Distributed Balancing in Digraphs under Interval Constraints

Christoforos N. Hadjicostis and Alejandro D. Domínguez-García

Abstract— **We consider networks the nodes of which are interconnected via directed edges, each able to admit a flow within a certain interval, with nonnegative end points that correspond to lower and upper flow limits. The paper proposes and analyzes a distributed algorithm for obtaining *admissable* and *balanced* flows, i.e., flows that are within the given intervals at each edge and are balanced (the total in-flow equals the total out-flow) at each node. The algorithm can also be viewed as a distributed method for obtaining a set of weights that balance a digraph for the case when there are upper and lower limit constraints on the edge weights. The proposed iterative algorithm assumes that communication among pairs of nodes that are interconnected is bidirectional (i.e., the communication topology is captured by the undirected graph that corresponds to the network digraph), and allows the nodes to asymptotically (with geometric rate) reach a set of balanced feasible flows, as long as the circulation conditions on the given digraph, with the given flow/weight interval constraints on each edge, are satisfied.**

I. Introduction

We consider a system comprised of multiple nodes that are interconnected via some directed links through which a certain commodity can flow. We assume that the flow on each link is constrained to lie within an interval the end points of which are nonnegative, corresponding to link upper and lower capacity limits. The objective is to find a feasible flow assignment i.e., find flows on all the links that are within the corresponding capacity limits and balance each node, i.e., the sum of in-flows must be equal to the sum of out-flows. In this paper, we propose a distributed algorithm that allows the nodes to compute a solution to this feasibility problem.

The problem of interest in this paper is a particular case of the standard network flow problem (see, e.g., [1]), where there is a cost associated to the flow on each link, and the objective is to minimize the total cost subject to the same constraints in the flow assignment problem described above. In this regard, it is common to assume that the individual costs are described by convex functions on the flow, which makes the optimization problem convex. Then, its solution can be obtained via the Lagrange dual, the formulation of which is well suited for algorithms that can be executed, in a distributed fashion, over a network that conforms to the same topology as that of the multi-node system (see, e.g., [2]); however, recovering the optimal primal solution from the dual one might not be straightforward [3].

By contrast, the distributed algorithm proposed in this paper does not exploit duality notions, and instead acts directly on the *primal variables*, i.e., the flows. In this regard, it can be shown that the algorithm is a gradient descent algorithm for a quadratic optimization program. In this program, the flows are constrained to lie within the corresponding link upper and lower capacity limits; and the cost function is the two-norm of the projection of the balance vector (the entries of which are the differences of the nodes in- and out-flows) onto the positive orthant. Here is important to note that finding a feasible flow assignment is equivalent to finding a zero-cost solution to this quadratic program. Also, if the solution of the quadratic program has a non-zero cost, then there is no solution to the flow assignment problem. [The quadratic program is always feasible as long as the set defined by lower and upper capacity limits is non-empty.] In terms of establishing convergence of our proposed algorithm, one could attempt to utilize off-the-shelf convergence results for optimization problems (see, e.g., [4]). However, with these results one can only establish optimality of all limits points of the sequence generated by our algorithm, but one cannot establish convergence; to address this issue, we utilize an alternative proof technique.

The problem we deal with in this paper can also be viewed as the problem of weight balancing a given digraph. A weighted digraph is a digraph in which each edge is associated with a real or integer value, called the edge weight. A weighted digraph is *weight-balanced* or *balanced* if, for each of its nodes, the sum of the weights of the edges outgoing from the node is equal to the sum of the weights of the edges incoming to the node. Weight-balanced digraphs find numerous applications in control, optimization, economics and statistics (see, e.g., [5]–[11]). Recently, quite a few works have appeared dealing with the problem of designing distributed algorithms for balancing a strongly connected digraph, for both real- and integer-weight balancing, for the case when there are no constraints on the edge weights in terms of the nonnegative values they admit [9]–[13]. Thus, the main difference in this paper is the presence of interval constraints on the link weights.

II. Mathematical Background and Notation

A distributed system the components of which can exchange a certain commodity via (possibly directed) links, can conveniently be captured by a digraph (directed graph). A digraph of order n ($n \geq 2$), is defined as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$,

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

Alejandro D. Domínguez-García is with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: aledan@illinois.edu.

where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges. A directed edge from node v_i to node v_j is denoted by $(v_j, v_i) \in \mathcal{E}$, and indicates a nonnegative flow from node v_i to node v_j. We will refer to the digraph $\mathcal{G}_d$ as the *flow topology*.

We assume that a pair of nodes v_j and v_i that are connected by an edge in the digraph $\mathcal{G}_d$ (i.e., $(v_j, v_i) \in \mathcal{E}$ and/or $(v_i, v_j) \in \mathcal{E}$) can exchange information among themselves. In other words, the *communication topology* is captured by the undirected graph $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$ that corresponds to a given directed graph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}_u = \cup_{(v_j,v_i)\in\mathcal{E}}\{(v_j, v_i), (v_i, v_j)\} = \mathcal{E} \cup \mathcal{E}_r$, with $\mathcal{E}_r = \{(v_i, v_j) \mid (v_j, v_i) \in \mathcal{E}\}$.

