

Distributed Balancing of Commodity Networks Under Flow Interval Constraints

Christoforos N. Hadjicostis, *Senior Member, IEEE*, and Alejandro D. Domínguez-García, *Member, IEEE*

Abstract—We consider networks the nodes of which are interconnected via directed edges, each able to admit a flow (or weight) 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 *admissible* 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 weighted digraph for the case when there are lower and upper 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 (necessary and sufficient) circulation conditions on the given digraph, with the given flow/weight interval constraints on each edge, are satisfied. We also provide a methodology that can be used by the nodes to asymptotically determine, in a distributed manner, when the circulation conditions are not satisfied (thus, making the problem infeasible). Finally, we provide several examples and simulation studies to highlight the role of the various parameters involved in the proposed distributed algorithm.

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 lower and upper 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 is equal to the sum of out-flows. In this paper, we propose an iterative algorithm that allows the nodes to distributively compute a solution to this feasibility problem.

The work of C. N. Hadjicostis has been supported in part by the European Commission (EC) through a grant for KIOS Center of Excellence in Research and Innovation. The work of A. D. Domínguez-García has been partially funded by the U.S. Department of Energy (DOE) within the GEARED Initiative, under grant DE-0006341; and the Advanced Research Projects Agency-Energy (ARPA-E) within the NODES program, under Award DE-AR0000695. Any opinions, findings, and conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of EC or DOE or ARPA-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.

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 such settings, 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 lower and upper 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, it 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 nonzero 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 that relies on showing that (i) the set of nodes with a positive balance enlarges monotonically (in a way that includes all nodes that previously had positive balance) as the algorithm progresses, reaching a set that includes all but one vertex, as long as a feasible flow assignment exists; and (ii) the sum of positive balances decreases monotonically. Then, since the sum of the balances must always be zero, this means that all balances will converge to zero asymptotically as long as a feasible flow assignment exists.

The proposed distributed algorithm for flow balancing is not aimed at replacing (centralized) approaches that are used

to balance the flows in a given network at design time. Instead, the application setting we have in mind is one where the network operation is highly dynamic and the proposed algorithm needs to be distributed and invoked (perhaps at multiple instances) during system operation. Examples of systems where such need arises include electric power distribution systems, where the commodity that flows along the links in the physical network is electric power [5], [6], [7]; upper and lower capacity limits on these types of edges correspond to upper and lower limits on electric power loads and generators, which may change over time. Link failures is another example of a change in the nominal operating conditions, which is very typical in electric power distribution systems. After such changes occur, the network is no longer balanced and the generators need to adjust the power they provide so as to re-balance the network.

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 positive 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. Examples of applications where balance plays a key role include modeling of flocking behavior [8], network adaptation and synchronization strategies [9], distributed adaptive strategies to tune the coupling weights of a network based on local information of node dynamics [10], prediction of distribution matrices for telephone traffic [11], and financial transactions [12]. It is also worth pointing out that weight balance is closely related to weights that form a doubly stochastic matrix [13], which find applications in multicomponent systems (such as sensor networks) where one is interested in distributively averaging measurements at each component. In particular, the distributed average consensus problem has received significant attention from the computer science community (see, e.g., [14]), and the control community (see, e.g., [15]) due to its applicability to diverse areas, including multi-agent systems, distributed estimation and tracking [16], distributed optimization [17], and coordination of distributed energy resources in electrical energy systems [6], [18].

Parts of the results in this paper appeared in [19]. This extended version provides complete proofs of the various theorems and propositions, a distributed algorithm for determining feasibility, as well as simulations/comparisons against previous work; all of these were not included in [19].

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 [13]–[19]. Thus, the contribution of this paper with respect to this earlier work is the development of pertinent distributed algorithms in the presence of interval constraints on the link weights. We should point out that, unlike [13]–[23], in which it is (indirectly) assumed that the communication topology

matches the flow (physical) topology, the algorithm developed in this paper requires a bidirectional communication topology (i.e., the assumptions on the communication topology are more restrictive, but the assumptions on the physical topology are relaxed). Nevertheless, the applicability of the proposed algorithm is not inhibited as there are many applications where the physical topology is directed but the communication topology is bidirectional. For example, traffic flow in an one way street of a transportation network is directional, but communication between traffic lights at the end points of the street (to regulate/balance the flows) could, in fact, be bidirectional. Other examples include water networks, where flows in certain links may be restricted in specific directions but regulation could rely on bidirectional communication between the points.

The remainder of this paper is organized as follows. Section II provides some background on graph-theoretic notions used throughout the paper, the formulation of the constrained flow assignment problem in commodity networks, and a well-known result on necessary and sufficient conditions for the existence of solutions to the constrained flow assignment problem. In Section III, we propose a distributed algorithm to solve the aforementioned constrained flow balancing problem and state the main convergence results. Section IV establishes some ancillary results that are needed for the proof of convergence of the proposed algorithm, which is provided in Section V. In Section VI, we provide an enhancement to the proposed algorithm that allows nodes to distributively detect whether or not there is a feasible solution to the constrained flow balancing problem. Simulation results showcasing the algorithms are presented in Section VII. Concluding remarks are presented in Section VIII.

II. MATHEMATICAL BACKGROUND AND NOTATION

In this section, we first provide some graph-theoretic notions used throughout the paper. Then, we formulate the constrained flow feasibility problem, i.e., the problem of finding a set of flows in a commodity network that satisfy some equality constraints associated to each node, and some interval constraints associated with edge capacity limits. We finish the section by introducing a well-known result in the network flow literature pertaining necessary and sufficient conditions for the existence of solutions to the constrained flow feasibility problem.

A. Graph-Theoretic Notions

A digraph (directed graph) of order n ($n \geq 2$), is defined as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ 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*. Note that the definition of \mathcal{G}_d excludes self-edges.

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 digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, 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 $\mathcal{E}_r = \{(v_i, v_j) \mid (v_j, v_i) \in \mathcal{E}\}$. [Recall that a graph is undirected if and only if $(v_j, v_i) \in \mathcal{E}$ implies $(v_i, v_j) \in \mathcal{E}$.]

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 $v_i \equiv v_{l_0}, v_{l_1}, \dots, v_{l_t} \equiv v_j$ such that $(v_{l_{\tau+1}}, v_{l_\tau}) \in \mathcal{E}$ for $\tau = 0, 1, \dots, t-1$. All nodes from which node v_j can receive flows are said to be in-neighbors of node v_j and belong to the set $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$. 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*¹ of node v_j , and $\mathcal{N}_j = \mathcal{N}_j^+ \cup \mathcal{N}_j^-$ denote the *neighbors* of node v_j .

B. Network Flows and Problem Formulation

A flow commodity network can be described by a digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, with nonnegative flows (sometimes, also viewed as weights) $f_{ji} \in \mathbb{R}$ associated with each edge $(v_j, v_i) \in \mathcal{E}$. In this paper, these flows will be restricted to lie in a real interval $[l_{ji}, u_{ji}]$, $0 < l_{ji} \leq f_{ji} \leq u_{ji}$. 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 $F(j, i) = f_{ji}$, $L(j, i) = l_{ji}$, and $U(j, i) = u_{ji}$ (and $F(j, i) = L(j, i) = U(j, i) = 0$ 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 *absolute imbalance* (or *total 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 *flow-balanced* (or *weight-balanced*) if its *absolute imbalance* (or *total imbalance*) is 0, i.e., $\varepsilon = \sum_{j=1}^n |b_j| = 0$.

With the definitions above, we can now formally state the flow assignment problem we are interested in solving in a distributed manner.

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} ($0 < l_{ji} \leq u_{ji}$) on 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 properties:

¹Note that the total degree \mathcal{D}_j counts twice each node that is both an in-neighbor and an out-neighbor of node v_j . Thus, in a graph of n nodes, \mathcal{D}_j is bounded by $2(n-1)$.

P1. $0 < l_{ji} \leq f_{ji} \leq u_{ji}$ for each edge $(v_j, v_i) \in \mathcal{E}$;

P2. $f_j^+ = f_j^-$ 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 digraph \mathcal{G}_d . (i.e., communication between nodes is bidirectional). \square

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 obtained 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} ($0 < l_{ji} \leq u_{ji}$) 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:* $0 < l_{ji} \leq f_{ji} \leq u_{ji}$, $\forall (v_j, v_i) \in \mathcal{E}$,

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

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

where

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

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

In the remainder of this paper, we assume that the above circulation conditions hold for a given directed graph, and propose a distributed algorithm for allowing the nodes to find a flow assignment satisfying Properties P1 and P2. Additionally, we discuss an enhancement to the algorithm that allows the nodes to distributively determine whether or not the conditions in Theorem 1 hold.

III. DISTRIBUTED FLOW ALGORITHM

In this section we first describe the distributed iterative algorithm that we propose to solve the constrained flow balancing problem. Then, we state the main results pertaining the convergence of the algorithm.

A. Algorithm Description

Each node maintains and iteratively updates estimates of the flows on its incoming and outgoing edges so as to attempt to balance itself, i.e., make the sum of the flow estimates on incoming edges equal to the sum of the flow estimates on outgoing edges. In the process, since any *two* connected nodes might be attempting to simultaneously adjust their flow estimate on the edge that connects them, they need to coordinate among themselves in order to reach an agreement on the flow estimate for that particular edge. We emphasize, once again, that this coordination between neighboring nodes is feasible because communication between nodes that are connected in the (physical) topology is assumed to be possible in both directions (bidirectional). Naturally, the nodes need to assign flows that respect the lower and upper limits on each edge; this, also needs to be taken into account by the nodes when adjusting the flow estimates they maintain.

