Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2017-01. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

The chip-firing halting problem for multigraphs and convex cost flows

Bálint Hujter

February 2017

The chip-firing halting problem for multigraphs and convex cost flows

Bálint Hujter*

Abstract

In this paper we address the chip-firing halting problem for undirected multigraphs. We give a polynomial algorithm for deciding the halting problem in the special case when the number of chips in the distribution is equal to the sum of the edge multiplicities of the graph. The key part of our algorithm uses convex cost flow optimization to give an efficient algorithmic proof of a theorem of An, Baker, Kuperberg and Shokrieh; improving a previous algorithm of Backman.

Keywords: chip-firing game; computational complexity

1 Introduction

Chip-firing is a solitary game on graphs defined by Björner, Lovász and Shor [6, 5]. Each node contains a pile of chips. A legal move is to choose a node with at least as many chips as its out-degree and let it send a chip along each outgoing edge.

Björner, Lovász and Shor proved that given an initial chip-distribution, either every chip-firing game can be continued indefinitely, or every game terminates after finitely many steps. Tardos [17] proved that on a simple undirected graph on n nodes, a finite chip-firing game terminates in $O(n^4)$ firings. On the other hand, for digraphs the number of firings in a terminating chip-firing game is not necessarily polynomial [9]. It is not as well-known that if we consider an undirected multigraph given by an adjacency matrix, then the number of firings in a terminating game is pseudopolynomial but not necessarily polynomial in the input size (see Example 2.4). Therefore, it is a nontrivial question whether there exists a polynomial algorithm deciding whether a chip-distribution x is terminating or not. This is called the *chip-firing halting problem.*

In [10], Farrell and Levine proved that the halting problem for (strongly connected) directed multigraphs is **NP**-complete and noted that the question is open for undirected multigraphs and directed simple graphs either.

In this paper, we focus on undirected multigraphs and aim the following conjecture.

*MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös Loránd University, Budapest, Hungary. Supported by the Hungarian Scientic Research Fund - OTKA K109240. Email: `hujterb@cs.elte.hu`

	Undirected	Directed (Strongly Connected)
Simple Graphs	in **P** [17]	in **NP** [10]
Multigraphs	in **NP** ∩ **coNP** [14]	**NP**-complete[10].

Table 1: Overview of the main complexity results concerning the halting problem.

Conjecture 1.1. *The chip-firing halting problem for undirected multigraphs is in* ***P****.*

Several reasons support this conjecture. First, the problem is in **NP**∩**coNP** [14]. Second, the closely related problem of reachability of chip-distributions [6] is in **P** [14]. Third, as the main result of this paper, we prove the following special case of the conjecture:

Theorem 1.2. *There is a polynomial algorithm deciding the halting problem for undirected multigraphs, in the special case of distributions of exactly M chips, where M denotes the sum of the edge multiplicities of the multigraph.*

This algorithm is based on a theorem of An, Baker, Kuperberg and Shokrieh [2, Theorem 4.10.] which states that the equivalence classes of chip-distributions of M chips can be represented by orientations of the (multi)graph. Backman's Algorithm 7.7 of [3] finds such a representing orientation using max-flow-min-cut computations. His algorithm is polynomial in the simple graph case, but the number of MFMC subroutines needed is pseudopolynomial in the multigraph case. The algorithm presented in Section 3 of this paper improves it by compressing the main part of the algorithm into one convex cost flow optimization problem. This way the algorithm remains polynomial in the multigraphs case, as well.

Once the representing orientation found, by considering the *sink-reversal game* started from this orientation, we are able to determine the finiteness of the given chip-firing distribution.

Chip-firing games have a strong connection to the graph divisor theory introduced by Baker and Norine in [4]. Section 4 briefly describes this connection and shows how the halting problem is represented in graph divisor theory.

2 Preliminaries

2.1 Multigraphs

A *multigraph* is an undirected graph $G = (V, E)$ with an edge multiplicity function $m : E \to \mathbb{N}$. As G is undirected, $vw \in E$ iff $wv \in E$ and $m_{vw} = m_{wv}$.

We use the notations $d(v) = \sum_{w:vw\in E} m_{vw}$ and $M = \sum_{e\in E} m(e)$. Note that $|E| = O(|V|^2)$, but M is not necessarily polynomial in $|V|$.

When we give a multigraph as an input to an algorithm, we always encode it by its adjacency matrix. As the adjacency matrix can be encoded in $O(|V|^2 \log M)$ bits.