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 $v_i \equiv v_{l_0}, v_{l_1}, \ldots, v_{l_t} \equiv v_j$ such that $(v_{l_{\tau+1}}, v_{l_\tau}) \in \mathcal{E}$ for $\tau = 0, 1, \ldots, t-1$. All nodes from which node v_j can receive flows are said to be in-neighbors of node v_j and belong to the set $\mathcal{N}_j^- = \{v_i \in V \mid (v_j, v_i) \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^-$ is called the *in-degree* of j and is denoted by $\mathcal{D}_j^-$. The nodes that receive flows from node v_j comprise its out-neighbors and are denoted by $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid (v_l, v_j) \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^+$ is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+$. We also let $\mathcal{D}_j = \mathcal{D}_j^+ + \mathcal{D}_j^-$ denote the *total degree* of node v_j.

Given a digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, we can associate nonnegative flows (sometimes, also viewed as weights) $f_{ji} \in \mathbb{R}$ on each edge $(v_j, v_i) \in \mathcal{E}$. In this paper, these flows will be restricted to lie in a real interval $[l_{ji}, u_{ji}]$ where $0 \leq l_{ji} \leq f_{ji} \leq u_{ji}$. We will also use matrix notation to denote (respectively) the flow, lower limit, and upper limit matrices by the $n \times n$ matrices $F = [f_{ji}]$, $L = [l_{ji}]$, and $U = [u_{ji}]$, where $L(j, i) = l_{ji}$, $F(j, i) = f_{ji}$, and $U(j, i) = u_{ji}$ (and $f_{ji} = l_{ji} = u_{ji} = 0$ when $(v_j, v_i) \notin \mathcal{E}$).

Definition 1: Given a digraph $\mathcal{G}_d(\mathcal{V}, \mathcal{E})$ of order n along with a flow assignment $F = [f_{ji}]$, the total *in-flow* of node v_j is denoted by f_j^-, and is defined as $f_j^- = \sum_{v_i\in\mathcal{N}_j^-} f_{ji}$, whereas the total *out-flow* of node v_j is denoted by f_j^+, and is defined as $f_j^+ = \sum_{v_l\in\mathcal{N}_j^+} f_{lj}$.

Definition 2: Given a digraph $\mathcal{G}_d(\mathcal{V}, \mathcal{E})$ of order n, along with a flow assignment $F = [f_{ji}]$, the *flow balance* of node v_j is denoted by b_j and is defined as $b_j = f_j^- - f_j^+$.

Definition 3: Given a digraph $\mathcal{G}_d(\mathcal{V}, \mathcal{E})$ of order n, along with a flow assignment $F = [f_{ji}]$, the *total imbalance* (or *absolute imbalance*) of digraph $\mathcal{G}_d$ is denoted by ε and is defined as $\varepsilon = \sum_{j=1}^n |b_j|$.

Definition 4: A digraph $\mathcal{G}_d(\mathcal{V}, \mathcal{E})$ of order n, along with a flow assignment $F = [f_{ji}]$, is called weight-balanced if its *total imbalance* (or *absolute imbalance*) is 0, i.e., $\varepsilon = \sum_{j=1}^n |b_j| = 0$.

Flow Assignment Problem: We are given a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, as well as lower and upper bounds l_{ji} and u_{ji} $(0 \leq l_{ji} \leq u_{ji})$ on each each edge $(v_j, v_i) \in \mathcal{E}$. We want to develop a distributed algorithm that allows the nodes to iteratively adjust the flows on their outgoing edges so that they eventually obtain a set of flows $\{f_{ji} \mid (v_j, v_i) \in \mathcal{E}\}$ that satisfy the following:

1) $0 \leq l_{ji} \leq f_{ji} \leq u_{ji}$ for each edge $(v_j, v_i) \in \mathcal{E}$;
2) $f_j^+ = f_j^-$ for every $v_j \in \mathcal{V}$.

The distributed algorithm needs to respect the communication constraints imposed by the undirected graph $\mathcal{G}_u$ that corresponds to the given directed graph $\mathcal{G}_d$. □

Remark 1: One of the main differences of the work in this paper with the works in [9]–[13] is that the algorithm developed in this paper requires a bi-directional communication topology, whereas most of the aforementioned works assume a communication topology that matches the flow (physical) topology. We should point out, however, that there are many applications where the physical topology is directed but the communication topology is bi-directional (e.g., traffic flow in an one way street is directional, but communication between traffic lights at the end points of the street will, in fact, be bi-directional). More generally, in many applications, the communication topology does not necessarily match the physical one; in our future work, we plan to enhance the algorithm proposed here to allow for different communication topologies (including the one that matches the physical topology).

If the necessary and sufficient conditions in the theorem below hold, obtaining a set of admissible flows (i.e., balanced and within the given constraints) can be accomplished via a variety of centralized algorithms [1].

Theorem 1: (Circulation Theorem [1]) Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, with lower and upper bounds l_{ji} and u_{ji} $(l_{ji} \leq u_{ji})$ on each edge $(v_j, v_i) \in \mathcal{E}$. The necessary and sufficient condition for the existence of a set of flows $\{f_{ji} \mid (v_j, v_i) \in \mathcal{E}\}$ that satisfy

1. *Interval constraints:* $0 \leq l_{ji} \leq f_{ji} \leq u_{ji}$, $\forall (v_j, v_i) \in \mathcal{E}$,
2. *Balance constraints:* $f_j^+ = f_j^-$, $\forall v_j \in \mathcal{V}$,

is the following: for each $\mathcal{S}$, $\mathcal{S} \subset \mathcal{V}$, we have

$$\sum_{(v_j,v_i)\in\mathcal{E}_\mathcal{S}^-} l_{ji} \leq \sum_{(v_l,v_j)\in\mathcal{E}_\mathcal{S}^+} u_{lj} \tag{1}$$

where

$$\mathcal{E}_\mathcal{S}^- = \{(v_j, v_i) \in \mathcal{E} \mid v_j \in \mathcal{S},\ v_i \in \mathcal{V} - \mathcal{S}\}\ , \tag{2}$$

$$\mathcal{E}_\mathcal{S}^+ = \{(v_l, v_j) \in \mathcal{E} \mid v_j \in \mathcal{S},\ v_l \in \mathcal{V} - \mathcal{S}\}\ . \tag{3}$$

In the remainder of this paper, we assume that the above circulation conditions hold for a given directed graph and aim at developing a distributed algorithm for allowing the nodes to assign flows, such that both the interval constraints and the balance constraints are satisfied. We believe that the algorithm we develop can also be used to distributively *check* whether or not the above conditions hold, but this is not something we explicitly address in this paper.

