Systems & Control Letters 154 (2021) 104980

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

# On integer balancing of directed graphs

Mohamed-Ali Belabbas [a,1], Xudong Chen [b,*,1]

[a] Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, IL, United States of America

[b] Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, CO, United States of America

ARTICLE INFO

*Article history:*
Received 9 February 2021
Received in revised form 20 May 2021
Accepted 28 May 2021
Available online 10 July 2021

*Keywords:*
Weighted digraphs
Graph balancing

ABSTRACT

A weighted digraph is balanced if the sums of the weights of the incoming and of the outgoing edges are equal at each vertex. We show that if these sums are integers, then for any edge weights satisfying the balance conditions, there exist integer weights obtained by rounding the original weights up or down that preserve both the balance condition and the sum of all edge weights.



## 1. Introduction

Let $G = (V, E)$ be a strongly connected digraph on $n$ vertices, possibly with self-loops but no multi-edges. We use $v_i v_j$ to denote an edge from $v_i \in V$ to $v_j \in V$. For a vertex $v_i$, let $N^-(v_i) := \{v_j \in V \mid v_i v_j \in E\}$ and $N^+(v_i) := \{v_k \in V \mid v_k v_i \in E\}$ be the sets of out-neighbors and in-neighbors of $v_i$, respectively.

We assign $w_{ij} \in \mathbb{R}$ to edges $v_i v_j$, for $v_i v_j \in E$, and denote by $\boldsymbol{w} \in \mathbb{R}^{|E|}$ the collection of these $w_{ij}$. We call $(G, \boldsymbol{w})$ a weighted digraph.

**Definition 1.1.** The weighted digraph $(G, \boldsymbol{w})$ is *balanced* if, for every vertex, the inflow is equal to the outflow:

$$u_i := \sum_{v_j \in N^-(v_i)} w_{ij} = \sum_{v_k \in N^+(v_i)} w_{ki}, \quad \forall v_i \in V. \tag{1.1}$$

We call $u_i$ the *weight* of vertex $v_i$ associated with $(G, \boldsymbol{w})$.

The vector $\boldsymbol{u} := (u_1, \ldots, u_n) \in \mathbb{R}^n$ for $G$ is said to be *feasible* if there exists a $\boldsymbol{w} \in \mathbb{R}^{|E|}$ such that (1.1) holds.

Balanced digraphs have a host of applications in engineering and applied sciences, including the study of flocking behaviors [1], sensor networks and distributed estimation [2]. While balancing over the real numbers is acceptable in some scenarios, others such as traffic management and fractional packing, require integer balancing [3–7].

In this note, we always assume that $\boldsymbol{u}$ is integer-valued. If all the $w_{ij}$ are integers, then clearly every $u_i$ is an integer. The question we are interested in is: Given a feasible integer-valued $\boldsymbol{u}$ and given a $\boldsymbol{w}$ that satisfies (1.1), can we find an integer-valued $\boldsymbol{w}^*$, as close to $\boldsymbol{w}$ as possible, satisfying (1.1)? We show that the answer is affirmative in a sense that every edge weight $w^*_{ij}$ can be obtained by taking either a ceil or a floor function of $w_{ij}$:

**Theorem 1.2.** *Let $G$ be a strongly connected digraph, $\boldsymbol{u} \in \mathbb{Z}^n$ be any feasible vector, and $\boldsymbol{w} \in \mathbb{R}^{|E|}$ be any solution to (1.1). Then, there exists a $\boldsymbol{w}^* \in \mathbb{Z}^{|E|}$ such that (1.1) holds (with the same $\boldsymbol{u}$) and $\|\boldsymbol{w} - \boldsymbol{w}^*\|_\infty < 1$.*

**Remark 1.3.** A corollary of Theorem 1.2 is that if $\boldsymbol{w}$ is a non-negative solution to (1.1), then so is $\boldsymbol{w}^*$.

The result also applies to the case of weakly connected digraph $G$, based upon the fact that $(G, \boldsymbol{w})$ is balanced if and only if every strongly connected component of $(G, \boldsymbol{w})$ is balanced [6].

We provide below a constructive proof of Theorem 1.2.

## 2. Algorithm and proofs

To proceed, we associate to the digraph $G = (V, E)$ on $n$ vertices an undirected bipartite graph $B = (X \sqcup Y, F)$ on $2n$ vertices, where $X \sqcup Y$ is the vertex set and $F$ is the edge set. Each of the two sets $X$ and $Y$ comprises $n$ vertices. The edge set $F$ is defined as follows: there is an edge $(x_i, y_j)$ in $B$ if $v_i v_j$ is an edge of $G$. See Fig. 1 for an illustration.