Let $f_{ji}^{(j)}[k]$, $v_i \in \mathcal{N}_j^-$, and $f_{lj}^{(j)}[k]$, $v_l \in \mathcal{N}_j^+$, respectively denote the flow estimates on incoming and outgoing edges maintained by node v_j at iteration k . These estimates are updated by node v_j based on its flow estimate imbalance at iteration k , denoted by $b_j^{(j)}[k]$, and defined as

$$b_j^{(j)}[k] \equiv \sum_{v_i \in \mathcal{N}_j^-} f_{ji}^{(j)}[k] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}^{(j)}[k]. \quad (4)$$

Initialization: Initially, each node v_j 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 $v_l \in \mathcal{N}_j^+$. Then, each node v_j initializes the flow estimates it maintains at the middle of the feasible interval, i.e.,

$$\begin{aligned} f_{ji}^{(j)}[0] &= \frac{l_{ji} + u_{ji}}{2}, \quad v_i \in \mathcal{N}_j^- \\ f_{lj}^{(j)}[0] &= \frac{l_{lj} + u_{lj}}{2}, \quad v_l \in \mathcal{N}_j^+. \end{aligned} \quad (5)$$

[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 updates the flow estimates on its incoming edges, $f_{ji}^{(j)}[k]$, $v_i \in \mathcal{N}_j^-$, and outgoing edges, $f_{lj}^{(j)}[k]$, $v_l \in \mathcal{N}_j^+$, by using Steps S1-S3 below.

[S1.] Each node v_j attempts to change its flow estimates of both incoming and outgoing edges. The way this is done depends on whether or not the node has a positive flow estimate imbalance or not. We discuss both cases below and then describe how to concisely capture both.

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

$$\hat{f}_{ji}^{(j)}[k+1] = f_{ji}^{(j)}[k] - \frac{b_j^{(j)}[k]}{\mathcal{D}_j}, \quad v_i \in \mathcal{N}_j^-, \quad (6)$$

$$\hat{f}_{lj}^{(j)}[k+1] = f_{lj}^{(j)}[k] + \frac{b_j^{(j)}[k]}{\mathcal{D}_j}, \quad v_l \in \mathcal{N}_j^+, \quad (7)$$

where $b_j^{(j)}[k] > 0$. Note that $\hat{f}_{ji}^{(j)}[k+1]$ and $\hat{f}_{lj}^{(j)}[k+1]$ are the flows that node v_j desires on its incoming and outgoing edges at iteration $k+1$; however, they are based solely on its own balance and Step [S2] (below) will also take into account the desired flows of its neighbors.

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

Note that no desirable change on the flow estimates can also be captured by (6)–(7) with $b_j^{(j)}[k] = 0$. Thus, regardless of whether node v_j has positive imbalance or not, we can capture the desirable new flow estimates on each incoming and outgoing edge as

$$\hat{f}_{ji}^{(j)}[k+1] = f_{ji}^{(j)}[k] - \frac{\tilde{b}_j^{(j)}[k]}{\mathcal{D}_j}, \quad v_i \in \mathcal{N}_j^-, \quad (8)$$

$$\hat{f}_{lj}^{(j)}[k+1] = f_{lj}^{(j)}[k] + \frac{\tilde{b}_j^{(j)}[k]}{\mathcal{D}_j}, \quad v_l \in \mathcal{N}_j^+, \quad (9)$$

where $\tilde{b}_j^{(j)}[k]$ is defined as

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

[S2.] Since the flow estimate $f_{ji}^{(j)}[k]$ on edge $(v_j, v_i) \in \mathcal{E}$ as maintained by v_j affects positively $b_j^{(j)}[k]$ (maintained by node v_j), whereas the flow estimate $f_{ji}^{(i)}[k]$ on edge $(v_j, v_i) \in \mathcal{E}$ as maintained by v_i affects negatively $b_i^{(i)}[k]$ (maintained by node v_i), we need to account for the possibility of both nodes attempting to inflict changes on their flow estimates. Thus, the new flow estimate on each edge $(v_j, v_i) \in \mathcal{E}$ as maintained by node v_j is taken to be

$$\tilde{f}_{ji}^{(j)}[k+1] = \frac{1}{2} \left(\hat{f}_{ji}^{(j)}[k+1] + \hat{f}_{ji}^{(i)}[k+1] \right). \quad (10)$$

Note that this step requires bidirectional exchange of values between pairs of nodes that are physically connected. Also note that $\tilde{f}_{ji}^{(j)}[k+1] = \tilde{f}_{ji}^{(i)}[k+1]$.

[S3.] If the value in (10) is within the interval $[l_{ji}, u_{ji}]$, then $f_{ji}^{(j)}[k+1] = \tilde{f}_{ji}^{(j)}[k+1]$; 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}):

$$f_{ji}^{(j)}[k+1] = \begin{cases} \tilde{f}_{ji}^{(j)}[k+1], & \text{if } l_{ji} \leq \tilde{f}_{ji}^{(j)}[k+1] \leq u_{ji}, \\ u_{ji}, & \text{if } \tilde{f}_{ji}^{(j)}[k+1] > u_{ji}, \\ l_{ji}, & \text{if } \tilde{f}_{ji}^{(j)}[k+1] < l_{ji}. \end{cases} \quad (11)$$

It follows from the initialization of the algorithm in (5), the averaging operation in (10), and the projection operation in (11), that $f_{ji}^{(j)}[k] = f_{ji}^{(i)}[k]$, $\forall k \geq 0$. [Note that even if $f_{ji}^{(j)}[0] \neq f_{ji}^{(i)}[0]$, as long as $f_{ji}^{(j)}[0] \in [l_{ij}, u_{ij}]$ and $f_{ji}^{(i)}[0] \in [l_{ij}, u_{ij}]$, after Step S3, we will have that $f_{ji}^{(j)}[1] = f_{ji}^{(i)}[1]$; thus, in this case, $f_{ji}^{(j)}[k] = f_{ji}^{(i)}[k]$, $\forall k > 0$.] Therefore, by defining $f_{ji}[k] \equiv f_{ji}^{(j)}[k] = f_{ji}^{(i)}[k]$, $\forall k \geq 0$, the progress of the algorithm in Steps S1-S3 can be summarized by a single iteration of the form

$$f_{ji}[k+1] = \left[f_{ji}[k] + \frac{1}{2} \left(\frac{\tilde{b}_i[k]}{\mathcal{D}_i} - \frac{\tilde{b}_j[k]}{\mathcal{D}_j} \right) \right]_{l_{ji}}^{u_{ji}}, \quad \forall (v_j, v_i) \in \mathcal{E}, \quad (12)$$

where $[\cdot]_{\underline{x}}^{\bar{x}}$ denotes the projection onto $[\underline{x}, \bar{x}]$; and

$$\tilde{b}_j[k] = \begin{cases} b_j[k], & \text{if } b_j[k] > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

Algorithm 1: Distributed feasible flow algorithm

Each node $v_j \in \mathcal{V}$ **does the following:**

Input: $l_{ji}, u_{ji}, \forall v_i \in \mathcal{N}_j^-$

Input: $l_{lj}, u_{lj}, \forall v_l \in \mathcal{N}_j^+$

Output: $f_{ji}^{(j)}, \forall v_i \in \mathcal{N}_j^-$

Output: $f_{lj}^{(j)}, v_l \in \mathcal{N}_j^+$

begin

Set $f_{ji}^{(j)}[0] = \frac{l_{ji} + u_{ji}}{2}, \forall v_i \in \mathcal{N}_j^-$

Set $f_{lj}^{(j)}[0] = \frac{l_{lj} + u_{lj}}{2}, \forall v_l \in \mathcal{N}_j^+$

Set $\mathcal{D}_j = \mathcal{D}_j^- + \mathcal{D}_j^+$

foreach iteration, $k = 0, 1, \dots$, **do**

Calculate: $b_j^{(j)}[k] = \sum_{v_i \in \mathcal{N}_j^-} f_{ji}^{(j)}[k] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}^{(j)}[k]$

Set: $\tilde{b}_j^{(j)}[k] = \begin{cases} b_j^{(j)}[k], & \text{if } b_j^{(j)}[k] > 0, \\ 0, & \text{otherwise.} \end{cases}$

Transmit: $\frac{\tilde{b}_j^{(j)}[k]}{\mathcal{D}_j}$ to $v_i \in \mathcal{N}_j^-$ and $v_l \in \mathcal{N}_j^+$

Receive: $\frac{\tilde{b}_i^{(i)}[k]}{\mathcal{D}_i}$ from all $v_i \in \mathcal{N}_j^-$, and

$\frac{\tilde{b}_l^{(l)}[k]}{\mathcal{D}_l}$ from all $v_l \in \mathcal{N}_j^+$

Calculate:

$\tilde{f}_{ji}^{(j)}[k+1] = f_{ji}^{(j)}[k] + \frac{1}{2} \left(\frac{\tilde{b}_i^{(i)}[k]}{\mathcal{D}_i} - \frac{\tilde{b}_j^{(j)}[k]}{\mathcal{D}_j} \right), v_i \in \mathcal{N}_j^-$

$\tilde{f}_{lj}^{(j)}[k+1] = f_{lj}^{(j)}[k] + \frac{1}{2} \left(\frac{\tilde{b}_j^{(j)}[k]}{\mathcal{D}_j} - \frac{\tilde{b}_l^{(l)}[k]}{\mathcal{D}_l} \right), v_l \in \mathcal{N}_j^+$

Set:

$f_{ji}^{(j)}[k+1] = \begin{cases} \tilde{f}_{ji}^{(j)}[k+1], & \text{if } \tilde{f}_{ji}^{(j)}[k+1] \in [l_{ji}, u_{ji}] \\ u_{ji}, & \text{if } \tilde{f}_{ji}^{(j)}[k+1] > u_{ji} \\ l_{ji}, & \text{if } \tilde{f}_{ji}^{(j)}[k+1] < l_{ji} \end{cases}$

$f_{lj}^{(j)}[k+1] = \begin{cases} \tilde{f}_{lj}^{(j)}[k+1], & \text{if } \tilde{f}_{lj}^{(j)}[k+1] \in [l_{lj}, u_{lj}] \\ u_{lj}, & \text{if } \tilde{f}_{lj}^{(j)}[k+1] > u_{lj} \\ l_{lj}, & \text{if } \tilde{f}_{lj}^{(j)}[k+1] < l_{lj} \end{cases}$

with

$$b_j[k] = \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}[k]. \quad (14)$$

The compact description of the algorithm progress given in (12)–(14) greatly simplifies the notation and is convenient for convergence analysis purposes. It also highlights the fact that for distributed implementation purposes, each node v_j can transmit its $\tilde{b}_j[k]/\mathcal{D}_j$ at iteration k to all of its in- and out-neighbors, as opposed to having to transmit $\tilde{f}_{ji}^{(j)}[k+1]$ to every $v_i \in \mathcal{N}_j^-$, and $\tilde{f}_{lj}^{(j)}[k+1]$ to every $v_l \in \mathcal{N}_j^+$. [Recall that the communication graph is assumed to be given by the undirected graph \mathcal{G}_u that corresponds to the digraph \mathcal{G}_d .] Then, nodes v_j and v_i can essentially use (12)–(14) to calculate $f_{ji}^{(i)}[k+1] = f_{ji}^{(j)}[k+1] = f_{ji}[k+1]$ for each $(v_j, v_i) \in \mathcal{E}$ in their immediate neighborhood. The pseudocode for the iterative procedure described in (5)–(11), also implementing the modification described above, is provided in Algorithm 1.