III. Distributed Flow Algorithm

The distributed algorithm we develop is iterative, and we use k to denote the iteration. For example, $f_j^+[k]$ will denote the value of the total out-flow of node v_j at iteration k, $k \in \mathbb{N}_0$. A pseudocode description of the algorithm is not included due to space limitations.

Overview. The algorithm is iterative and operates by having, at each iteration, nodes with positive imbalance attempt to change the flows on both their incoming and outgoing edges, so as to get closer to being balanced. In the process, since the flow on each edge affects the balance of *two* nodes (both of which may be simultaneously attempting to adjust the edge flow), the nodes need to coordinate with the corresponding neighbor (whether an in-neighbor or an out-neighbor) in order to reach an agreement on the flow for that particular edge. Naturally, the nodes need to assign flows that respect the lower and upper limits on each edge.

Initialization. At initialization, each node is aware of the feasible flow interval on each of its incoming and outgoing edges, i.e., node v_j is aware of l_{ji}, u_{ji} for each $v_i \in \mathcal{N}_j^-$ and l_{lj}, u_{lj} for each $v_l \in \mathcal{N}_j^+$. Furthermore, the flows are initialized at the middle of the feasible interval, i.e., $f_{ji}[0] = (l_{ji} + u_{ji})/2$. (This initialization is not critical and could be any value in the feasible flow interval $[l_{ji}, u_{ji}]$.)

Iteration. At each iteration $k \geq 0$, node v_j is aware of the flows on its incoming edges $\{f_{ji}[k] \mid v_i \in \mathcal{N}_j^-\}$ and outgoing edges $\{f_{lj}[k] \mid v_l \in \mathcal{N}_j^+\}$, and updates them using the following three steps:

[Step 1.] Each node v_j attempts to change the flows in both its incoming edges and its outgoing edges. The way this is done depends on whether the node has a positive imbalance or not. We discuss both cases below and then describe how to concisely capture both.

1. Nodes with positive imbalance: If $b_j[k] > 0$, node v_j attempts to change the flows at both its incoming edges $\{f_{ji}[k+1] \mid v_i \in \mathcal{N}_j^-\}$, and outgoing edges $\{f_{lj}[k+1] \mid v_l \in \mathcal{N}_j^+\}$ in a way that drives its balance $b_j[k+1]$ to zero (at least if no other changes are inflicted on the flows). More specifically, since node v_j is associated with $\mathcal{D}_j = \mathcal{D}_j^- + \mathcal{D}_j^+$ edges, it attempts to change each incoming flow by $-\frac{b_j[k]}{\mathcal{D}_j}$ and each outgoing flow by $+\frac{b_j[k]}{\mathcal{D}_j}$, i.e., from the perspective of node v_j, the desirable flows at the next iteration are

$$f_{ji}^{(j)}[k+1] = f_{ji}[k] - \frac{b_j[k]}{\mathcal{D}_j} , \quad v_i \in \mathcal{N}_j^- , \tag{4}$$

$$f_{lj}^{(j)}[k+1] = f_{lj}[k] + \frac{b_j[k]}{\mathcal{D}_j} , \quad v_l \in \mathcal{N}_j^+ , \tag{5}$$

where $b_j[k] > 0$. Note that if the above changes on the flows were adopted, then the new balance of node v_j would be

$$\begin{aligned} b_j^{(j)}[k+1] &\equiv \textstyle\sum_{v_i \in \mathcal{N}_j^-} f_{ji}^{(j)}[k+1] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}^{(j)}[k+1] \\ &= \textstyle\sum_{v_i \in \mathcal{N}_j^-} (f_{ji}[k] - \frac{b_j[k]}{\mathcal{D}_j}) - \sum_{v_l \in \mathcal{N}_j^+} (f_{lj}[k] + \frac{b_j[k]}{\mathcal{D}_j}) \\ &= b_j[k] - \mathcal{D}_j^- \frac{b_j[k]}{\mathcal{D}_j} - \mathcal{D}_j^+ \frac{b_j[k]}{\mathcal{D}_j} = 0 . \end{aligned}$$

2. Nodes with non-positive imbalance. If node v_j has balance $b_j[k]$ that is negative or zero ($b_j[k] \leq 0$), then node v_j does not attempt to make any flow changes.