Note that the directed edges in $G$ are in one-to-one correspondence with the undirected edges in $B$. Thus, we can assign the edge weights $w_{ij}$, for $v_i v_j \in E$, to the edges $(x_i, y_j)$ in $B$.

The balance relation (1.1), when applied to the bipartite representation of $G$, is now turned into

$$u_i = \sum_{y_j \in N(x_i)} w_{ij} = \sum_{x_k \in N(y_i)} w_{ki}, \quad \forall i = 1, \ldots, n. \tag{2.1}$$

* Corresponding author.
*E-mail addresses:* belabbas@illinois.edu (M.-A. Belabbas), xudong.chen@colorado.edu (X. Chen).

[1] The two authors M.-A. Belabbas and X. Chen contributed equally to the manuscript (in all categories).

https://doi.org/10.1016/j.sysconle.2021.104980
0167-6911/

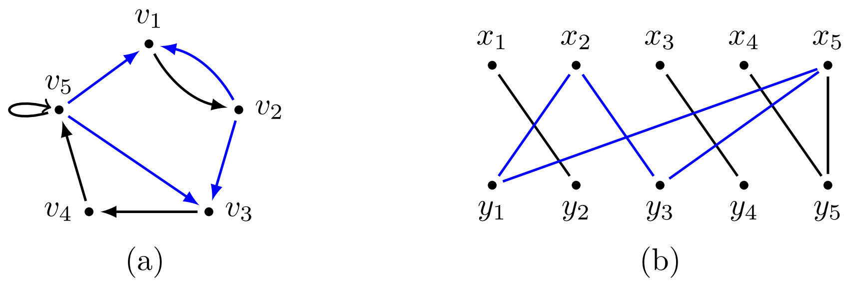


**Fig. 1.** *Left*: A digraph $G$. *Right*: Its bipartite counterpart $B$. A cycle in $B$, and the corresponding edges in $G$, are marked in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

If the above relations hold for some real numbers $u_i$, then $(B, \boldsymbol{w})$ is said to be balanced with vertex weights $u_i$ for both $x_i$ and $y_i$. The following result is an immediate consequence of the above construction of $(B, \boldsymbol{w})$:

**Lemma 2.1.** *The digraph* $(G, \boldsymbol{w})$ *is balanced if and only if* $(B, \boldsymbol{w})$ *is balanced.*

Now, let $\boldsymbol{u} \in \mathbb{Z}^n$ be a feasible vector for $G$ and $\boldsymbol{w} \in \mathbb{R}^{|E|}$ be such that (2.1) is satisfied. We refer to elements of $\mathbb{R}\backslash\mathbb{Z}$ as *decimal* numbers.

Every cycle in $B$ has an even number of edges, and the number is at least 4. A cycle in $B$ does not correspond to a (directed) cycle in $G$, as illustrated in Fig. 1. Instead, if $x_{\alpha_1}y_{\beta_1}\cdots x_{\alpha_p}y_{\beta_p}x_{\alpha_1}$ is a cycle in $B$, then each vertex $v_{\alpha_i}$ in $G$ has two *outgoing* edges $v_{\alpha_i}v_{\beta_i}$ and $v_{\alpha_i}v_{\beta_{i-1}}$ (with $\beta_0$ identified with $\beta_p$) while each vertex $v_{\beta_i}$ has two *incoming* edges $v_{\alpha_i}v_{\beta_i}$ and $v_{\alpha_{i+1}}v_{\beta_i}$ (with $\alpha_{p+1}$ identified with $\alpha_1$).

Given a balanced $(B, \boldsymbol{w})$, our aim is to obtain a set of integer edge weights $w^*_{ij} \in \mathbb{Z}$ that satisfy (1.1). Algorithm 1 does so in a finite number of steps. For that, we need the following definition:

**Definition 2.2.** An edge in $(B, \boldsymbol{w})$ is *decimal* if its weight is a decimal number. A cycle $C$ in $(B, \boldsymbol{w})$ is *completely decimal* if all its edges are decimal.

The following lemma says that it is computationally simple both to decide whether a completely decimal cycle exists and to exhibit one.