Example 1: In this example, we illustrate the operation of Algorithm 1 for a randomly generated (strongly connected)

digraph with seven nodes. The adjacency matrix A for the digraph is shown in Fig. 1, along with the lower and upper bound matrices L and U , for the edges in this graph. [Recall that the $n \times n$ adjacency matrix A of a digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ has $A(j, i) = 1$ if $(v_j, v_i) \in \mathcal{E}$, otherwise $A(j, i) = 0$.] It can be verified that the circulation conditions in Theorem 1 are satisfied.

On the left of Fig. 2, we plot the flow balance, $b_j[k]$, $j = 1, 2, \dots, 7$, of each of the seven nodes against the iteration k of the distributed flow balancing algorithm (Algorithm 1). Notice that the sum $\sum_{j=1}^7 b_j[k]$ is identically equal to 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; in the end, only one node retains a negative balance, and all node balances asymptotically go to zero (this is also something we establish in the next section). In the middle of Fig. 2, we plot the evolution of the absolute (total) imbalance $\varepsilon[k]$ against the iteration k . Notice that $\varepsilon[k]$ monotonically goes to zero (again, this is a key result in our proof of convergence in the next section).

On the right of Fig. 2, we plot the values of the flows $f_{ji}[k]$ for each $(v_j, v_i) \in \mathcal{E}$. It can be observed that the flows eventually stabilize to fixed values, given by the matrix of flows as follows:

$$F = \begin{bmatrix} 0 & 4.8848 & 5.4988 & 0 & 0 & 1 & 7.9926 \\ 5.6152 & 0 & 0 & 0 & 1 & 0 & 8.6078 \\ 7.0012 & 0 & 0 & 0 & 2 & 3.7488 & 0 \\ 0 & 2.9461 & 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 \\ 4.7525 & 4 & 7.2512 & 0 & 5 & 0 & 0 \\ 2.0074 & 3.3922 & 0 & 6.9461 & 0 & 4.2549 & 0 \end{bmatrix},$$

which are easily seen to be feasible and result in a balanced digraph. \square

B. Convergence Results

Next, we state the main convergence results of the paper; namely that Algorithm 1 converges to a set of feasible and balanced flows, as long as the necessary and sufficient condition in (1) holds. In particular, the absolute (total) imbalance $\varepsilon[k]$ in Definition 3 goes to zero as k goes to infinity, with a geometric rate.² This, in turn, implies that the flow balance $b_j[k]$ for each node $v_j \in \mathcal{V}$ goes to zero; therefore, $f_{ji}[k] = f_{ji}^{(j)}[k] = f_{ji}^{(i)}[k]$, $\forall (v_j, v_i) \in \mathcal{E}$, converge to f_{ji}^* , $\forall (v_j, v_i) \in \mathcal{E}$, as k goes to infinity, such that $l_{ji} \leq f_{ji}^* \leq u_{ji}$, $\forall (v_j, v_i) \in \mathcal{E}$. The proofs of these results is deferred to Sections IV and V.

Theorem 2: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. During the execution of (12)–(14), with the initial conditions in (5), it holds that

$$\varepsilon[k+n] \leq (1-c)\varepsilon[k], \quad \forall k \geq 0,$$

²A variant of the bound in Theorem 2 is sharp, at least for certain topologies and initial flow values.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad L = \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 1 & 1 \\ 3 & 0 & 0 & 0 & 1 & 0 & 3 \\ 3 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 2 & 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 3 & 0 & 0 \\ 2 & 3 & 0 & 1 & 0 & 3 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & 5 & 9 & 0 & 0 & 2 & 10 \\ 11 & 0 & 0 & 0 & 3 & 0 & 12 \\ 12 & 0 & 0 & 0 & 3 & 9 & 0 \\ 0 & 9 & 0 & 0 & 4 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 & 0 \\ 6 & 4 & 9 & 0 & 13 & 0 & 0 \\ 7 & 6 & 0 & 10 & 0 & 13 & 0 \end{bmatrix}$$

Fig. 1. Adjacency matrix A , matrix of lower bounds L , and matrix of upper bounds U , for the digraph used in Example 1.

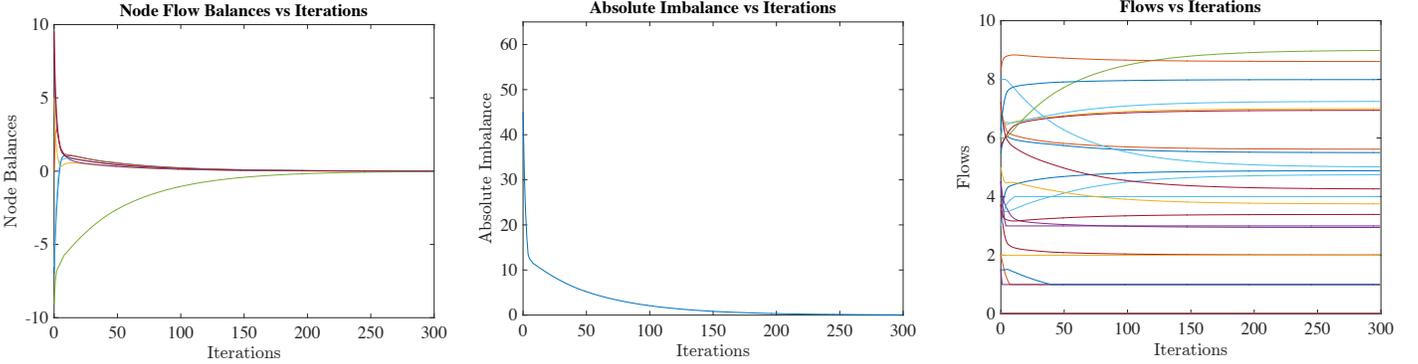


Fig. 2. Shown as a function of the iteration k for the digraph used in the Example 1: Left: node flow balances $b_j[k]$, $j = 1, 2, \dots, 7$; Middle: absolute (total) imbalance $\varepsilon[k]$; Right: flow values $f_{ji}[k]$, for $(v_j, v_i) \in \mathcal{E}$.

where $\varepsilon[k] \geq 0$ is the absolute (total) imbalance of the network at iteration k (refer to Definition 3), with

$$c = \frac{1}{2n} \left(\frac{1}{2D_{\max}} \right)^n,$$

where $D_{\max} = \max_{v_j \in \mathcal{V}} D_j$. [Note that D_{\max} necessarily satisfies $1 \leq D_{\max} \leq 2(n-1)$.]

Corollary 1: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. The execution of (12)–(14), with the initial conditions in (5) 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 $\lim_{k \rightarrow \infty} f_{ji}[k] = f_{ji}^*$, $\forall (v_j, v_i) \in \mathcal{E}$, 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) $\sum_{v_i \in \mathcal{N}_j^-} f_{ji}^* = \sum_{v_l \in \mathcal{N}_j^+} f_{lj}^*$, $\forall v_j \in \mathcal{V}$.

IV. ANCILLARY RESULTS

In this section, we establish several results that are utilized in Section V for proving the main result in Theorem 2. For easing subsequent developments, we make use of the following notation.

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.,

$$\Delta f_{ji}[k] \equiv f_{ji}[k+1] - f_{ji}[k], \quad (15)$$

a variable that captures the combined effect of both (10) and (11). Similarly, we define the changes in the flow balances at

each node $v_j \in \mathcal{V}$ and the absolute (total) imbalance of the network:

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

The following proposition establishes some basic relationships governing the flow node balances of sets of nodes and the absolute (total) imbalance of the network.

Proposition 1: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. At each iteration k during the execution of (12)–(14), with the initial conditions in (5), it holds that

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

Proof 1: To prove the first statement, let

$$\mathcal{E}_{\mathcal{S}} = \{(v_j, v_i) \in \mathcal{E} \mid v_j \in \mathcal{S}, v_i \in \mathcal{S}\}$$

be the set of edges that are internal to the set \mathcal{S} . From the definition of the flow balance for node v_j , we have (after re-

arranging the summations)

$$\begin{aligned}
\sum_{v_j \in \mathcal{S}} b_j[k] &= \sum_{v_j \in \mathcal{S}} \left(\sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k] - \sum_{v_i \in \mathcal{N}_j^+} f_{lj}[k] \right) \\
&= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} f_{lj}[k] + \\
&\quad + \sum_{(v_j, v_i) \in \mathcal{E}_S} f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S} f_{lj}[k] \\
&= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} f_{lj}[k].
\end{aligned}$$

For the second statement, we can take any $\mathcal{S} \subset \mathcal{V}$ and argue (using the first statement twice, once for \mathcal{S} and once for $\mathcal{V}-\mathcal{S}$ and the facts that $\mathcal{E}_S^+ = \mathcal{E}_{\mathcal{V}-\mathcal{S}}^-$ and $\mathcal{E}_S^- = \mathcal{E}_{\mathcal{V}-\mathcal{S}}^+$)

$$\sum_{v_j \in \mathcal{V}} b_j[k] = \sum_{v_j \in \mathcal{S}} b_j[k] + \sum_{v_j \in \mathcal{V}-\mathcal{S}} b_j[k] = 0.$$

For the third statement, notice that, from the definition of $\varepsilon[k]$ in Definition 3, we obtain

$$\begin{aligned}
\varepsilon[k] &= \sum_{v_j \in \mathcal{V}} |b_j[k]| = \sum_{v_j \in \mathcal{V}^+[k]} |b_j[k]| + \sum_{v_j \in \mathcal{V}-\mathcal{V}^+[k]} |b_j[k]| \\
&= \sum_{v_j \in \mathcal{V}^+[k]} b_j[k] - \sum_{v_j \in \mathcal{V}-\mathcal{V}^+[k]} b_j[k] = 2 \sum_{v_j \in \mathcal{V}^+[k]} b_j[k],
\end{aligned}$$

utilizing the definition of $\mathcal{V}^+[k]$ (all nodes have positive balance) and the second statement of this proposition. \square

The following proposition establishes certain monotonic properties governing the nodes that have positive flow balance.