Note that no desirable change on the flows can also be captured by (4)–(5) with $b_j[k] = 0$. Thus, regardless of whether node v_j has positive imbalance or not, we can capture the desirable new flows on each incoming and outgoing edge as

$$f_{ji}^{(j)}[k+1] = f_{ji}[k] - \frac{\tilde{b}_j[k]}{\mathcal{D}_j} , \quad v_i \in \mathcal{N}_j^- , \tag{6}$$

$$f_{lj}^{(j)}[k+1] = f_{lj}[k] + \frac{\tilde{b}_j[k]}{\mathcal{D}_j} , \quad v_l \in \mathcal{N}_j^+ , \tag{7}$$

where $\tilde{b}_j[k]$ is defined as

$$\tilde{b}_j[k] = \begin{cases} b_j[k] , & \text{if } b_j[k] > 0 , \\ 0 , & \text{otherwise.} \end{cases}$$

[Step 2.] Since the flow f_{ji} on each edge $(v_j, v_i) \in \mathcal{E}$ affects positively the balance $b_j[k]$ of node v_j and negatively the flow of node v_i, we need to account for the possibility of both nodes attempting to inflict changes on the flows. Thus, the new flow on each edge $(v_j, v_i) \in \mathcal{E}$ is taken to be

$$\begin{aligned} \tilde{f}_{ji}[k+1] &= \frac{1}{2} \left(f_{ji}^{(j)}[k] + f_{ji}^{(i)}[k] \right) \\ &= f_{ji}[k] + \frac{1}{2} \left(\frac{\tilde{b}_i[k]}{\mathcal{D}_i} - \frac{\tilde{b}_j[k]}{\mathcal{D}_j} \right) . \end{aligned} \tag{8}$$

[Step 3.] If the above value is in interval $[l_{ji}, u_{ji}]$, then $f_{ji}[k+1] = \tilde{f}_{ji}[k+1]$; otherwise, if it is above u_{ji} (respectively, below l_{ji}), it is set to the upper bound u_{ji} (respectively, to the lower bound l_{ji}):

$$f_{ji}[k+1] = \begin{cases} \tilde{f}_{ji}[k+1], & \text{if } l_{ji} \leq \tilde{f}_{ji}[k+1] \leq u_{ji} , \\ u_{ji}, & \text{if } \tilde{f}_{ji}[k+1] > u_{ji} , \\ l_{ji}, & \text{if } \tilde{f}_{ji}[k+1] < l_{ji} . \end{cases} \tag{9}$$

Once the values $\{f_{ji}[k+1] \mid (v_j, v_i) \in \mathcal{E}\}$ are obtained, the iteration is repeated.

Example 1: In this example, we illustrate the operation of the algorithm described by (4)–(9) for a randomly generated (strongly connected) digraph of $n = 7$ nodes with interval constraints on its edges. It was verified that the circulation conditions in Theorem 1 were satisfied. On the left of Fig. 1, we plot the flow balance, $b_j[k]$, $j = 1, 2, ..., 7$, of each of the seven nodes against the iteration k (notice that the sum $\sum_{j=1}^{7} b_j[k]$ is identically zero for all k as expected–this is established in the next section). We observe that nodes with a positive flow balance retain a positive flow balance as k increases and in the end only one node retains a negative balance, with all flows asymptotically going to zero (this is also something we establish in the next section). In the middle of Fig. 1 we plot the evolution of the total imbalance $\varepsilon[k]$ against the iteration k. Notice that $\varepsilon[k]$ monotonically goes to zero (this is a key result in our proof of convergence in the next section). Finally, on the right of Fig. 1, we plot the values of the flows $f_{ji}[k]$ for each $(v_j, v_i) \in \mathcal{E}$. $\square$

IV. Proof of Convergence

Setting. We are given a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ of order $n \geq 2$, with lower and upper bounds l_{ji} and u_{ji} ($l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. The algorithm described by (4)–(9) is executed iteratively for steps $k = 0, 1, 2, ...$

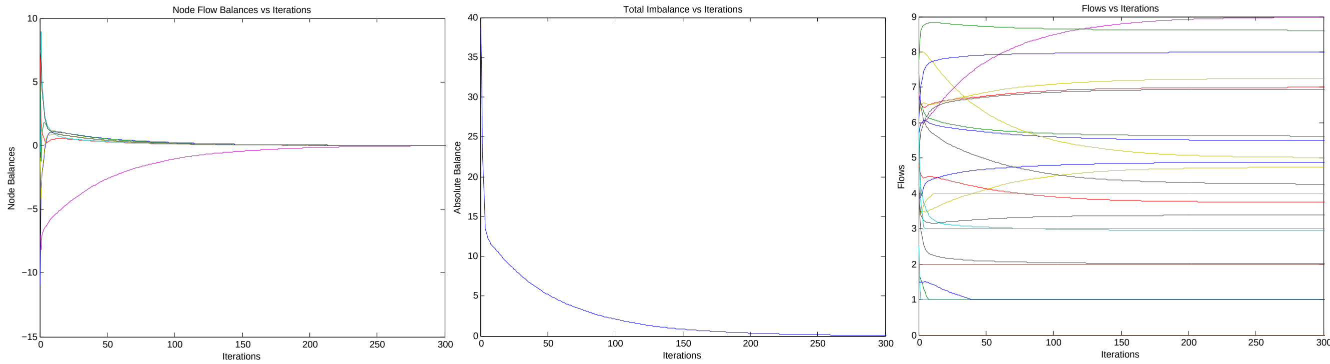

Fig. 1: Shown as a function of the iteration k for the digraph used in the example: (i) Left: node flow balances $b_j[k]$, $j = 1, 2, ..., 7$; Middle: total imbalance $\varepsilon[k]$; Right: flow values $f_{ji}[k]$, for $(v_j, v_i) \in \mathcal{E}$.