Hence the size of the input is not increased by the values of the edge multiplicities, just the logarithms of it.

Remark 2.1. We are only considering undirected graphs. Note that some of the following results remains true or have an equivalent version for directed graphs, but the main results of the paper only concern undirected graphs.

The *Laplacian matrix* of a multigraph G is the following matrix $L \in \mathbb{Z}^{V\times V}$:

$$L(v,w) = \begin{cases} -d(v) & \text{if } v = w; \\ m_{vw} & \text{if } v \neq w \text{ and } vw \in E; \\ 0 & \text{if } v \neq w \text{ and } vw \notin E. \end{cases}$$

We define orientations of multigraphs the following way: if we have an edge vw with multiplicity m_{vw}, then we consider it as m_{vw} pieces of single edges, and one may orient each single edge independently from the others. An orientation of G is described by a number $0 \leq \overrightarrow{m}_{vw} \leq m_{vw}$ for each $vw \in E$, meaning that $\overrightarrow{m}_{vw}$ pieces of edges are oriented from v to w and $\overrightarrow{m}_{wv} = m_{vw} - \overrightarrow{m}_{vw}$ pieces are oriented from w to v.

We call an edge vw *consistently oriented* if $\overrightarrow{m}_{wv} \in \{0, m_{vw}\}$, and call an orientation *consistent*, if all the edges are consistently oriented. We use the following notation for the in-degree of a node v: $d^-_{\mathcal{O}}(v) = \sum_{w:wv\in E} \overrightarrow{m}_{wv}$. Node v is called a *sink* if $d^-_{\mathcal{O}}(v) = d(v)$ and a *source* if $d^-_{\mathcal{O}}(v) = 0$.

We introduce notations for some vectors of $\mathbb{Z}^V$. $\mathbf{1}$ stands for the vector with all 1s; $\mathbf{1}_v$ stands for the vector with 1 on a single node v and 0 elsewhere. $\mathbf{d}$ denotes the vector of $d(v)$s (degrees) in G and $\mathbf{d}^-_{\mathcal{O}}$ denotes the vector of $d^-_{\mathcal{O}}(v)$s (indegrees) of the orientation $\mathcal{O}$ of G.

2.2 Chip-firing games

Chip-firing games were first defined by Björner, Lovász and Shor [6] and have a straightforward generalization for multigraphs. We consider a multigraph G with a pile of chips on each of its nodes. A position of the game, called a *chip-distribution* (or just distribution) is described by a vector $x \in \mathbb{Z}^V$, where $x(v)$ is interpreted as the number of chips on node $v \in V$. We denote the set of all chip-distributions on G by $\mathrm{Chip}(G)$.

The basic move of the game is *firing* a node. Firing node v means that for every $vw \in E$, node v passes m_{vw} chips to w. In other words, firing a node v means taking the new chip-distribution $x + L\mathbf{1}_v$ instead of x.

A node $v \in V$ is active with respect to a chip-distribution x if $x(v) \geq d(v)$. The firing of a node $v \in V$ is *legal*, if v was active before the firing (i.e. v has a nonnegative amount of chips after the firing). A *legal game* is a sequence of distributions in which every distribution is obtained from the previous one by a legal firing. A legal game terminates if it arrives at *stable* distribution, which is a chip-distribution without any active nodes. For a legal game, let us call the vector $f \in \mathbb{Z}^V$, where $f(v)$ equals the number of times v has been fired, the *firing vector* of the game.

We use the notation $\deg(x) = \sum_{v\in V} x(v)$ for any $x \in \mathrm{Chip}(G)$. It is easy to check that $\deg(x)$ is invariant during a chip-firing game.

2.2.1 Halting problem

The following theorem of Björner, Lovász and Shor describes a very important "Abelian" property of the chip-firing game.

Theorem 2.2. *[6, Remark 2.4] From a given initial chip-distribution, either every legal game can be continued indefinitely, or every legal game terminates after finitely many steps. The firing vector of every maximal legal game is the same.*

Based on this fact, we call a distribution x *terminating* if a legal game (hence, all legal games) started from x terminates, and we call x *non-terminating* otherwise.

Problem 2.3 (Chip-firing halting problem)**.** Given a graph G and $x \in Chip(G)$, determine whether x is terminating or non-terminating.