Proposition 2: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. During the execution of (12)–(14), with the initial conditions in (5), 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) $\mathcal{V}^+[k] \subseteq \mathcal{V}^+[k+1]$.

Proof 2: Consider a node $v_j \in \mathcal{V}^+[k]$ with balance $b_j[k] > 0$. Suppose for now that (i) all neighbors of node v_j do not belong in $\mathcal{V}^+[k]$ (i.e., $\mathcal{N}_j \cap \mathcal{V}^+[k] = \emptyset$) and that (ii) during the update of the flows (following (10)), the flows on each edge of node v_j are within the lower and upper limits. Then, since nodes outside the set $\mathcal{V}^+[k]$ possess a b of zero, it follows that the flows are updated as

$$\begin{aligned}
f_{ji}[k+1] &= f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}, \forall v_i \in \mathcal{N}_j^-, \\
f_{lj}[k+1] &= f_{lj}[k] + \frac{b_j[k]}{2\mathcal{D}_j}, \forall v_l \in \mathcal{N}_j^+.
\end{aligned}$$

Thus, the flow balance of node v_j satisfies

$$\begin{aligned}
b_j[k+1] &= \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k+1] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}[k+1] \\
&= \sum_{v_i \in \mathcal{N}_j^-} \left(f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j} \right) - \sum_{v_l \in \mathcal{N}_j^+} \left(f_{lj}[k] + \frac{b_j[k]}{2\mathcal{D}_j} \right) \\
&= -\mathcal{D}_j^- \frac{b_j[k]}{2\mathcal{D}_j} - \mathcal{D}_j^+ \frac{b_j[k]}{2\mathcal{D}_j} + b_j[k] \\
&= -\mathcal{D}_j \frac{b_j[k]}{2\mathcal{D}_j} + b_j[k] = \frac{1}{2}b_j[k].
\end{aligned}$$

For the general case (when node v_j may have neighbors with positive flow balance and/or the flows reach the lower or upper flow limits on the corresponding edges), we make the following observations:

1. If an in-neighbor $v_i \in \mathcal{N}_j^-$ has positive flow balance, the flow $\tilde{f}_{ji}[k+1]$ will satisfy $\tilde{f}_{ji}[k+1] \geq f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}$ (from (10)). Clearly, if the flow $\tilde{f}_{ji}[k+1]$ is within the lower and upper limits (in (11)), we have $f_{ji}[k+1] = \tilde{f}_{ji}[k+1] \geq f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}$. Even if the flows are not within the lower and upper limits, we still have

$$f_{ji}[k+1] \geq f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}, \forall v_i \in \mathcal{N}_j^-.$$

To see this, notice that:

- (i) If $\tilde{f}_{ji}[k+1] > u_{ji}$, then $f_{ji}[k+1] = u_{ji} \geq f_{ji}[k] \geq f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}$ (because $l_{ji} \leq f_{ji}[k] \leq u_{ji}$).
- (ii) If $\tilde{f}_{ji}[k+1] < l_{ji}$, then $f_{ji}[k+1] = l_{ji} > \tilde{f}_{ji}[k+1] \geq f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}$.

2. Similar arguments (but reversed) can be used to establish that if an out-neighbor $v_l \in \mathcal{N}_j^+$ has positive flow balance, the flow $f_{lj}[k+1]$ will satisfy

$$f_{lj}[k+1] \leq f_{lj}[k] + \frac{b_j[k]}{2\mathcal{D}_j}, \forall v_l \in \mathcal{N}_j^+.$$

Thus, we conclude that in the general case we still have $b_j[k+1] \geq \sum_{v_i \in \mathcal{N}_j^-} (f_{ji}[k] - \frac{b_j[k]}{2\mathcal{D}_j}) - \sum_{v_l \in \mathcal{N}_j^+} (f_{lj}[k] + \frac{b_j[k]}{2\mathcal{D}_j}) = -\mathcal{D}_j \frac{b_j[k]}{2\mathcal{D}_j} + b_j[k] = \frac{1}{2}b_j[k]$.

The second statement in the proposition follows trivially from the first: a positive node remains positive; thus, the set $\mathcal{V}^+[k]$ can only be enlarged. \square

The following proposition establishes that the absolute (total) imbalance of the network is monotonically non-increasing.

Proposition 3: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. During the execution of (12)–(14), with the initial conditions in (5), it holds that $0 \leq \varepsilon[k+1] \leq \varepsilon[k]$.

Proof 3: Let $\mathcal{V}^+[k] \subset \mathcal{V}$ be the set of nodes with positive flow balance at iteration k , i.e., $\mathcal{V}^+[k] = \{v_j \in \mathcal{V} \mid b_j[k] > 0\}$ (note that $\mathcal{V}^+[k]$ has to be a strict subset of \mathcal{V} , e.g., because of the second statement of Proposition 1). Taking $\mathcal{S} = \mathcal{V}^+[k] \subset \mathcal{V}$, define \mathcal{E}_S^- and \mathcal{E}_S^+ as in (2) and (3) respectively, and let $\mathcal{E}_S = \{(v_j, v_i) \in \mathcal{E} \mid v_j \in \mathcal{S}, v_i \in \mathcal{S}\}$ be the set of edges that are internal to the set \mathcal{S} . Note that $\Delta f_{ji}[k]$ necessarily satisfies (see the discussion in the beginning of the proof of Proposition 2):

$$\begin{aligned}
\Delta f_{ji}[k] &\leq 0, \forall (v_j, v_i) \in \mathcal{E}_S^-, \\
\Delta f_{ji}[k] &\geq 0, \forall (v_j, v_i) \in \mathcal{E}_S^+, \\
\Delta f_{ji}[k] &= 0, \forall (v_j, v_i) \in \mathcal{E} - (\mathcal{E}_S^+ \cup \mathcal{E}_S^- \cup \mathcal{E}_S).
\end{aligned}$$

Since all nodes in $\mathcal{S} = \mathcal{V}^+[k]$ have positive flow balance at iteration k , and all remaining nodes have non-positive flow

balance at iteration k , we have

$$\begin{aligned}\varepsilon[k] &= \sum_{v_j \in \mathcal{V}} |b_j[k]| = \sum_{v_j \in \mathcal{S}} b_j[k] + \sum_{v_j \in \mathcal{V}-\mathcal{S}} |b_j[k]| \\ &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} f_{lj}[k] + \sum_{v_j \in \mathcal{V}-\mathcal{S}} |b_j[k]|,\end{aligned}$$

where in the last line we utilized the result of the first statement of Proposition 1.

Similarly, since from Proposition 2, it holds $\mathcal{S} = \mathcal{V}^+[k] \subseteq \mathcal{V}^+[k+1]$, we have

$$\begin{aligned}\varepsilon[k+1] &= \sum_{v_j \in \mathcal{S}} b_j[k+1] + \sum_{v_j \in \mathcal{V}-\mathcal{S}} |b_j[k+1]| \\ &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} f_{ji}[k+1] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} f_{lj}[k+1] \\ &\quad + \sum_{v_j \in \mathcal{V}-\mathcal{S}} |b_j[k+1]|.\end{aligned}$$

Putting the two above equations together, we obtain:

$$\begin{aligned}\Delta\varepsilon[k] &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] + \\ &\quad + \sum_{v_j \in \mathcal{V}-\mathcal{S}} (|b_j[k+1]| - |b_j[k]|).\end{aligned}\quad (16)$$

For $v_j \in \mathcal{V} - \mathcal{S}$, we have

$$\begin{aligned}b_j[k+1] &= \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k+1] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}[k+1] \\ &= \sum_{v_i \in \mathcal{N}_j^-} (f_{ji}[k] + \Delta f_{ji}[k]) - \sum_{v_l \in \mathcal{N}_j^+} (f_{lj}[k] + \Delta f_{lj}[k]) \\ &= b_j[k] + \sum_{v_i \in \mathcal{N}_j^-} \Delta f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+} \Delta f_{lj}[k]\end{aligned}$$

and by the triangle inequality

$$|b_j[k+1]| \leq |b_j[k]| + \sum_{v_i \in \mathcal{N}_j^-} |\Delta f_{ji}[k]| + \sum_{v_l \in \mathcal{N}_j^+} |\Delta f_{lj}[k]|.$$

Thus, the last term in (16) satisfies

$$\begin{aligned}&\sum_{v_j \in \mathcal{V}-\mathcal{S}} (|b_j[k+1]| - |b_j[k]|) \\ &\leq \sum_{v_j \in \mathcal{V}-\mathcal{S}} \left(\sum_{v_i \in \mathcal{N}_j^-} |\Delta f_{ji}[k]| + \sum_{v_l \in \mathcal{N}_j^+} |\Delta f_{lj}[k]| \right) \\ &\leq \sum_{(v_j, v_i) \in \mathcal{E}_S^-} |\Delta f_{ji}[k]| + \sum_{(v_l, v_j) \in \mathcal{E}_S^+} |\Delta f_{lj}[k]|,\end{aligned}$$

where the last line follows from the fact that the edges, the flows of which change and are incident with nodes in the set $\mathcal{V} - \mathcal{S}$, are the edges in \mathcal{E}_S^- and \mathcal{E}_S^+ (the other flows in question do not change).