Definition 5: The flow change incurred at the flow of edge $(v_j, v_i) \in \mathcal{E}$ at iteration k is denoted by $\Delta f_{ji}[k]$, i.e.,

$$\Delta f_{ji}[k] := f_{ji}[k+1] - f_{ji}[k] \ , \tag{10}$$

a variable that captures the combined effect of both (8) and (9). Similarly, we define the changes in the flow balances at each node $v_j \in \mathcal{V}$ and the total imbalance of the network:

$$\begin{aligned} \Delta b_j[k] &\equiv b_j[k+1] - b_j[k] \ , \ \forall v_j \in \mathcal{V} \ , \\ \Delta \varepsilon[k] &\equiv \varepsilon[k+1] - \varepsilon[k] \ . \end{aligned}$$

The bulk of this section will be devoted to proving the following theorem.

Theorem 2: Consider the setting described above (where, in particular, the circulation conditions in Theorem 1 hold). During the execution of the algorithm described by (4)–(9), it holds that $\varepsilon[k+n] \leq (1-c)\varepsilon[k] \ , \quad \forall k \geq 0 \ ,$ where $\varepsilon[k] \geq 0$ is the total imbalance of the network at iteration k (refer to Definition 3), with $c = \frac{1}{2n}\left(\frac{1}{2\mathcal{D}_{\max}}\right)^n$, where $\mathcal{D}_{\max} = \max_{v_j \in \mathcal{V}} \mathcal{D}_j$. [Note that $\mathcal{D}_{\max}$ necessarily satisfies $1 \leq \mathcal{D}_{\max} \leq 2(n-1)$.]

Corollary 1: Consider the setting described at the beginning of this section. The execution of the algorithm described by (4)–(9) asymptotically leads to a set of flows $\{f^*_{ji} \mid (v_j, v_i) \in \mathcal{E}\}$ that satisfy the interval constraints and balance constraints, i.e., we have $\lim_{k\to\infty} f_{ji}[k] = f^*_{ji} \ , \ \forall (v_j, v_i) \in \mathcal{E}$, where the set of flows $\{f^*_{ji} \mid (v_j, v_i) \in \mathcal{E}\}$ satisfy

1) $l_{ji} \leq f^*_{ji} \leq u_{ji}$, $\forall (v_j, v_i) \in \mathcal{E}$;
2) $\sum_{v_i \in \mathcal{N}^-_j} f^*_{ji} = \sum_{v_l \in \mathcal{N}^+_j} f^*_{lj}$, $\forall v_j \in \mathcal{V}$.

Proof: From Theorem 2, we have that $\lim_{k\to\infty} \varepsilon[k] = \lim_{k\to\infty} \sum_{j=1}^n |b_j[k]| = 0$, which implies that $\lim_{k\to\infty} b_j[k] = 0 \ , \ \ \forall v_j \in \mathcal{V}$. From the flow updates in (8) and (9), the flow $f_{ji}[k]$ on each edge $(v_j, v_i) \in \mathcal{E}$ stabilizes to a value f^*_{ji}, i.e., $f^*_{ji} = \lim_{k\to\infty} f_{ji}[k]$ exist for all edges $(v_j, v_i) \in \mathcal{E}$. Clearly, the algorithm described by (4)–(9) results in flows f^*_{ji} that are within the upper and lower bounds on each edge (i.e., $l_{ji} \leq f^*_{ji} \leq u_{ji}$). Furthermore, since $\lim_{k\to\infty} b_j[k] = 0$, we easily obtain that $\sum_{v_i \in \mathcal{N}^-_j} f^*_{ji} = \sum_{v_l \in \mathcal{N}^+_j} f^*_{lj}$. ■

We next state some propositions that are useful for proving the main result in Theorem 2; we do not provide proofs for these propositions due to space limitations.

Proposition 1: Consider the setting described at the beginning of this section. At each iteration k during the execution of the algorithm described by (4)–(9), it holds that

1) For any subset of nodes $\mathcal{S} \subset \mathcal{V}$, let $\mathcal{E}^-_{\mathcal{S}}$ and $\mathcal{E}^+_{\mathcal{S}}$ be defined by (2) and (3) respectively. Then, $\sum_{v_j \in \mathcal{S}} b_j[k] = \sum_{(v_j, v_i) \in \mathcal{E}^-_{\mathcal{S}}} f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}^+_{\mathcal{S}}} f_{lj}[k]$;
2) $\sum_{j=1}^n b_j[k] = 0$;
3) $\varepsilon[k] = 2\sum_{v_j \in \mathcal{V}^+[k]} b_j[k]$ where $\mathcal{V}^+[k] = \{v_j \in \mathcal{V} \mid b_j[k] > 0\}$.

Proposition 2: Consider the setting described in the beginning of this section. Let $\mathcal{V}^+[k] \subset \mathcal{V}$ be the set of nodes with positive flow balance at iteration k, i.e., $\mathcal{V}^+[k] = \{v_j \in \mathcal{V} \mid b_j[k] > 0\}$. During the execution of the algorithm described by (4)–(9), we have the following:

1) $b_j[k+1] \geq \frac{1}{2} b_j[k] > 0$, for all $v_j \in \mathcal{V}^+[k]$;
2) $\mathcal{V}^+[k] \subseteq \mathcal{V}^+[k+1]$.

Proposition 3: Consider the setting described at the beginning of this section. During the execution of the algorithm described by (4)–(9), it holds that $0 \leq \varepsilon[k+1] \leq \varepsilon[k]$.