**Lemma 2.3.** *Let* $(B, \boldsymbol{w})$ *be balanced. If there is a decimal edge in* $(B, \boldsymbol{w})$*, then there is a completely decimal cycle and it can be found in* $B$ *in at most* $2n$ *steps.*

**Proof.** Let $x_i$ be a vertex incident to a decimal edge. Denote by $\lambda(x_i)$ the number of decimal edges incident to $x_i$. Since the vertex weight $u_i$ of $x_i$ is integer-valued, $\lambda(x_i) \geq 2$. Now, fix a decimal edge $(x_i, y_j)$ in $(B, \boldsymbol{w})$. By the same arguments, $\lambda(y_j) \geq 2$. Thus, there exists another decimal edge incident to $y_j$, say $(y_j, x_k)$ and, similarly, $\lambda(x_k) \geq 2$. Iterating this procedure, we will return to some previously encountered vertex $x_\ell$ (since $B$ is finite) in at most $2n$ steps. By construction, the vertices obtained in the process yield a completely decimal cycle. □

We are now in a position to present the algorithm:

**Algorithm 1.** Start with a balanced $(B, \boldsymbol{w}(0))$ with integer vertex weights, and denote by $\boldsymbol{w}(k)$ the vector of edge weights at iteration $k$. While there exists a completely decimal cycle in $(B, \boldsymbol{w}(k))$, perform the following steps sequentially:

- select a completely decimal cycle $C$ in $(B, \boldsymbol{w}(k))$;
  select an edge $e$ in $C$ whose weight is closest to an integer;
- write $C = x_{\alpha_1}y_{\beta_1}\cdots x_{\alpha_p}y_{\beta_p}x_{\alpha_1}$, with $e = (x_{\alpha_1}, y_{\beta_1})$ and $p = |C|/2$;
  set $\epsilon_\ell := w_{\alpha_1\beta_1} - \lfloor w_{\alpha_1\beta_1}\rfloor$, $\epsilon_h := \lceil w_{\alpha_1\beta_1}\rceil - w_{\alpha_1\beta_1}$;
- *if* $\epsilon_\ell \leq \epsilon_h$
$$\begin{aligned} w_{\alpha_i\beta_i}(k+1) &:= w_{\alpha_i\beta_i}(k) - \epsilon_\ell \\ w_{\beta_i\alpha_{i+1}}(k+1) &:= w_{\beta_i\alpha_{i+1}}(k) + \epsilon_\ell \end{aligned} \quad \text{for } 1 \leq i \leq p; \tag{2.2}$$
  *else*
$$\begin{aligned} w_{\alpha_i\beta_i}(k+1) &:= w_{\alpha_i\beta_i}(k) + \epsilon_h \\ w_{\beta_i\alpha_{i+1}}(k+1) &:= w_{\beta_i\alpha_{i+1}}(k) - \epsilon_h \end{aligned} \quad \text{for } 1 \leq i \leq p; \tag{2.3}$$
  where $\alpha_{p+1}$ is identified with $\alpha_1$;
- for any edge $(x_i, y_j) \in F$ not in $C$, set $w_{ij}(k+1) := w_{ij}(k)$;
- increase the value of $k$ by 1.

When there are no decimal edges left, return $\boldsymbol{w}^* := \boldsymbol{w}(k)$.

Theorem 1.2 is then a direct consequence of the following result:

**Theorem 2.4.** *Given a balanced bipartite graph* $(B, \boldsymbol{w}(0))$ *with vertex weights* $\boldsymbol{u} \in \mathbb{Z}^n$*, Algorithm 1 terminates in a finite number of steps and returns a balanced* $(B, \boldsymbol{w}^*)$ *with* $w^* \in \mathbb{Z}^{|E|}$ *and the same vertex weights* $\boldsymbol{u}$*. Moreover,* $\|\boldsymbol{w}^* - \boldsymbol{w}\|_\infty < 1$.

We establish Theorem 2.4 and start with the following proposition:

**Proposition 2.5.** *Let* $(B, \boldsymbol{w})$ *be balanced with vertex weights* $\boldsymbol{u} \in \mathbb{Z}^{|E|}$*. Denote by* $(B, \boldsymbol{w}')$ *the bipartite graph obtained after one iteration on* $\boldsymbol{w}$ *performed by Algorithm 1. Let* $\boldsymbol{u}'$ *be the vertex weights associated with* $(B, \boldsymbol{w}')$*. Then,* $(B, \boldsymbol{w}')$ *is balanced with* $\boldsymbol{u}' = \boldsymbol{u}$*, and* $\lfloor \boldsymbol{w}_{ij}\rfloor \leq \boldsymbol{w}'_{ij} \leq \lceil \boldsymbol{w}_{ij}\rceil$ *holds for every edge weight.*

**Proof.** Let $C = x_{\alpha_1}y_{\beta_1}\cdots x_{\alpha_p}y_{\beta_p}x_{\alpha_1}$ be a completely decimal cycle as described in the algorithm. For any vertex $x_{\alpha_i}$ in the cycle, we have that
$$\sum_{\gamma \in N(x_{\alpha_i})} (w'_{\alpha_i\gamma} - w_{\alpha_i\gamma}) = (w'_{\alpha_i\beta_{i-1}} + w'_{\alpha_i\beta_i}) - (w_{\alpha_i\beta_{i-1}} + w_{\alpha_i\beta_i}), \tag{2.4}$$
where $\beta_0$ is identified with $\beta_p$ for the case $i = 1$. By (2.2) and (2.3), the two expressions in parentheses on the right hand side of (2.4) are equal, so the difference is 0. It follows that $u'_{\alpha_i} = u_{\alpha_i}$. The same arguments can be applied to vertices $y_{\beta_i}$.