Going back to (16), we have that

$$\begin{aligned}\Delta\varepsilon[k] &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] + \\ &\quad + \sum_{v_j \in \mathcal{V}-\mathcal{S}} (|b_j[k+1]| - |b_j[k]|) \\ &\leq \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] + \\ &\quad + \sum_{(v_j, v_i) \in \mathcal{E}_S^-} |\Delta f_{ji}[k]| + \sum_{(v_l, v_j) \in \mathcal{E}_S^+} |\Delta f_{lj}[k]| \\ &= 0,\end{aligned}$$

where the last line follows from the fact that $\Delta f_{ji}[k] \leq 0$ $\forall (v_j, v_i) \in \mathcal{E}_S^-$, and $\Delta f_{lj}[k] \geq 0$, $\forall (v_l, v_j) \in \mathcal{E}_S^+$. \square

The following proposition is a refinement of Proposition 3. It basically states that at each iteration k of the algorithm, described by (5)–(11), the change in the absolute (total) imbalance of the network depends exclusively on (i) the changes in flows on edges that connect positive nodes in $\mathcal{V}^+[k]$ and negative nodes in $\mathcal{V} - \mathcal{V}^+[k]$, and (ii) the changes in the flow balances of nodes with non-positive balance that are directly connected with nodes with positive flow balance.

Proposition 4: Consider 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} ($0 < l_{ji} \leq u_{ji}$) on each edge $(v_j, v_i) \in \mathcal{E}$, such that the necessary and sufficient condition in (1) holds. At time step k of the execution of (12)–(14), with the initial conditions in (5), 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 $\bar{\mathcal{S}} = \mathcal{V} - \mathcal{S}$ be the remaining nodes (with zero or negative flow balance). Define \mathcal{E}_S^- and \mathcal{E}_S^+ as in (2) and (3) respectively, and let $\mathcal{T} \subseteq \bar{\mathcal{S}}$ be the subset of nodes in $\bar{\mathcal{S}}$ directly connected to nodes in \mathcal{S} (i.e., $\mathcal{T} = \{v_j \in \bar{\mathcal{S}} \mid \exists v_i \text{ s.t. } (v_i, v_j) \in \mathcal{E}_S^- \text{ or } (v_j, v_i) \in \mathcal{E}_S^+\}$). We have

$$\begin{aligned}\Delta\varepsilon[k] &= 2 \left(\sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] \right) + \\ &\quad + \sum_{v_j \in \mathcal{T}} (|b_j[k+1]| + b_j[k+1])\end{aligned}\quad (17)$$

$$= \sum_{v_j \in \mathcal{T}} \Delta\varepsilon_j[k],\quad (18)$$

where

$$\begin{aligned}\Delta\varepsilon_j[k] &= |b_j[k+1]| + b_j[k+1] + \\ &\quad - \sum_{v_i \in \mathcal{N}_j^- \cap \mathcal{S}} 2\Delta f_{ji}[k] + \sum_{v_l \in \mathcal{N}_j^+ \cap \mathcal{S}} 2\Delta f_{lj}[k] \\ &\leq 0.\end{aligned}\quad (19)$$

Proof 4: From (16) in the proof of Proposition 3, we have

$$\begin{aligned}\Delta\varepsilon[k] &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] + \\ &\quad + \sum_{v_j \in \bar{\mathcal{S}}} (|b_j[k+1]| + b_j[k+1]) \\ &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k] + \\ &\quad + \sum_{v_j \in \bar{\mathcal{S}}} (|b_j[k+1]| + b_j[k+1]) \\ &\quad - \sum_{v_j \in \bar{\mathcal{S}}} (b_j[k+1] - b_j[k]),\end{aligned}$$

where we used the fact that $b_j[k] \leq 0$ for $v_j \in \bar{\mathcal{S}}$. Furthermore, we have

$$\begin{aligned}&-\sum_{v_j \in \bar{\mathcal{S}}} (b_j[k+1] - b_j[k]) \\ &= \sum_{v_j \in \mathcal{S}} (b_j[k+1] - b_j[k]) \\ &= \sum_{(v_j, v_i) \in \mathcal{E}_S^-} \Delta f_{ji}[k] - \sum_{(v_l, v_j) \in \mathcal{E}_S^+} \Delta f_{lj}[k],\end{aligned}$$

where the second (third) line follows from Statement 2 (Statement 1) in Proposition 1 and the definition of $\Delta f_{ji}[k]$.

To arrive at (17), we realize that the term $|b_j[k+1]| + b_j[k+1]$ is necessarily zero for all nodes in the set $\bar{\mathcal{S}} - \mathcal{T}$: the reason is that the flows on edges incident to these nodes do not change at all; thus, $b_j[k+1] = b_j[k] \leq 0$, and therefore $|b_j[k+1]| = -b_j[k+1]$.

To complete the proof, we reorder the two summations of the flow changes in the set of edges in \mathcal{E}_S^- (\mathcal{E}_S^+) in terms of outgoing (incoming) edges of nodes in the set \mathcal{T} , which leads us to (18). The fact that $\Delta \varepsilon_j[k] \leq 0$ can be seen as follows:

(i) If $b_j[k+1] \leq 0$, then $|b_j[k+1]| + b_j[k+1] = 0$; thus, since $\Delta f_{ji}[k] \geq 0$ for all $v_j \in \mathcal{T}$ and $v_i \in \mathcal{N}_j^- \cap \mathcal{S}$, and $\Delta f_{lj}[k] \leq 0$ for all $v_j \in \mathcal{T}$ and $v_l \in \mathcal{N}_j^+ \cap \mathcal{S}$, we have $\Delta \varepsilon_j[k] \leq 0$.

(ii) If $b_j[k+1] \geq 0$, then since $b_j[k+1] - b_j[k] = +\sum_{v_i \in \mathcal{N}_j^- \cap \mathcal{S}} \Delta f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+ \cap \mathcal{S}} \Delta f_{lj}[k]$ and $b_j[k] \leq 0$, we have

$$0 \leq b_j[k+1] \leq + \sum_{v_i \in \mathcal{N}_j^- \cap \mathcal{S}} \Delta f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+ \cap \mathcal{S}} \Delta f_{lj}[k].$$

Thus, $|b_j[k+1]| + b_j[k+1] = 2b_j[k+1]$ and

$$2b_j[k+1] \leq + \sum_{v_i \in \mathcal{N}_j^- \cap \mathcal{S}} 2\Delta f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+ \cap \mathcal{S}} 2\Delta f_{lj}[k];$$

thus, $\Delta \varepsilon_j[k] \leq 0$. \square

V. PROOF OF MAIN CONVERGENCE RESULTS

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

Proof of Theorem 2: 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 $t = 0, 1, 2, \dots$, we have (from the first statement of Proposition 2)

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

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

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

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

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

where

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

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

$$\left(\frac{1}{2D_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - g[t] > 0, \quad (20)$$

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

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

which implies that

$$\varepsilon[k+t_0] \leq \varepsilon[k] - \left(\frac{1}{2D_{\max}}\right)^{t_0} \frac{\varepsilon[k]}{2n} \quad (21)$$

$$\leq \left(1 - \frac{1}{2n} \left(\frac{1}{2D_{\max}}\right)^{t_0}\right) \varepsilon[k], \quad (22)$$

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 (20) gets violated for the first time at some $t_0 \in \{0, 1, 2, \dots, n-1\}$, which will establish our proof. Suppose that the inequality (20) holds for all $t \in \{0, 1, 2, \dots, n-1\}$. Then, we argue below that each node v_j in the set V_{k+t} , $t \in \{0, 1, 2, \dots, n-1\}$, has flow balance that satisfies

$$b_j[k+t] \geq \left(\frac{1}{2D_{\max}}\right)^t \frac{\varepsilon[k]}{2n} - g[t] > 0. \quad (23)$$

[This is established at the end of the proof.] Assuming (for now) that (23) holds, we have

$$\sum_{v_j \in V_{k+t}} b_j[k+t] > 0, \forall t \in \{0, 1, \dots, n-1\}. \quad (24)$$

Since, by construction, we have

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

and $|\mathcal{V}| = n$, we need to have $V_{k+t} = V_{k+t-1}$ for some $t \in \{1, 2, \dots, n-1\}$. 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 (24) (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}_S^- and \mathcal{E}_S^+ as in (2) and (3) respectively. Then, from the recursive definition of V_{k+t} we have

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

[Note that both \mathcal{E}_S^+ and \mathcal{E}_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}_S^+ (respectively, \mathcal{E}_S^-), the set V_{k+t} would strictly contain V_{k+t-1} .] Thus, from the first statement of Proposition 1, we have

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

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

We now argue that if the inequality (20) holds for $t \in \{0, 1, \dots, n-1\}$, then inequality (23) also holds for $t \in \{0, 1, \dots, n-1\}$. The proof is by induction. Clearly, the inequality holds for $t = 0$. Suppose that (23) 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 absolute 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 $v_j \in V_{k+t+1}$.

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 $\tilde{f}_{ji}[k+t+1] \geq l_{ji}$;

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 (20) holds for $t \in \{0, 1, \dots, n-1\}$.

(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 (19) (and the fact that $\Delta \varepsilon_l[k] \leq 0$ for $v_l \in \mathcal{T}$) to establish that

$$\Delta \varepsilon_l[k+t] \leq |b_l[k+t+1]| + b_l[k+t+1] - 2 \frac{b_j[k+t]}{2\mathcal{D}_j}, \quad (25)$$

where the second inequality follows from (19) 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 (19) are nonnegative whereas changes in the flows on edges in the second summation in (19) 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.

(a) If $b_l[k+t+1] \leq 0$, then (25) implies $\Delta \varepsilon_l[k+t] \leq -\frac{b_j[k+t]}{\mathcal{D}_j}$ and, since $g[t+1] = g[t] - \Delta \varepsilon_l[k+t]$ (with $g[t] \geq 0$) we, have

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

which violates inequality (20) (i.e., it leads to the contradiction we are trying to establish and we are done).