In [17] Tardos proved that any terminating chip-firing game terminates within $O(|V|^2 M)$ moves. As a consequence, the trivial algorithm of playing a chip-firing game for at most $O(|V|^2 M)$ decides the halting problem. This bound is polynomial in the case of simple graphs but only pseudopolynomial for multigraphs. We show an example of a chip-firing game which terminates, but not in polynomial time.

Example 2.4. Let the multigraph G be described by the following adjacency matrix A_G with chip-distribution x.

$$A_G = \begin{bmatrix} 0 & 2^n & 0 & 0 \\ 2^n & 0 & 1 & 0 \\ 0 & 1 & 0 & 4^n \\ 0 & 0 & 4^n & 0 \end{bmatrix} \qquad x = \begin{bmatrix} 2^{n+1}-1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

It is easy to check that the only possible firing sequence is obtained by the first and second node firing alternatingly. The game terminates after 2^n such turns.

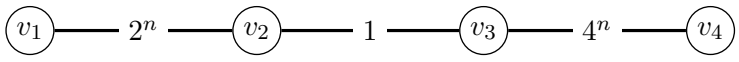

Figure 1: Example of a multigraph with a chip distribution which terminates, but not in polynomial time.

2.2.2 Linear equivalence

The following definition originates in the graph divisor theory of Baker and Norine [4].

Definition 2.5. For $x, y \in \mathrm{Chip}(G)$, we say that x and y are *linearly equivalent* (and denote it by $x \sim y$) if there exists $z \in \mathbb{Z}^V$ such that $x = y + Lz$.

Remark 2.6. It is easy to check that $\sim$ is an equivalence relation on $\mathrm{Chip}(G)$. Note that x and y are linearly equivalent iff y is reachable from x in the unconstrained variant of chip-firing game, in which a node does not need to be active before it fires.

The next lemma appeared in [7, Lemma 4.3.] and [10, Lemma 2.1].

Lemma 2.7. *Let G be a multigraph and $x, y \in \mathrm{Chip}(G)$. If $x \sim y$, then x is terminating if and only if y is terminating.*

To be self-contained, we include the proof from [15].

Proof. By symmetry, it is enough to prove that if x is terminating, then y is also terminating.

Let x be a terminating chip-distribution. We play the chip-firing game starting from x until it terminates. Let the final configuration be x^*. Clearly, $x^* \sim x \sim y$. Let z be an integer vector with $x^* = y + Lz$. We can suppose that z is non-negative (otherwise we can add 1 to each of its coordinates, maintaining $x^* = y + Lz$). Now we start a *bounded* chip-firing game from y defined by the following rule: if there is a node v with at least $d(v)$ chips that has been fired less than $z(v)$ times, then one such node is fired. If there is no such node, the game ends. Clearly, after at most $\sum_{v \in V} z(v)$ firings, the bounded game terminates. We claim that the final distribution $y' = y + Lz'$ (where $z' \leq z$) is stable. Indeed, as the game stopped, for any node v with $y'(v) \geq d(v)$, $z'(v) = z(v)$. As x^* is stable, $x^*(v) < d(v)$, hence from $x^* = y' + L(z - z')$ and $z(v) = z'(v)$, we get $d(v) > x^*(v) \geq y'(v)$, which is a contradiction. □

2.3 Sink-reversal games

Variants of sink-reversal games have appeared in many forms in literature: [6, 8, 13] etc. Here we give a short and self-contained summary of some basic results needed in this paper.

Let G be a multigraph with an orientation $\mathcal{O}$. The *sink-reversal game* on G is defined as follows. The basic move of the game is *reversing a sink*, which means that if a node v is a sink then we may reverse the orientation of all edges incident to v (this way v becomes a source in the newly obtained orientation). A game terminates if there are no sink nodes in the actual orientation.

Let $\mathcal{O}$ be any orientation of G and consider the in-degree vector $\mathbf{d}^-_{\mathcal{O}}$ as a chip-distribution. Notice that we can fire node v if and only if v is a sink in $\mathcal{O}$. After firing v, the resulting chip-distribution is the in-degree vector $\mathbf{d}^-_{\mathcal{O}'}$ of orientation $\mathcal{O}'$ obtained by sink-reversing v in $\mathcal{O}$. These observations imply the following claim.

Claim 2.8. *The chip-firing game started from $\mathbf{d}^-_{\mathcal{O}}$ has the same dynamics as the sink-reversal game started from $\mathcal{O}$: both are terminating or both are non-terminating. For any node v, the number of sink-reversals at v equals the number of firings at v.*

By Theorem 2.2, we get that from a given initial orientation, either every sink-reversal game can be continued indefinitely, or every legal game terminates after finitely many steps. The advantage of sink-reversal games is that we have a nice characterization of terminating games.