Proposition 4: Consider the setting described at the beginning of this section. At time step k of the execution of the algorithm described by (4)–(9), let $\mathcal{S} = \mathcal{V}^+[k] \subset \mathcal{V}$ be the set of nodes with positive flow balance at iteration k (i.e., $\mathcal{V}^+[k] = \{v_j \in \mathcal{V} \mid b_j[k] > 0\}$) and let $\overline{\mathcal{S}} = \mathcal{V} - \mathcal{S}$ be the remaining nodes (with zero or negative flow balance). Define $\mathcal{E}^-_{\mathcal{S}}$ and $\mathcal{E}^+_{\mathcal{S}}$ as in (2) and (3) respectively, and let $\mathcal{T} \subseteq \overline{\mathcal{S}}$ be the subset of nodes in $\overline{\mathcal{S}}$ directly connected to nodes in $\mathcal{S}$ (i.e., $\mathcal{T} = \{v_j \in \overline{\mathcal{S}} \mid \exists v_i \text{ s.t. } (v_i, v_j) \in \mathcal{E}^-_{\mathcal{S}} \text{ or } (v_j, v_i) \in$

$\mathcal{E}_{\mathcal{S}}^+\}$). We have

$$\begin{aligned}\Delta\varepsilon[k] &= 2\left(\sum_{(v_j,v_i)\in\mathcal{E}_{\mathcal{S}}^-}\Delta f_{ji}[k] - \sum_{(v_l,v_j)\in\mathcal{E}_{\mathcal{S}}^+}\Delta f_{lj}[k]\right) + \\ &\quad + \sum_{v_j\in\mathcal{T}}(|b_j[k+1]| + b_j[k+1]) \qquad (11)\\ &= \sum_{v_j\in\mathcal{T}}\Delta\epsilon_j[k]\ , \qquad (12)\end{aligned}$$

where

$$\begin{aligned}\Delta\varepsilon_j[k] &= |b_j[k+1]| + b_j[k+1] + \\ &\quad -\sum_{v_i\in\mathcal{N}_j^-\cap\mathcal{S}} 2\Delta f_{ji}[k] + \sum_{v_l\in\mathcal{N}_j^+\cap\mathcal{S}} 2\Delta f_{lj}[k] \\ &\leq 0\ . \qquad (13)\end{aligned}$$

We are now ready to move with the proof of Theorem 2.

Proof of Theorem 2: Consider the execution of the algorithm described by (4)–(9) under the setting described at the beginning of this section. At iteration k, let $\varepsilon[k]$ be the total imbalance of the network. Let $v_{j_{\max}} \in \mathcal{V}^+[k]$ be the node with the maximum (positive) flow balance at iteration k. It follows from the third statement of Proposition 1 that $b_{j_{\max}}[k] \geq \frac{\varepsilon[k]}{2|\mathcal{V}^+[k]|} \geq \frac{\varepsilon[k]}{2n}$ (a tighter lower bound would have been $\varepsilon[k]/(2(n-1)$ but it is more convenient to use the above); therefore, for all $t = 0, 1, 2, ...,$ we have (from the first statement of Proposition 2)

$$b_{j_{\max}}[k+t] \geq \left(\frac{1}{2}\right)^t \frac{\varepsilon[k]}{2n} \geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^t \frac{\varepsilon[k]}{2n}\ .$$

Note that there also exists a node $v_{j_{\min}}$ with the minimum (negative) flow balance at iteration k whose flow balance satisfies $b_{j_{\min}}[k] \leq -\frac{\varepsilon[k]}{2|\mathcal{V}-\mathcal{V}^+[k]|} \leq -\frac{\varepsilon[k]}{2n}$.

We recursively define the sets of nodes V_k, V_{k+1}, V_{k+2}, ..., V_{k+n-1}, all of which are subsets of $\mathcal{V}$:

1. $V_k = \{v_{j_{\max}}\}$
2. For $t = 1, 2, ..., n-2$, we let

$$V_{k+t} = V_{k+t-1} \cup V_{k+t-1}^+ \cup V_{k+t-1}^-$$

where

$$\begin{aligned}V_{k+t-1}^+ &= \{v_l \in \mathcal{V} \mid \exists v_j \in V_{k+t-1} \text{ s.t. } (v_l, v_j) \in \mathcal{E} \\ &\qquad \text{and } \tilde{f}_{lj}[k+t] \leq u_{lj}\}\ , \\ V_{k+t-1}^- &= \{v_i \in \mathcal{V} \mid \exists v_j \in V_{k+t-1} \text{ s.t. } (v_j, v_i) \in \mathcal{E} \\ &\qquad \text{and } \tilde{f}_{ji}[k+t] \geq l_{ji}\}\ .\end{aligned}$$

For $t = 0, 1, 2, ..., n-1$, consider the inequality

$$\left(\frac{1}{2\mathcal{D}_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - g[t] > 0\ , \qquad (14)$$

where $g[t] \equiv \varepsilon[k] - \varepsilon[k+t] \geq 0$ is the *gain* in the total imbalance after t iterations. Note that if the above inequality is violated at some $t_0 \in \{1, 2, ..., n-1\}$ (without loss of generality, let t_0 be the smallest such integer when the inequality is violated for the first time), then we have

$$g[t_0] \geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t_0} \frac{\varepsilon[k]}{2n}\ ,$$

which implies that

$$\begin{aligned}\varepsilon[k+t_0] &\leq \varepsilon[k] - \left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t_0}\frac{\varepsilon[k]}{2n} \qquad (15)\\ &\leq \left(1 - \frac{1}{2n}\left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t_0}\right)\varepsilon[k]\ , \qquad (16)\end{aligned}$$

which immediately leads to the proof of the theorem (since, by Proposition 3, $\varepsilon[k+n] \leq \varepsilon[k+t_0]$ for $n \geq t_0$).