(b) If $b_l[k+t+1] > 0$, then (25) implies

$$\Delta \varepsilon_l[k+t] \leq 2b_l[k+t+1] - \frac{b_j[k+t]}{\mathcal{D}_j}.$$

Thus, since in (19), all $\Delta \varepsilon_j[k+t] \leq 0$, for $v_j \in \mathcal{T}$, we have

$$\begin{aligned} g[t] - g[t+1] &= \Delta \varepsilon_l[k+t] \leq \Delta \varepsilon_l[k+t] \\ &\leq 2b_l[k+t+1] - \frac{b_j[k+t]}{\mathcal{D}_j}. \end{aligned}$$

Using the induction invariant and the monotonicity of $g[t]$:

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

This completes the proof of Theorem 2. \square

Having completed the proof of Theorem 2, we now proceed to complete the convergence argument by proving Corollary 1.

Proof of Corollary 1: From Theorem 2, we have that $\lim_{k \rightarrow \infty} \varepsilon[k] = \lim_{k \rightarrow \infty} \sum_{j=1}^n |b_j[k]| = 0$, which implies that $\lim_{k \rightarrow \infty} b_j[k] = 0$, $\forall v_j \in \mathcal{V}$. From the flow updates in (10) and (11), the flow $f_{ji}[k]$ on each edge $(v_j, v_i) \in \mathcal{E}$ stabilizes to a value f_{ji}^* , i.e., $f_{ji}^* = \lim_{k \rightarrow \infty} f_{ji}[k]$ exist

for all edges $(v_j, v_i) \in \mathcal{E}$. Clearly, the algorithm described by (5)–(11) results in flows f_{ji}^* that are within the lower and upper bounds on each edge (i.e., $l_{ji} \leq f_{ji}^* \leq u_{ji}$). Furthermore, since $\lim_{k \rightarrow \infty} b_j[k] = 0$, we easily obtain that $\sum_{v_i \in \mathcal{N}_j^-} f_{ji}^* = \sum_{v_i \in \mathcal{N}_j^+} f_{ji}^*$. \square

VI. DISTRIBUTED CHECKING OF CIRCULATION CONDITIONS

Algorithm 1 presented in the previous section allows the nodes to reach a set of flows that are feasible and balanced if the circulation conditions in Theorem 1 are satisfied. In case these conditions are *not* satisfied, we would like the nodes to have a way to determine, in a distributed manner, that this is the case. In particular, we would like a way for nodes that are balanced to determine that there is an imbalance in other parts of the topology.³ In this section we develop a distributed algorithm (referred to as Algorithm 2) that builds on Algorithm 1 and allows the nodes to do exactly that.

We start by making the following observation. When the circulation conditions in Theorem 1 are *not* satisfied, Algorithm 1 converges to a set of flows that lie in the set of allowable flows as determined by the interval constraints on each edge flow, but are not necessarily balanced. The fact that the algorithm indeed converges to a set of flows is not as obvious, but can be derived from the analysis of Algorithm 1 in the previous section. To see this, suppose (for now and for simplicity) that the circulation conditions in Theorem 1 are violated for a single set of nodes \mathcal{S} , $\mathcal{S} \subset \mathcal{V}$ (i.e., the conditions are not violated for any other subset of nodes). From the proof of convergence in the previous section, we see that Algorithm 1 will drive the flows on the edges that are incoming to \mathcal{S} or outgoing from \mathcal{S} as follows:

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

where $\mathcal{E}_\mathcal{S}^-$ and $\mathcal{E}_\mathcal{S}^+$ are given by (2) and (3) respectively. Nodes in the set \mathcal{S} (the absolute balance of which will necessarily be positive) will eventually all reach positive balance and stabilize to a set of flows (and corresponding balances) that remain invariant. [We do not explicitly show this due to space limitations, but it can be shown based on the facts that (i) the flows on edges $\mathcal{E}_\mathcal{S}^-$ ($\mathcal{E}_\mathcal{S}^+$) cannot be decreased (increased), and (ii) there are no further violations of the circulation conditions within \mathcal{S} .] Moreover, nodes in the set $\mathcal{V} - \mathcal{S}$ (the absolute balance of which will necessarily be negative) will all reach negative or zero balance and stop updating the flows. The situation is slightly more complicated if there are multiple/intersecting sets for which the circulation conditions are violated, but the important fact for the analysis in this section is that the flows in Algorithm 1 converge even when the conditions in Theorem 1 are violated (this is also seen in the next section when we present simulations for various types of graphs, including graphs in which the circulation conditions are violated—see Fig. 4 which shows how flows are updated

³Even if there is no violation of the circulation conditions, the establishment of balanced flows via Algorithm 1 would be asymptotic.

on a graph of 20 nodes in which the circulation conditions are violated).

Clearly, once the nodes converge to a set of flows f_{ji}^* , $\forall (v_j, v_i) \in \mathcal{E}$, a simple way to determine that the resulting solution does not correspond to a set of balanced flows is to run an average consensus on the absolute values of the resulting balances b_j^* , $\forall v_j \in \mathcal{V}$. The graph is balanced (i.e., $b_j = 0$, $\forall v_j \in \mathcal{V}$) if and only if the average of the absolute balances satisfies

$$\frac{\sum_{v_i \in \mathcal{V}} |b_i^*|}{n} = 0.$$

If each node v_j was aware of its value b_j^* , since the communication graph \mathcal{G}_u is undirected, one simple way to obtain the above average is to run a linear iterative scheme where each node maintains a variable $x_j[k]$ (initialized at $x_j[0] = |b_j^*|$) and, at each iteration k , each node updates its value as a weighted sum of its own value and the values of its neighbors (see, e.g., [8], [24]). In particular,

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{i \in \mathcal{N}_j} p_{ji}x_i[k], \quad (26)$$

where $\mathcal{N}_j = \mathcal{N}_j^+ \cup \mathcal{N}_j^-$ is the set of neighbors of node v_j and p_{ji} are a set of (fixed) weights, chosen so that the weight matrix $P = [p_{ji}]$ (with weights p_{ji} satisfying $p_{ji} = 0$ if $i \notin \mathcal{N}_j \cup \{j\}$) is doubly stochastic with a simple eigenvalue at 1 and all other eigenvalues having magnitude smaller than 1. In such case, it can be shown that iteration (26) asymptotically reaches average consensus [15], i.e.,

$$\lim_{k \rightarrow \infty} x_j[k] = \frac{\sum_{v_l \in \mathcal{V}} x_l[0]}{n}, \forall v_j \in \mathcal{V}.$$

Given an undirected graph $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$, there are many different ways of having the nodes distributively assign weights p_{ji} such that the resulting matrix $P = [p_{ji}]$ is a primitive doubly stochastic matrix. For instance, assuming the nodes know the total number of nodes n or an upper bound $n' \geq n$, each node v_j can choose

$$p_{ji} = \begin{cases} \frac{1}{n'}, & \text{if } (v_j, v_i) \in \mathcal{E}_u, \\ 0, & \text{if } (v_j, v_i) \notin \mathcal{E}_u, \\ 1 - \frac{|\mathcal{N}_j|}{n'}, & \text{if } v_j = v_i, \end{cases}$$

In practice, the value of b_j^* only becomes available to node v_j asymptotically. Thus, we implement a running average consensus algorithm where we also update the balance of node v_j . In particular, in parallel to Algorithm 1 (which makes available to each node v_j the value $b_j[k+1]$) we run the iteration:

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{i \in \mathcal{N}_j} p_{ji}x_i[k] + \underbrace{|b_j[k+1]| - |b_j[k]|}_{\Delta b_{\text{abs}}[k]}, \quad (27)$$

with initial conditions $x_j[0] = 0$ and $\Delta b_{\text{abs}}[0] = 0$. The above iteration resembles iteration (26) and ensures that $\sum_{v_l \in \mathcal{V}} x_l[k] = \sum_{v_l \in \mathcal{V}} |b_l[k]|$ for all k (because the matrix $P = [p_{ji}]$ is column stochastic). Eventually, for large k , since the flows (and thus the balances) converge, $\Delta b_j[k]$ converges to zero and the above iteration becomes equivalent to iteration

Algorithm 2: Distributed algorithm enhanced for detection of violation of circulation conditions

Each node $v_j \in \mathcal{V}$ **separately does the following:**

Input: $l_{ji}, u_{ji}, \forall v_i \in \mathcal{N}_j^-$

Input: $l_{lj}, u_{lj}, \forall v_l \in \mathcal{N}_j^+$

Input: n' upper bound on number of nodes ($n' \geq n$)

Output: $f_{ji}, \forall v_i \in \mathcal{N}_j^-$

Output: $f_{lj}, \forall v_l \in \mathcal{N}_j^+$

Output: Maintain $x_j[k]$ as indicator for infeasible circulation conditions

begin

Set $f_{ji}[0] = \frac{l_{ji} + u_{ji}}{2}, \forall v_i \in \mathcal{N}_j^-$

Set $f_{lj}[0] = \frac{l_{lj} + u_{lj}}{2}, \forall v_l \in \mathcal{N}_j^+$

Set $\mathcal{D}_j = \mathcal{D}_j^- + \mathcal{D}_j^+$

Set $p_{ji} = \frac{1}{n'}$ for $j \in \mathcal{N}_j, p_{jj} = 1 - \frac{|\mathcal{N}_j|}{n'}$

Set $x_j[0] = 0, \hat{b}_j[0] = 0, \text{ and } x_i[0] = 0, \forall v_i \in \mathcal{N}_j$

foreach iteration, $k = 0, 1, \dots$, do

Calculate:

$$b_j[k] = \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k] - \sum_{v_l \in \mathcal{N}_j^+} f_{lj}[k]$$

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{i \in \mathcal{N}_j} p_{ji}x_i[k] + |b_j[k]| - |\hat{b}_j[k]|$$