Proposition 2.9. *A sink-reversal game started from orientation $\mathcal{O}$ of G is terminating if and only if $\mathcal{O}$ contains a directed cycle.*

The proof of Proposition 2.9 is a consequence of the forthcoming Claim 2.10 and Proposition 2.12.

Claim 2.10. *A* sink-reversal game *started from an acyclic orientation $\mathcal{O}$ of G is non-terminating.*

Proof. When we reverse a sink node v in an acyclic orientation, we cannot get a new directed circle as all reversed edges are incident to v but v becomes a source in the new orientation. Hence the resulting orientation remains acyclic and therefore contains a sink. □

Next we turn our attention to orientations containing directed cycles.

Lemma 2.11. *If there is a directed path from v to a node w contained in a directed cycle then v can never be sink-reversed.*

Proof. As w is contained in a directed cycle, there is a directed walk:

$$v_0 = v \to v_1 \to \ldots \to v_{k-1} \to v_k = w \to v_{k+1} \to v_{\ell-1} \to v_\ell = w$$

None of the nodes of the sequence is originally a sink. v_{i+1} must be sink-reversed before the first sink-reversal of v_i, unless v_i cannot become a sink. But as $v_k = v_\ell$, it means that none of the nodes of the set $\{v_0, v_1, \ldots, v_\ell\}$ can be the first to be sink-reversed. □

Now let $\mathcal{O}$ be an orientation of the multigraph $G = (V, E)$. We define the weighted digraph $G^*_{\mathcal{O}}(V, A^*)$ the following way: if there is at least one edge oriented from v to w in $\mathcal{O}$ then we add an arc $\overrightarrow{vw}$ to A^* with weight 0 (forward arc). If there is at least one edge oriented from v to w in $\mathcal{O}$ then we add an arc $\overrightarrow{wv}$ to A^* with weight 1 (backward arc). Let $S = \{v \in V : v \text{ is contained in a directed cycle of } \mathcal{O}\}$.

Let us define the integer-valued potential function $\pi_{\mathcal{O}} : V \to Z$ the following way: let $\pi_{\mathcal{O}}(v)$ be the weight of the minimal weight dipath from S to v in $G^*_{\mathcal{O}}(V, A^*)$. In particular, $\pi_{\mathcal{O}}(v) = 0$ if and only if $v \in S$.

Proposition 2.12. *Let $\mathcal{O}$ be an orientation of G containing at least one directed cycle. The sink-reversal game started from orientation $\mathcal{O}$ is terminating and each node v is sink-reversed exactly $\pi_{\mathcal{O}}(v)$ times.*

Proof. If v is a sink node in $\mathcal{O}$, then $\pi_{\mathcal{O}}(v) > 0$ as no directed circle can be reached from v in $\mathcal{O}$. It can be easily checked that when the sink node v is reversed, $\pi_{\mathcal{O}}(v)$ decreases by 1 and for any other $w \in V$, $\pi_{\mathcal{O}}(w)$ remains unchanged.

The sink-reversal game cannot halt while there is a node with a positive $\pi_{\mathcal{O}}$ value. Indeed, if there is no sink node in $\mathcal{O}$, then form any node v, we can greedily find a dipath leading to a directed circle; so by Lemma 2.11; if there is any node v with $\pi_{\mathcal{O}}(v) > 0$, then there is a sink. □

Corollary 2.13. *Any terminating sink-reversal game terminates in less than $|V|^2$ sink-reversals.*

Proof. $\pi_{\mathcal{O}}(v) \leq |V| - 1$ for any $v \in V$. □

Note that this result has much in common with [8, Corollary 3.3.].

Proposition 2.14. *There is an algorithm running in $O(|E|)$ time for the following problem: given an orientation $\mathcal{O}$ of multigraph G, decide whether the sink-reversal game $\mathcal{O}$ is terminating or non-terminating.*

Proof. By Proposition 2.9, we only need to check whether $\mathcal{O}$ contains a directed cycle or not, which can be done by a depth first search. If there is any edge which is not consistently oriented (i.e. $0 < \overrightarrow{m}_{vw} < m_{vw}$), then it gives rise to a directed cycle. If G is consistently oriented, it is enough to consider one copy of the consistently oriented parallel edges. Hence the running time is $O(|E|)$. □

Remark 2.15. Another possibility is to simply start a sink-reversal game from $\mathcal{O}$. By Corollary 2.13, if the game does not terminate in $|V|^2$ steps then it is non-terminating.