We will argue, by contradiction, that the inequality in (14) gets violated for the first time at some $t_0 \in \{0, 1, 2, ..., n-1\}$, which will establish our proof. Suppose that the inequality (14) holds for all $t \in \{0, 1, 2, ..., n-1\}$. Then, we argue below that each node v_j in the set V_{k+t}, $t \in \{0, 1, 2, ..., n-1\}$, has flow balance that satisfies

$$b_j[k+t] \geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - g[t] > 0 \qquad (17)$$

(this is established at the end of the proof). Assuming (for now) that (17) holds, we have

$$\sum_{v_j\in V_{k+t}} b_j[k+t] > 0\ , \forall t \in \{0, 1, ..., n-1\}\ . \qquad (18)$$

Since, by construction, we have

$$V_k \subseteq V_{k+1} \subseteq V_{k+2} \subseteq ... \subseteq V_{k+n-1} \subseteq \mathcal{V}$$

and $|\mathcal{V}| = n$, we need to have $V_{k+t} = V_{k+t-1}$ for some $t \in \{1, 2, ..., n-1\}$. Then, we have two possibilities, both of which lead to a contradiction:

(1) $V_{k+t} = \mathcal{V}$, which immediately leads to a contradiction in (18) (because $\sum_{v_j\in\mathcal{V}} b_j[k] = 0$ for all k by the second statement of Proposition 1).

(2) If $V_{k+t} \subset \mathcal{V}$, let $\mathcal{S} = V_{k+t} = V_{k+t-1}$ and define $\mathcal{E}_{\mathcal{S}}^-$ and $\mathcal{E}_{\mathcal{S}}^+$ as in (2) and (3) respectively. Then, from the recursive definition of V_{k+t} we have

$$\begin{aligned}f_{ji}[k+t] &= l_{ji}\ , \forall (v_j, v_i) \in \mathcal{E}_{\mathcal{S}}^-\ , \\ f_{lj}[k+t] &= u_{lj}\ , \forall (v_l, v_j) \in \mathcal{E}_{\mathcal{S}}^+\ .\end{aligned}$$

[Note that both $\mathcal{E}_{\mathcal{S}}^+$ and $\mathcal{E}_{\mathcal{S}}^-$ are nonempty sets; otherwise, the given graph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ would not be strongly connected. Furthermore, if the upper (respectively, lower) limits were not reached for edges in $\mathcal{E}_{\mathcal{S}}^+$ (respectively, $\mathcal{E}_{\mathcal{S}}^-$), the set V_{k+t} would strictly contain V_{k+t-1}.] Thus, from the first statement of Proposition 1, we have

$$\begin{aligned}&\textstyle\sum_{v_j\in\mathcal{S}} b_j[k+t] = \\ &\quad = \textstyle\sum_{(v_j,v_i)\in\mathcal{E}_{\mathcal{S}}^-} f_{ji}[k+t] - \sum_{(v_l,v_j)\in\mathcal{E}_{\mathcal{S}}^+} f_{lj}[k+t] \\ &\quad = \textstyle\sum_{(v_j,v_i)\in\mathcal{E}_{\mathcal{S}}^-} l_{ji} - \sum_{(v_l,v_j)\in\mathcal{E}_{\mathcal{S}}^+} u_{lj}\ .\end{aligned}$$

Since, all nodes in V_{k+t} have strictly positive balance, we have $\sum_{(v_j,v_i)\in\mathcal{E}_{\mathcal{S}}^-} l_{ji} - \sum_{(v_l,v_j)\in\mathcal{E}_{\mathcal{S}}^+} u_{lj} > 0$, which contradicts the circulation conditions in Theorem 1.

We now move to argue that if the inequality (14) holds for $t \in \{0, 1, ..., n-1\}$, then inequality (17) also holds for $t \in \{0, 1, ..., n-1\}$. The proof is by induction. Clearly, the inequality holds for $t = 0$. Suppose that (17) holds for $k+t$, i.e., for all $v_j \in V_{k+t}$, we have

$$b_j[k+t] \geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - g[t] > 0 \ ,$$

where $g[t] \equiv \varepsilon[k] - \varepsilon[k+t]$ is the gain in the total balance after t iterations. We need to argue that

$$b_j[k+t+1] \geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t+1} \frac{\varepsilon[k]}{2n} - g[t+1] > 0$$

for all $v_j \in \mathcal{V}_{k+t+1}$.

Since $g[t+1] \geq g[t] \geq 0$ (follows from Proposition 3) and $b_j[k+1] \geq \frac{1}{2} b_j[k]$ for nodes with positive flow balance (from the first statement of Proposition 2), the above trivially holds for nodes in the set V_{k+t} (which necessarily belong in the set V_{k+t+1}). Let us now consider nodes in the set $V_{k+t+1} - V_{k+t}$, which (by construction of the set V_{k+t+1}) have to necessarily share edges with nodes in the set V_{k+t}. Thus, we consider three possibilities:
Case 1: $v_l \in V_{k+t+1} - V_{k+t}$, such that there exists at least one $v_j \in V_{k+t}$ with $(v_l, v_j) \in \mathcal{E}$ and $\tilde{f}_{lj}[k+t+1] \leq u_{lj}$;
Case 2: $v_i \in V_{k+t+1} - V_{k+t}$, such that there exists at least one $v_j \in V_{k+t}$ with $(v_j, v_i) \in \mathcal{E}$ and $\tilde{f}_{ji}[k+t+1] \geq l_{lj}$;
Case 3: A combination of the above two cases.