Set: $\hat{b}_j[k] = b_j[k]$

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

Transmit: $\frac{\tilde{b}_j[k]}{\mathcal{D}_j}$ and $x_j[k+1]$ to $v_i \in \mathcal{N}_j^-$ and $v_l \in \mathcal{N}_j^+$

Receive: $\frac{\tilde{b}_i[k]}{\mathcal{D}_i}$ from all $v_i \in \mathcal{N}_j^-$ and

$$\frac{\tilde{b}_l[k]}{\mathcal{D}_l} \text{ from all } l \in \mathcal{N}_j^+$$

$x_i[k+1]$ from all $i \in \mathcal{N}_j$

Calculate:

$$\tilde{f}_{ji}[k+1] = f_{ji}[k] + \frac{1}{2} \left(\frac{\tilde{b}_i[k]}{\mathcal{D}_i} - \frac{\tilde{b}_j[k]}{\mathcal{D}_j} \right), v_i \in \mathcal{N}_j^-$$

$$\tilde{f}_{lj}[k+1] = f_{lj}[k] + \frac{1}{2} \left(\frac{\tilde{b}_j[k]}{\mathcal{D}_j} - \frac{\tilde{b}_l[k]}{\mathcal{D}_l} \right), v_l \in \mathcal{N}_j^+$$

Set:

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

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

(26). Thus, the nodes reach consensus on the average of the absolute values of their eventual balances, i.e.,

$$\lim_{k \rightarrow \infty} x_j[k] = \frac{\sum_{v_l \in \mathcal{V}} |b_l^*|}{n}, \forall v_j \in \mathcal{V}.$$

This average is greater than zero if and only if the circulation conditions are violated, providing a distributed way for the nodes to determine the violation. The algorithm pseudocode is provided in Algorithm 2.

Example 2: In this example, we revisit the digraph in Example 1 where we modify (decrease) the upper bounds on two edges, so as to create a violation of the circulation conditions. In particular, we take $u_{1,7} = 2$ and $u_{2,7} = 4$, and execute Algorithm 2. On the left of Fig. 3, we plot the values of the flows $f_{ji}[k]$ for each $(v_j, v_i) \in \mathcal{E}$. It can be observed that the flows eventually stabilize to fixed values as argued earlier in this section. In the middle of Fig. 3, we plot the evolution of the absolute (total) imbalance $\varepsilon[k]$ against the iteration k . Notice that $\varepsilon[k]$ is monotonically non-increasing, but does not go to zero. On the right of Fig. 3, we plot the values $x_j[k]$ in Algorithm 2 for each node $v_j \in \mathcal{V}$. These values eventually converge to the positive value 2.2857 which is the average of the eventual absolute balance (which is 16 in this case). Note that the running averages of the absolute balances do not evolve monotonically. \square

VII. SIMULATION RESULTS

In this section, we present simulation results for Algorithm 1. Specifically, we first present numerical results for a random graph of size $n = 20$ illustrating the behavior of Algorithm 1 for two different cases: (i) the case when the circulation conditions in Theorem 1 do *not* hold, thus, a set of flows that balance the digraph cannot be obtained; (ii) the case when the circulation conditions in Theorem 1 hold and a set of flows that balance the graph can be obtained. We also present numerical results for graphs of various sizes ($n = 20, 50$, and 100) when the circulation conditions in Theorem 1 hold and a set of flows that balance the graph can be obtained. All graphs are randomly generated by creating, independently for each ordered pair (v_j, v_i) of two nodes v_j and v_i ($v_j \neq v_i$), a directed edge from node v_i to node v_j with probability p ($0 < p < 1$) while the flows are initialized at the middle of the feasible interval i.e., $f_{ji}[0] = (l_{ji} + u_{ji})/2$.

Fig. 4 shows what happens in the case of a randomly created graph of 20 nodes, in which the circulation conditions in Theorem 1 do not hold. In the first case, we plot the absolute (total) imbalance $\varepsilon[k] = \sum_{j=1}^n |b_j[k]|$, and in the second case, the nodes balances $b_j[k]$ as a function of the number of iterations k for the distributed algorithm. The plots suggest that the proposed distributed algorithm (Algorithm 1) is unable to obtain a set of flows that balance the corresponding digraph. Nevertheless, as argued in Section VI, the flows converge to a feasible set of values (yet not balanced).

Fig. 5 shows the same case as Fig. 4, with the difference that the circulation conditions in Theorem 1 hold. Here, the

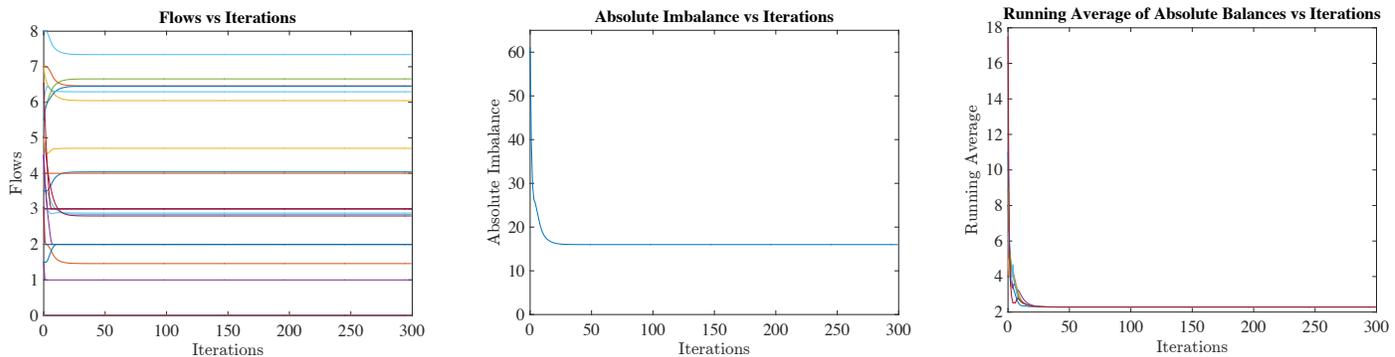


Fig. 3. Shown as a function of the iteration k for the digraph used in Example 2: Left: flow values $f_{ji}[k]$, for $(v_j, v_i) \in \mathcal{E}$; Middle: absolute (total) imbalance $\varepsilon[k]$; Right: running averages $x_j[k]$ for $v_j \in \mathcal{V}$.

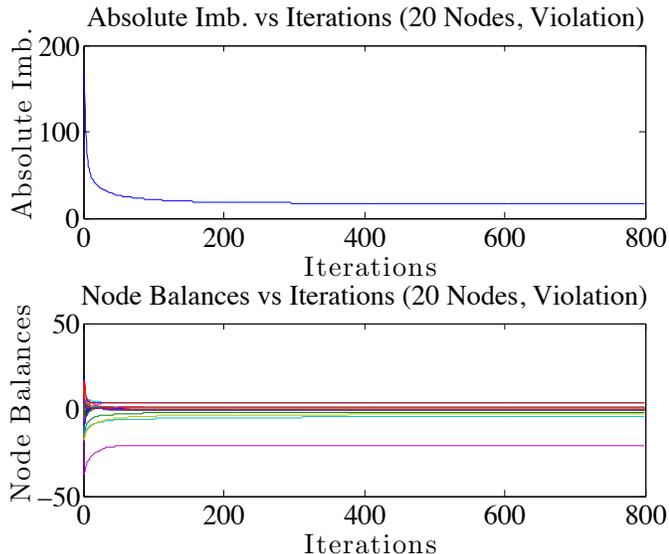


Fig. 4. Execution of Algorithm 1 for the case when the circulation conditions in Theorem 1 do not hold for a random graph of 20 nodes. Top: Absolute (total) imbalance $\varepsilon[k]$ plotted against the number of iterations. Bottom: Node flow balances $b_j[k]$ plotted against the number of iterations.

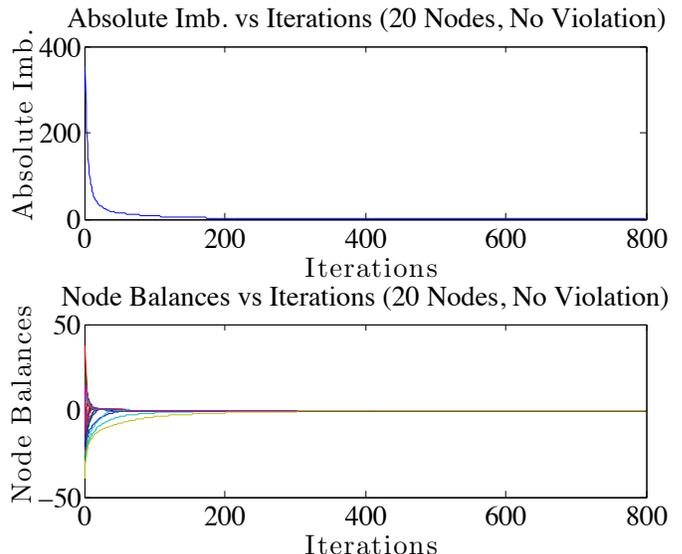


Fig. 5. Execution of Algorithm 1 in the case where the circulation conditions in Theorem 1 hold for a random graph of 20 nodes. Top: Absolute (total) imbalance $\varepsilon[k]$ plotted against the number of iterations. Bottom: Node flow balances $b_j[k]$ plotted against the number of iterations.

plots suggest that Algorithm 1 is able to obtain a set of flows that balance the corresponding digraph.

Fig. 6 shows what happens in the case of 100 graphs of 20 (left), 50 (middle), and 100 (right) nodes each when the circulation conditions in Theorem 1 hold. We plot the average absolute (total) imbalance $\varepsilon[k] = \sum_{j=1}^n |b_j[k]|$ as a function of the number of iterations k for Algorithm 1. The plot suggests that Algorithm 1 is able to obtain a set of flows that balance the corresponding graph.

Finally, we include simulation results for a larger graph of 200 nodes (randomly generated with connection probability $p = 0.25$). The absolute (total) imbalance and the node balances are shown in Figs. 7 and 8, respectively, as a function of the number of iterations.