2.4 Convex cost flows

A detailed description of convex cost flow problems and solution techniques can be found in Section 14 of [1]. Here we only give a short summary. We expect that the reader is familiar with the standard notations, theorems and techniques of minimum cost flow problems with linear cost functions (see for example [1, Chapters 9-11.], [11, Chapter 3.6.] or [16, Chapter 12.]).

In the general convex cost flow problem a directed network $\mathcal{N}$ is given with:

- node set V and arc set A;
- an *excess* $b(v) \in \mathbb{R}$ for any $v \in V$ such that $\sum_{v \in V} b(v) = 0$;
- a *capacity* $u_{vw} \in \mathbb{R}^+$ and a *convex cost function* $c_{vw} : [0, u_{vw}] \to \mathbb{R}^+$ for all $\overrightarrow{vw} \in A$ (c is convex in the common sense of convexity).

$f : A \to \mathbb{R}^+$ is a called a *flow* in $\mathcal{N}$ iff it satisfies the following two conditions:

$$\sum_{w:\overrightarrow{vw} \in A} f_{vw} - \sum_{w:\overrightarrow{wv} \in A} f_{wv} = b(v) \quad \text{for all} \quad v \in V \tag{1}$$

$$0 \leq f_{vw} \leq u_{vw} \quad \text{for all} \quad \overrightarrow{vw} \in A. \tag{2}$$

Our goal is to find an optimal flow in the following sense:

$$\sum_{\overrightarrow{vw} \in A} c_{vw}(f_{vw}) \to \min . \tag{3}$$

An important special case of the general convex cost flow problem is when c_{vw} is a piecewise linear integral convex cost function.

Definition 2.16. Let $u \in \mathbb{N}$. Then $c : [0, u] \to \mathbb{R}^+$ is a *piecewise linear integral convex cost function* iff c is convex, $c(k) \in \mathbb{Z}$ for all $k \in \mathbb{Z}$ and $c(x)$ is linear between any pair of consecutive integers.

In the case of piecewise linear integral convex cost functions, the convex cost flow problem can be transformed to a linear cost flow problem in the following way: we replace each arc $\overrightarrow{vw}$ by u_{vw} pieces of arcs, each with capacity 1 and the kth one with cost $c_{vw}(k) - c_{vw}(k-1)$ for all $k \in \{1, \ldots, u_{vw}\}$, we call this new network $\mathcal{N}'$. Since a minimal cost flow in $\mathcal{N}'$ does not use an arc vw unless all arcs from v to w with a lower cost are saturated, there is a natural correspondence between min cost flows in $\mathcal{N}$ and $\mathcal{N}'$.

As network $\mathcal{N}'$ has linear cost functions, optimal flows in $\mathcal{N}'$ can be described by optimality criteria using potential functions (see [1, Theorem 9.3]).

Proposition 2.17. *Suppose that in network $\mathcal{N}$, all excesses $b(v)$ and capacities u_{vw} are integers and cost functions c_{vw} are piecewise linear integral convex cost functions. Then there exists an integral* potential function *$\pi : V \to \mathbb{Z}$ such that a flow f is optimal if and only if it satisfies the following set of inequalities (so called* optimality conditions*):*

$$c_{vw}(f_{vw} + 1) - c_{vw}(f_{vw}) \geq \pi(w) - \pi(v) \geq c_{vw}(f_{vw}) - c_{vw}(f_{vw} - 1) \text{ for all } vw \in A.$$

Here $c_{vw}(-1)$ counts as $-\infty$ and $c_{vw}(u_{vw} + 1)$ counts as ∞, representing empty and saturated arcs.

The major drawback of the transformation is that it expands the network substantially. Note that the number of arcs in $\mathcal{N}'$ is typically not polynomial in $|V|$. Still, by a tricky scaling algorithm of [1, Section 14.5], one can determine the optimal integral flow and potential in (weakly) polynomial time.

Let U denote the largest arc capacity, C denote the largest cost function value and $S(|V|, |A|, C))$ denote the time needed for computing a shortest path in a network with $|V|$ nodes, $|A|$ arcs and cost functions bounded by C.

