weight-balanced adjacency matrix

$$A = \begin{pmatrix} 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 0 \\ 5 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Now, if the agents execute the modification of the digraph by adding self-loops with appropriate weight, they

compute the adjacency matrix

$$A = \begin{pmatrix} 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 0 \\ 5 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 0 & 5 \end{pmatrix}.$$

Finally, by executing a division on the weight of the out-edges, the agents compute the doubly stochastic adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 5/6 & 0 & 1/6 & 0 & 0 \\ 1/6 & 0 & 1/6 & 1/2 & 1/6 \\ 0 & 0 & 1/6 & 0 & 5/6 \end{pmatrix}. \qquad \bullet$$

5.2 The `load-pushing algorithm`

Given a digraph $G = (V,E)$ and $C \in \mathbb{Z}_{>0}$, in this section we introduce the `load-pushing algorithm`. This is a distributed strategy over the mirror digraph of G, that, upon completion, allows the group of agents to declare whether the graph G is C-regular. If it is, the strategy finds a set of weights that makes the graph C-regular. Note that, once such weights have been found, agents can easily determine a doubly stochastic weight assignment by dividing all entries by C.

Our strategy is essentially an adaptation of the Goldberg-Tarjan's algorithm [15] for the problem under consideration. This point will be clearer when we address the proof of its correctness in Theorem 5.3. The `load-pushing algorithm` is formally described in Table 4.

The next result characterizes the convergence properties of the `load-pushing algorithm`.

Theorem 5.3. (The `load-pushing algorithm` finds a C-regular weight assignment iff the digraph is doubly stochasticable): *Let $G = (V,E)$ be a digraph and $C \geq \max\{\max_{v\in V} d_{\text{out}}(v), \max_{v\in V} d_{\text{in}}(v)\}$. If the `load-pushing algorithm` announces that G is not C-regular and $C \geq |E| - |V| + 1$, then the digraph is not doubly stochasticable. If G is doubly stochasticable and $C \geq \text{ds}(G)$, then the `load-pushing algorithm` converges to a C-regular digraph. In both cases, the algorithm terminates in $O(|V|^2 \times |E|)$ steps.*

Proof. We start by showing that the problem of finding a C-regular digraph can be reduced to a maximum flow problem with positive lower bounds on the edge loads [1, 4]. Consider the bipartite digraph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, where

- $\widetilde{V}$ contains two copies of V, named V_u and V_w, a source node s, and a target node t, i.e.,

$$\widetilde{V} = \{s\} \cup V_u \cup V_w \cup \{t\}.$$

Setup

the communication topology is given by a strongly connected digraph G
integer $C \in \mathbb{Z}_{>0}$ is known to all agents
each agent can send/receive messages to/from its in- and out-neighbors

At initialization, each agent $i \in \{1,\dots,n\}$

assigns unit weight $a_{ij} := 1$ to each of its out-edges (v_i, v_j) and sends this information to its corresponding out-neighbor v_j
computes in-degree with messages received
initializes (*source-load*, *target-load*) vector $(L_s(v_i), L_t(v_i)) := (C - d_{\text{out}}(v_i), d_{\text{in}}(v_i) - C)$
initializes (*source-height*, *target-height*) vector $(H_s(v_i), H_t(v_i)) := (2, 1)$
sends $H_s(v_i)$ to its out-neighbors and $H_t(v_i)$ to its in-neighbors
computes $H_s^{\text{max-in}}(v_i) = \max_{v_j \in \mathscr{N}^{\text{in}}(v_i)} H_s(v_j)$ with messages received

At each round, each agent $i \in \{1,\dots,n\}$ synchronously

if $L_s(v_i) > 0$ **then**
  if there exists an out-neighbor $v_j \in \mathscr{N}^{\text{out}}(v_i)$, where $a_{ij} < C$ and $H_s(v_i) > H_t(v_j)$ **then**
    sends load $L_s(v_i)$ to v_j and sets $a_{ij} := a_{ij} + L_s(v_i)$ and $L_s(v_i) := 0$ (*push forward*)
  else
    announces that digraph is not C-regular, algorithm terminates (*declare digraph not C-regular*)
  end if
end if
if $L_t(v_i) > 0$ **then**
  if there exists an in-neighbor $v_k \in \mathscr{N}^{\text{out}}(v_i)$, where $a_{ki} > L_t(v_i) + 1$ and $H_t(v_i) > H_s(v_k)$ **then**
    sends load $L_t(v_i)$ to v_k and sets $a_{ki} := a_{ki} - L_t(v_i)$ and $L_t(v_i) := 0$ (*push backward*)
  else
    sets $H_t(v_i) := H_s^{\text{max-in}}(v_i) + 1$ and sends $H_t(v_i)$ to in-neighbors (*increase target-height*)
  end if
end if

Table 4: The `load-pushing algorithm`.