We focus on Case 1 since Cases 2 and 3 can be treated similarly. We have two possibilities to consider: (i) $b_l[k+t] > 0$ and (ii) $b_l[k+t] \leq 0$.
(i) If $b_l[k+t] > 0$, then $\tilde{f}_{lj}[k+t+1] = f_{lj}[k+t] + \frac{b_j[k+t]}{2\mathcal{D}_j} - \frac{b_l[k+t]}{2\mathcal{D}_l}$ and, since $\tilde{f}_{lj}[k+t+1] \leq u_{lj}$ (by construction of V_{k+t+1}), we have $f_{lj}[k+t+1] \geq f_{lj}[k+t] + \frac{b_j[k+t]}{2\mathcal{D}_j} - \frac{b_l[k+t]}{2\mathcal{D}_l}$. Consider the flow balance of node v_l at iteration $k+t+1$. Using an argument similar to the proof of the first statement of Proposition 2, but also taking into account the flow $f_{lj}[k+t+1]$, we have

$$\begin{aligned}
b_l[k+t+1] &\geq \frac{1}{2} b_l[k+t] + \frac{b_j[k+t]}{2\mathcal{D}_j} \\
&\geq \frac{1}{2\mathcal{D}_j}\left(\frac{1}{2\mathcal{D}_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - \frac{1}{2\mathcal{D}_j} g[t] \\
&\geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t+1} \frac{\varepsilon[k]}{2n} - g[t] \\
&\geq \left(\frac{1}{2\mathcal{D}_{\max}}\right)^{t+1} \frac{\varepsilon[k]}{2n} - g[t+1] > 0 \ ,
\end{aligned}$$

where in the second line we used the induction hypothesis and the fact that $b_l[k+t] > 0$, in the third line we used the fact that $\mathcal{D}_{\max} \geq \mathcal{D}_j > 0$ and $g[t] \geq 0$, and in the fourth line we used the fact that $g[t+1] \geq g[t]$. Notice that the last quantity is greater than zero since we are assuming that inequality (14) holds for $t \in \{0, 1, ..., n-1\}$.
(ii) If $b_l[k+t] \leq 0$, then $\tilde{f}_{lj}[k+t+1] = f_{lj}[k+t] + \frac{b_j[k+t]}{2\mathcal{D}_j}$ and, since $\tilde{f}_{lj}[k+t+1] \leq u_{lj}$ (by construction of V_{k+t+1}), we have $f_{lj}[k+t+1] \geq f_{lj}[k+t] + \frac{b_j[k+t]}{2\mathcal{D}_j}$ or, equivalently $\Delta f_{lj}[k+t] \geq \frac{b_j[k+t]}{2\mathcal{D}_j} > 0$.

Since $v_l \in \mathcal{T}$ in the proof of Proposition 4, we can use (13) (and the fact that $\Delta\varepsilon_l[k] \leq 0$ for $v_l \in \mathcal{T}$) to establish that

$$\Delta\varepsilon_l[k+t] \leq |b_l[k+t+1]| + b_l[k+t+1] - 2\frac{b_j[k+t]}{2\mathcal{D}_j} \ , \quad (19)$$

where the second inequality follows from (13) and the fact that $\Delta f_{lj}[k] \geq \frac{b_j[k+t]}{2\mathcal{D}_j}$. [Recall that changes in the flows on edges in the first summation in (13) are nonnegative whereas changes in the flows on edges in the second summation in (13) are nonpositive.]

There are two possibilities to consider: (a) $b_l[k+t+1] \leq 0$ and (b) $b_l[k+t+1] > 0$, both of which lead to the desired conclusion (we do not provide the arguments due to space limitations). This completes the proof of Theorem 2. ■

V. Concluding Remarks

In this paper, we introduced and analyzed a distributed algorithm for assigning balanced flows, within specified intervals, in a given digraph. In the future, we plan to explore methodologies for allowing the nodes to distributively identify when such flows are not feasible. We also plan to investigate ways of relaxing the assumption regarding the necessity of bi-directional communication between neighboring nodes is bi-directional (even though flows might be directional).

References

[1] L. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton University Press, 2010.
[2] D. P. Bertsekas, P. A. Hosein, and P. Tseng, "Relaxation methods for network flow problems with convex arc costs," *SIAM Journal of Control and Optimization*, vol. 25, no. 5, pp. 1219–1243, 1987.
[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.
[4] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
[5] M. H. Schneider and S. A. Zenios, "A comparative study of algorithms for matrix balancing," *Operations research*, vol. 38, no. 3, pp. 439–455, 1990.
[6] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, Massachusetts: Athena Scientic, 1998.
[7] P. DeLellis, M. di Bernardo, F. Garofalo, and M. Porfiri, "Evolution of complex networks via edge snapping," *IEEE Transactions on Circuits and Systems*, vol. 57, no. 8, pp. 2132–2143, August 2010.
[8] W. Yu, P. DeLellis, G. Chen, M. di Bernardo, and J. Kurths, "Distributed adaptive control of synchronization in complex networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2153–2158, August 2012.
[9] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539–557, 2012.
[10] A. D. Domínguez-García 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.
[11] A. Rikos, T. Charalambous, and C. N. Hadjicostis, "Distributed weight balancing over digraphs," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 190–201, May 2014.
[12] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, "A decentralized algorithm for balancing a strongly connected weighted digraph," in *Proc. of American Control Conference*, June 2013, pp. 6547–6552.
[13] C. N. Hadjicostis and A. I. Rikos, "Distributed strategies for balancing a weighted digraph," in *Proc. of Mediterranean Conference on Control Automation*, July 2012, pp. 1141–1146.