For any vertex $x_i$ (or $y_i$) not in $C$, the weights of edges incident to it are not updated, so $u'_i = u_i$.

Finally, let $\epsilon_\ell$ and $\epsilon_h$ be defined as in Algorithm 1, and let $\epsilon := \min\{\epsilon_\ell, \epsilon_h\}$. Then, $\lfloor w_{ij}\rfloor \leq w_{ij} - \epsilon \leq w'_{ij} \leq w_{ij} + \epsilon \leq \lceil w_{ij}\rceil$, which completes the proof. □

We next have the following result strengthening Lemma 2.3:

**Proposition 2.6.** *Let $(B, \boldsymbol{w})$ be a balanced bipartite graph, with vertex weights $\boldsymbol{u} \in \mathbb{Z}^n$. Then, the following statements are equivalent:*

1. *There is no decimal edge;*
2. *There is no completely decimal cycle;*
3. *The vector $\boldsymbol{w}$ is integer-valued.*

**Proof.** It is immediate that (1) implies (2) and that (3) implies (1). Assuming (2) holds, we prove (3). Let $F' \subseteq F$ be the collection of decimal edges. Suppose that $F' \neq \varnothing$; then, we let $X' \subseteq X$ and $Y' \subseteq Y$ be the collections of vertices incident to the edges in $F'$. It follows that the subgraph $B' = (X' \sqcup Y', F')$ induced by $X' \sqcup Y'$ is acyclic by assumption (2). Its connected components are thus trees. Pick a connected component of $B'$ and one of its leaves, say $x_i$. On the one hand, there exists one and only one edge $(x_i, y_j)$ in $B'$ such that the weight $w_{ij}$ is decimal. By construction, this edge is also the only decimal edge in $B$ incident to $x_i$. On the other hand, since $(B, \boldsymbol{w})$ is balanced, we have that $w_{ij} = u_i - \sum_{y_{j'} \in N(x_i) \setminus \{y_j\}} w_{ij'}$. The right hand side of the expression is integer-valued, which is a contradiction. □

We now present a proof of Theorem 2.4:

**Proof of Theorem 2.4.** Every iteration of Algorithm 1 on a balanced $(B, \boldsymbol{w})$, with integer-valued vertex weights, affects only its decimal edges and reduces the number of its completely decimal cycles by at least one. Thus, Theorem 2.4 follows as an immediate consequence of Propositions 2.5 and 2.6. □

Finally, note that the arguments in the above proof also imply that every decimal edge of such $(B, \boldsymbol{w})$ is contained in a completely decimal cycle. Indeed, if $(x_i, y_j)$ is a decimal edge that is not contained in any completely decimal cycle; then, the weight $w_{ij}$ will not be affected by executing Algorithm 1, which is a contradiction.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported by the National Science Foundation [ECCS-1809076, ECCS-1809315] and the Air Force Office of Scientific Research [FA9550-20-1-0076, FA9550-20-1-0333].

## References

[1] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Automat. Control 48 (6) (2003) 988–1001.
[2] R. Carli, A. Chiuso, L. Schenato, S. Zampieri, Distributed Kalman filtering based on consensus strategies, IEEE J. Sel. Areas Commun. 26 (4) (2008) 622–633.
[3] N. Garg, J. Koenemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, SIAM J. Comput. 37 (2) (2007) 630–652.
[4] S.A. Plotkin, D.B. Shmoys, É. Tardos, Fast approximation algorithms for fractional packing and covering problems, Math. Oper. Res. 20 (2) (1995) 257–301.
[5] D.P. Bertsekas, Network Optimization: Continuous and Discrete Models, Athena Scientific, 1998.
[6] L. Hooi-Tong, On a class of directed graphs—with an application to traffic-flow problems, Oper. Res. 18 (1) (1970) 87–94.
[7] A. Rikos, C. Hadjicostis, Distributed integer weight balancing in the presence of time delays in directed graphs, IEEE Trans. Control Netw. Syst. 5 (3) (2017) 1300–1309.