In other words, the nodes $u_i \in V_u$ and $w_i \in V_w$ correspond to the agent $v_i \in V$;

- $\widetilde{E}$ is defined as follows: there is no edge between vertices in V_u and there is no edge between vertices in V_w. For each edge (v_i, v_j) in E, there exists an edge $(u_i, w_j) \in \widetilde{E}$; there is an out edge between s to all u_i in V_u and an out-edge from each w_i in V_w to t.

Next, let us define a maximum flow problem on $\widetilde{G}$. Let the capacity on each edge of the form (s, u_i) or $(w_j, t) \in \widetilde{E}$, $i, j \in \{1,\dots,|V|\}$ be exactly C. Let the capacity on each edge $(u_i, w_j) \in \widetilde{E}$ be lower bounded by 1 and upper bounded by C. We refer to the weight of such edges as $\widetilde{a}_{ij}$, and therefore, $1 \le \widetilde{a}_{ij} \le C$. Consider the following problem: find the maximum flow that can be sent from the source s to the target t. It is not difficult to see that, by definition of C-regularity, the digraph G is C-regular if and only if the maximum flow of the problem just introduced is $C|V|$.

Following [1, 4], the maximum flow problem with lower bounds described above can be transformed into a regular maximum flow problem as follows. Define $\bar{a}_{ij} = \widetilde{a}_{ij} - 1$. The bounds $1 \le \widetilde{a}_{ij} \le C$ now become $0 \le \bar{a}_{ij} < C$.

For each edge of the form $(s,u_i) \in \widetilde{E}$, let the capacity be

$$C - \sum_{(u_i,w_j)\in\widetilde{E}} 1 = C - d_{\text{out}}(v_i).$$

For each edge of the form $(w_i,t) \in \widetilde{E}$, let the capacity be

$$-C + \sum_{(u_j,w_i)\in\widetilde{E}} 1 = -C + d_{\text{in}}(v_i).$$

The proof now follows from noting that the execution of the `load-pushing algorithm`, when transcribed to this maximum flow problem, exactly corresponds to the execution of the distributed version presented in [32] of the preflow-push algorithm of Goldberg-Tarjan algorithm [15]. Note that, given our discussion in Section 3, it is enough to execute `load-pushing algorithm` for $C \geq |E| - |V| + 1$ (cf. Lemma 3.14) to determine if the digraph G is indeed doubly stochasticable. The time complexity of the algorithm is a consequence of [15, Theorem 3.11]. □

It is worth mentioning that without the characterization of doubly stochasticable digraphs, a negative answer from `load-pushing algorithm` for a given C would be inconclusive to determine if the digraph is indeed doubly stochasticable. At the same time, the results of Section 3 imply that, if a digraph is doubly stochasticable and the `load-pushing algorithm` is executed with $C \geq \text{ds}(G)$ (which in particular is guaranteed if $C \geq |E| - |V| + 1$), then the strategy will find a set of appropriate weights.

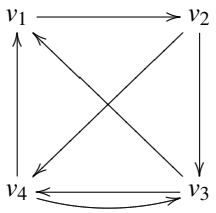

Figure 9: A doubly stochasticable digraph with $\text{ds}(G) = 2$.

Example 5.4. (Execution of the `load-pushing algorithm`): Consider the digraph shown in Figure 9. One can easily check that this digraph is doubly stochasticable and $\text{ds}(G) = 2$. Suppose the agents want to find a C-regular weight assignment and they run the `load-pushing algorithm` for $C = 3 \geq \text{ds}(G) = 2$. Figure 10 shows the execution of the algorithm. To represent its evolution, we associate to each vertex v_i, $i \in \{1,\ldots,4\}$, the source- and target-loads and a subindex with the source- and target-height. The algorithm converges to a 3-regular digraph in 5 iterations. •