VIII. CONCLUDING REMARKS

In this paper, we introduced and analyzed a distributed algorithm for assigning flows to the edges of a commodity flow network, described by a digraph, so as to balance the in- and out-flows on each node, while satisfying capacity limits on the edges. In addition, we provided an enhancement to such algorithm that allows the nodes to determine whether a feasible and balanced flow assignment does not exist.

Note that the focus in this paper was on obtaining a feasible and balanced set of flows when the circulation conditions are satisfied. It would be interesting to determine what kind of flows are obtained by the proposed algorithm, i.e., whether they are optimal under a particular criterion, and how the algorithm can be modified to obtain, in a distributed manner, feasible and balanced flows that are optimal under other criteria of interest, including cases where flows on different edges are not weighted equally.

In the future, we also plan to investigate ways of relaxing the

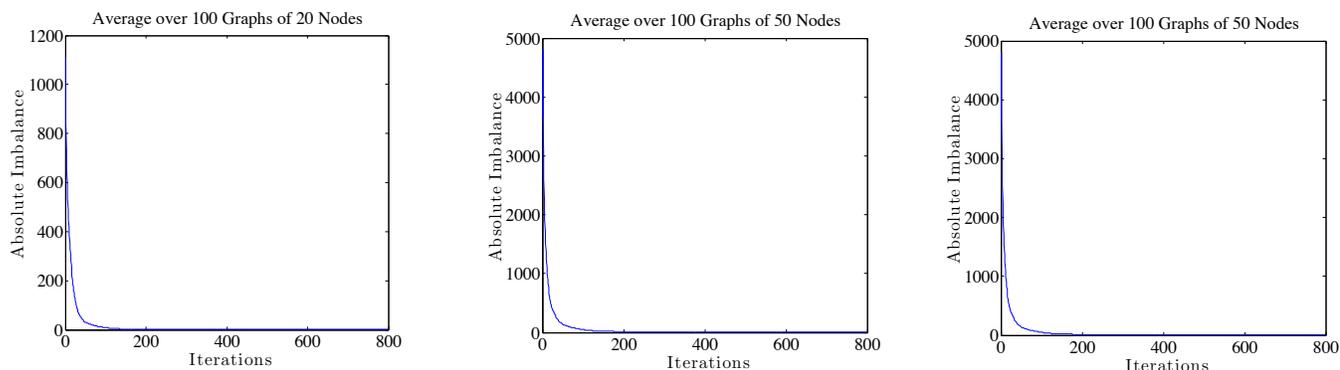


Fig. 6. Absolute (total) imbalance $\varepsilon[k]$ plotted against the number of iterations for Algorithm 1, averaged over 100 graphs of 20 nodes (left), 50 nodes (middle), and 100 nodes (right), in the case where the circulation conditions in Theorem 1 hold.

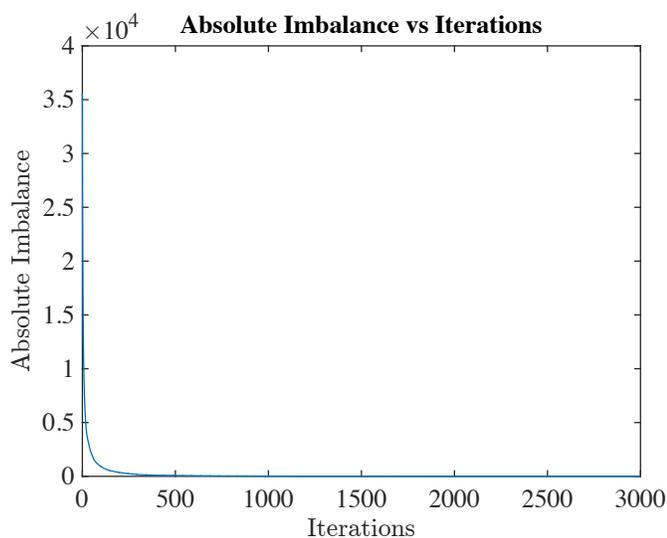


Fig. 7. Absolute (total) imbalance $\varepsilon[k]$ plotted against the number of iterations for Algorithm 1, for a randomly generated graph of 200 nodes.

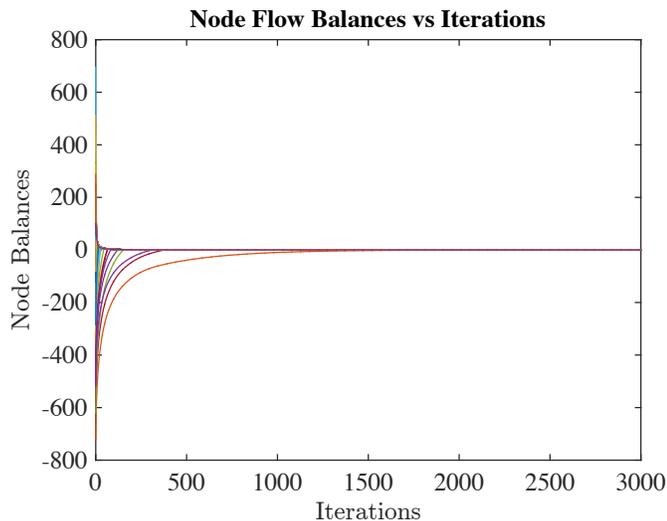


Fig. 8. Node balances plotted against the number of iterations for Algorithm 1, for a randomly generated graph of 200 nodes.

assumption regarding bidirectional communication between neighboring nodes. Since in many applications the communication topology does not necessarily match the physical one, we plan to investigate ways to enhance the algorithm proposed here to allow for different communication topologies. We also plan to study asynchronous and/or event-driven operation of Algorithm 1 in an effort to minimize communication (transmission) costs (e.g., nodes that do not desire any flow changes on a particular link do not have to communicate with the corresponding in-/out-neighbor). Another interesting question is whether Algorithm 2 can be modified, perhaps taking into account existing work in distributed stopping algorithms [25], [26], and Theorem 2 (which guarantees a decrease in the total absolute imbalance after a certain number of steps), to guarantee the detection of violation on the circulation conditions (in Theorem 1) in finite time.

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] A. D. Domínguez-García, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," in *Proc. of IEEE Conference on Decision and Control*, 2012, pp. 3688–3693.
- [6] S. T. Cady, A. D. Domínguez-García, and C. N. Hadjicostis, "A distributed generation control architecture for islanded ac microgrids," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1717–1735, 2015.
- [7] S. T. Cady, M. Zholbarysov, A. D. Domínguez-García, and C. N. Hadjicostis, "A distributed frequency regulation architecture for islanded inertialess ac microgrids," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 1961–1977, November 2017.
- [8] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.
- [9] 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.
- [10] 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.

- [11] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, Massachusetts: Athena Scientific, 1998.
- [12] 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.
- [13] B. Ghahesifard 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.
- [14] N. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1996.
- [15] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, September 2004.
- [16] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, May 2008.
- [17] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [18] A. D. Domínguez-García and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *Proc. of First IEEE International Conference on Smart Grid Communications*, 2010, pp. 537–542.
- [19] C. N. Hadjicostis and A. D. Domínguez-García, "Distributed balancing in digraphs under interval constraints," in *Proc. of IEEE Conference on Decision and Control*, 2016, pp. 1769–1774.
- [20] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed matrix scaling and application to average consensus in directed graphs," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 667–681, March 2013.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 650–660, 2008.
- [25] N. E. Manitará and C. N. Hadjicostis, "Distributed stopping for average consensus in undirected graphs via event-triggered strategies," *Automatica*, vol. 70, pp. 121–127, 2016.
- [26] —, "Distributed stopping for average consensus in digraphs," *IEEE Transactions on Control of Network Systems*, 2017.



Christoforos N. Hadjicostis (M'99, SM'05) received the S.B. degrees in electrical engineering, computer science and engineering, and in mathematics, the M.Eng. degree in electrical engineering and computer science in 1995, and the Ph.D. degree in electrical engineering and computer science in 1999, all from Massachusetts Institute of Technology, Cambridge. In 1999, he joined the Faculty at the University of Illinois at Urbana-Champaign, where he served as Assistant and then Associate Professor with the Department of Electrical and Computer

Engineering, the Coordinated Science Laboratory, and the Information Trust Institute. Since 2007, he has been with the Department of Electrical and Computer Engineering, University of Cyprus, where he is currently Professor. His research focuses on fault diagnosis and tolerance in distributed dynamic systems, error control coding, monitoring, diagnosis and control of large-scale discrete-event systems, and applications to network security, anomaly detection, energy distribution systems, medical diagnosis, biosequencing, and genetic regulatory models. He currently serves as Departmental Editor of the *Journal of Discrete Event Dynamic Systems*, and as Associate Editor of *Automatica* and *IEEE Transactions on Automation Science and Engineering*. He has also served as Associate Editor of *IEEE Transactions on Automatic Control*, *IEEE Transactions on Control Systems Technology*, and *IEEE Transactions on Circuits and Systems I*.



Alejandro D. Domínguez-García (S'02, M'07) received the degree of Electrical Engineer from the University of Oviedo (Spain) in 2001 and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 2007.

He is an Associate Professor with the Department of Electrical and Computer Engineering (ECE), and a Research Associate Professor with the Coordinated Science Laboratory and the Information Trust Institute, all at the University of Illinois at Urbana-Champaign. He is affiliated with the ECE Power and Energy Systems area, and has been a Grainger Associate since August 2011.

His research interests are in the areas of system reliability theory and control, and their applications to electric power systems, power electronics, and embedded electronic systems for safety-critical/fault-tolerant aircraft, aerospace, and automotive applications.

Dr. Domínguez-García received the NSF CAREER Award in 2010, and the Young Engineer Award from the IEEE Power and Energy Society in 2012. In 2014, he was invited by the National Academy of Engineering to attend the US Frontiers of Engineering Symposium, and was selected by the University of Illinois at Urbana-Champaign Provost to receive a Distinguished Promotion Award. In 2015, he received the U of I College of Engineering Dean's Award for Excellence in Research.

He also served as an editor of the *IEEE Transactions on Power Systems* and *IEEE Power Engineering Letters* from 2011 to 2017.