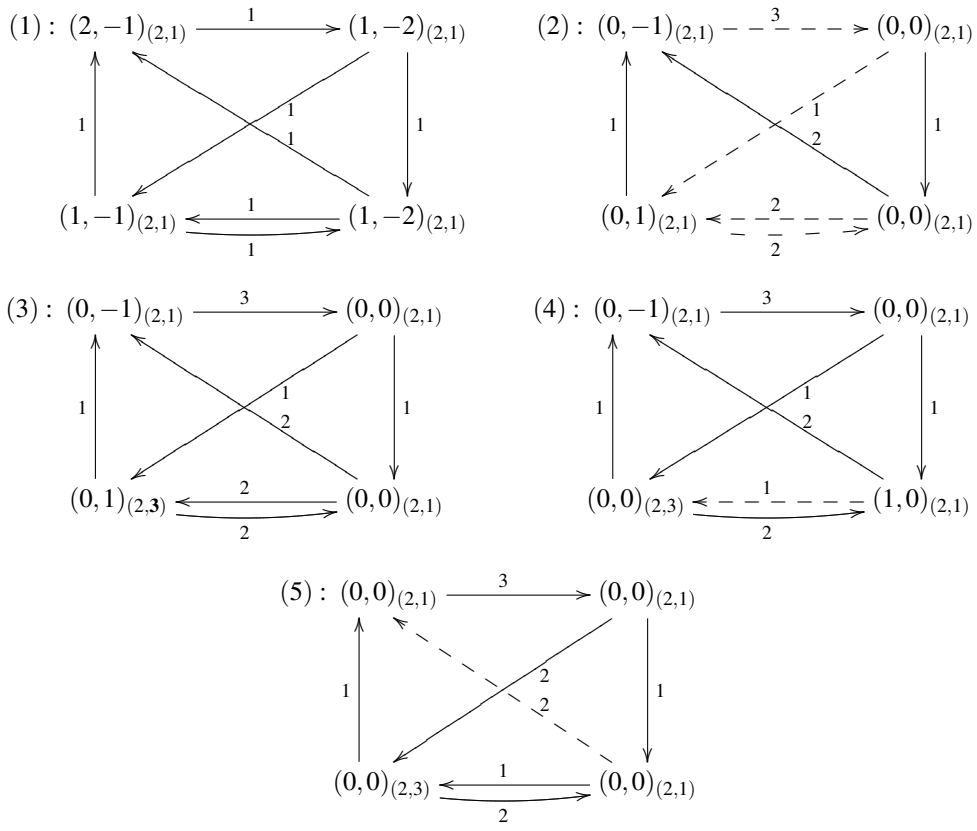

Figure 10: The execution of the `load-pushing algorithm` for the digraph in Figure 9. At each iteration of the algorithm, the pair $(L_s, L_t)_{(H_s,H_t)}$ is shown for each vertex. At each iteration, dashed lines represent the edges used by the vertices to send loads. Observe that (2), (3), (4), (5), respectively, correspond to the operations push forward, increasing target height, push backward, and push forward.

6 Conclusions

In this paper, we have studied two important classes of digraphs: weight-balanced and doubly stochastic digraphs. The first contribution is a necessary and sufficient condition for doubly stochasticability of a digraph. We have unveiled the particular connection of the class of doubly stochasticable digraphs with a special subset of weight-balanced digraphs. The second contribution is the design of two discrete-time dynamical systems for constructing a weight-balanced digraph from a strongly connected digraph: (i) the `imbalance-correcting algorithm`, running synchronously on a group of agents, and (ii) the `mirror imbalance-correcting algorithm`, distributed over the mirror of the original digraph. We have established the finite-time convergence of both strategies via the set-valued discrete-time LaSalle Invariance Principle. We have also characterized the time complexity of the `mirror imbalance-correcting algorithm`, which is substantially better than that of the existing centralized algorithm. The third contribution is the design of two discrete-time dynamical systems for constructing a doubly stochastic adjacency matrix for a doubly stochasticable strongly connected digraph: (i) the `imbalance-correcting algorithm with self-loop addition`, works under the assumption that agents are allowed to add weighted self-loops. We showed that any strongly connected digraph can be assigned a doubly stochastic adjacency matrix with this procedure; (ii) the `load-pushing algorithm`,

distributed over the mirror digraph, for the case when agents are not allowed to add self-loops. The convergence of this algorithm, and its time complexity, has been established by formulating the problem as a constrained maximum flow problem.

Our results greatly enlarge the domain of systems on which a variety of distributed formation, consensus, and optimization algorithms can be executed by providing strategies that agents can use to form weight-balanced and doubly stochastic communication topologies. Regarding future work, we would like to better understand the gap between $\mathrm{ds}(G)$ and $p(G)$ for doubly stochasticable digraphs and develop a full chart of the worst-case and average time complexities of the proposed algorithms. We would also like to study the case when zero edge weights are allowed and, in particular, develop algorithmic procedures that can identify a strongly connected spanning subdigraph which is doubly stochasticable. To our knowledge, the synthesis of a distributed dynamical system that computes doubly stochastic weight assignments when agents communicate only over the original digraph, and not over the mirror digraph, is still an open problem. Finally, we would like to employ the results of this paper to extend the range of applicability of existing distributed algorithms for coordination, formation control, and optimization tasks.

Acknowledgements

The authors would like to thank Professors David Gregory and Robert Israel for enlightening discussions and the various anonymous referees for valuable observations that greatly improved the presentation of the paper. In particular, we are indebted to a reviewer who suggested the formulation of the doubly stochastic weight assignment problem as a constrained maximum flow problem.

References

[1] R. K Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 2 edition, 1994.

[3] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucumán, Revista A*, 5:147–151, 1946.

[4] C. Bornstein, A. S Ribeiro, and A. Lucena. Maximum flow problems under special nonnegative lower bounds on arc flows. *Electronic Notes in Discrete Mathematics*, 7:66–69, 2005.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[6] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[7] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self Organization in Biological Systems*. Princeton University Press, Princeton, NJ, 2001.

[8] M. Cao and C. W. Wu. Topology design for fast convergence of network consensus algorithms. In *The 2007 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1029–1032, New Orleans, LA, USA, 2007.

[9] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[10] M. C. de Gennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *IEEE Conf. on Decision and Control*, pages 3628–3633, San Diego, CA, December 2006.

[11] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2005.

[12] B. Gharesifard and J. Cortés. Distributed strategies for making a digraph weight-balanced. In *Allerton Conf. on Communications, Control and Computing*, pages 771–777, Monticello, IL, October 2009.

[13] B. Gharesifard and J. Cortés. When does a digraph admit a doubly stochastic adjacency matrix? In *American Control Conference*, pages 2440–2445, Baltimore, MD, June 2010.

[14] A. Ghosh and S. Boyd. Upper bounds on algebraic connectivity via convex optimization. *Linear Algebra and its Applications*, 418:693–707, 2006.

[15] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinary*, 35(4):921–940, 1988.

[16] S. Gueron and S. A. Levin. Self-organization of front patterns in large wildebeest herds. *Journal of Theoretical Biology*, 165:541–552, 1993.

[17] S. Gueron, S. A. Levin, and D. I. Rubenstein. The dynamics of herds: From individuals to aggregations. *Journal of Thoeretical Biology*, 182:85–98, 1996.

[18] L. Hooi-Tong. On a class of directed graphs - with an application to traffic-flow problems. *Operations Research*, 18(1):87–94, 1970.

[19] J. Hu and Y. Hong. Leader-following coordination of multi-agent systems with coupling time delays. *Physica A*, 374(2):853–863, 2007.

[20] B. Johansson, M. Rabi, and M. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Control and Optimization*, 20(3):1157–1170, 2009.

[21] D. Lee and M. W. Spong. Stable flocking of multiple inertial agents on balanced graphs. *IEEE Transactions on Automatic Control*, 52(8):1469–1475, 2007.

[22] J. Liu, A. S. Morse, B. D. O. Anderson, and C. Yu. The contraction coefficient of a complete gossip sequence. *Proceedings of the IEEE*, 2009. Submitted.

[23] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 2 edition, 1984.

[24] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997.

[25] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Applied Mathematics Series. Princeton University Press, 2010.

[26] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[27] A. Okubo. Dynamical aspects of animal grouping: swarms, schools, flocks and herds. *Advances in Biophysics*, 22:1–94, 1986.

[28] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

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

[30] J. K. Parrish, S. V. Viscido, and D. Grunbaum. Self-organized fish schools: an examination of emergent properties. *Biological Bulletin*, 202:296–305, 2002.

[31] D. Peleg. *Distributed Computing. A Locality-Sensitive Approach*. Monographs on Discrete Mathematics and Applications. SIAM, 2000.

[32] T. L. Pham, I. Lavallee, M. Bui, and S. H. Do. A distributed algorithm for the maximum flow problem. In *Proceeding of the* 4*th International Symposium on Parallel and Distributed Computing*, pages 131–138, Lille, France, July 2005.

[33] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communications and Control Engineering. Springer, 2008.

[34] H. Shi, L. Wang, and T. Chu. Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions. *Physica D*, 213(1):51–65, 2006.

[35] S. H. Strogatz. From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1):1–20, 2000.

[36] S. H. Strogatz. *SYNC: The Emerging Science of Spontaneous Order*. Hyperion, 2003.

[37] D. J. T. Sumpter. *Collective Animal Behavior*. Princeton University Press, Princeton, NJ, 2010.

[38] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 2 edition, 2001.

[39] C. W. Wu. Algebraic connectivity of directed graphs. *Linear and Multilinear Algebra*, 53(3):203–223, 2005.

[40] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53:65–78, 2004.

[41] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.