Theorem 2.18 ([1]:Theorem 14.1)**.** *There is a capacity scaling algorithm which obtains an integer optimal flow and an integer potential π for a convex cost flow problem in $O\left((|A| \cdot \log U) \cdot S(|V|, |A|, C)\right)$ time.*

Remark 2.19. $S(|V|, |A|, C)$ is polynomial if $\log(C)$ is polynomial, see for example [16, Chapter 7].

3 The main result

In this section we present the main theorem of this paper, which is the following.

Theorem 3.1. *There is a polynomial algorithm deciding the halting problem for an undirected multigraph G and a distribution $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$.*

The algorithm is based on the following theorem of An, Baker, Kuperberg and Shokrieh [2, Theorem 4.10.].

Theorem 3.2 (An–Baker–Kuperberg–Shokrieh)**.** *Given any $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$, there is an orientation $\mathcal{O}$ such that $x \sim \mathbf{d}^-_{\mathcal{O}}$.*
(The in-degree vector $\mathbf{d}^-_{\mathcal{O}}$ is considered as a chip-firing distribution here.)

Note that in [2], the theorem is stated for divisors of degree $g - 1 = M - n$, but they subtract 1 from the indegrees. Although multigraphs are not mentioned there, the original proof works for multigraphs without any difficulty. It is already noted in [2, Remark 4.14.] that while their proof is not algorithmic, Backman's *Unfurling algorithm* [3, Section 4] gives an algorithm which finds such an orientation. Notice that Backman's algorithm is polynomial for simple graphs but only pseudopolynomial for multigraphs. Therefore we need a new algorithm to prove the following theorem:

Theorem 3.3. *Let G be a multigraph G and $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$. Then there is a polynomial algorithm that computes an orientation $\mathcal{O}$ of G with $\mathbf{d}^-_{\mathcal{O}} \sim x$.*

Proof. Let $\mathcal{O}_1$ be an arbitrarily chosen consistent orientation of G. $\mathcal{O}_1$ being consistent, it defines a simple directed graph $G_1 = (V, A_1)$; a single arc $\overrightarrow{vw} \in A_1$ represents the multiedge vw if it is oriented from v to w in $\mathcal{O}_1$.

Let $d^-_1(v) = d^-_{\mathcal{O}_1}(v)$ denote the indegree of node v in the orientation $\mathcal{O}_1$. $\Gamma(v), \Gamma^+_1(v)$ and $\Gamma^-_1(v)$ denote the set of neighbors of v in undirected and directed sense:

$$\Gamma(v) = \{w : vw \in E\};\ \Gamma^+_1(v) = \{w : \overrightarrow{vw} \in A_1\};\ \Gamma^-_1(v) = \{w : \overrightarrow{wv} \in A_1\}.$$

Let $U = n \cdot \max_{v \in V}\{|x(v)|\}$. Now we are ready to define the auxiliary network $\mathcal{N}$.

1. The node set of $\mathcal{N}$ is V.
2. In $\mathcal{N}$ there is a *forward arc* $\overrightarrow{vw}$ (directed as in $\mathcal{O}_1$) and *backward arc* $\overrightarrow{wv}$ (directed oppositely as in $\mathcal{O}_1$) corresponding to every arc $\overrightarrow{vw} \in A_1$, both with capacity U.
3. The excess of node v is $b(v) = x(v) - d^-_1(v)$, for all $v \in V$.
4. We define a piecewise linear integral convex cost function on each arc $\overrightarrow{vw}$ of $\mathcal{N}$. On forward edges, the cost function is:

$$C^{\rightarrow}_{vw}(t) = \binom{\lfloor \frac{t}{m_{vw}} \rfloor}{2} + \left\lfloor \frac{t}{m_{vw}} \right\rfloor \left(\frac{t}{m_{vw}} - \left\lfloor \frac{t}{m_{vw}} \right\rfloor \right),$$

   which is the piecewise linear function with slope k between $k \cdot m_{vw}$ and $(k+1) \cdot m_{vw}$ for any $k \in \mathbb{N} \cup \{0\}$.

   On backward edges, the cost function is:

$$C^{\leftarrow}_{vw}(t) = \binom{\lfloor \frac{t}{m_{vw}} \rfloor + 1}{2} + \left(\left\lfloor \frac{t}{m_{vw}} \right\rfloor + 1 \right) \left(\frac{t}{m_{vw}} - \left\lfloor \frac{t}{m_{vw}} \right\rfloor \right),$$

   which is the piecewise linear function with slope k between $(k-1) \cdot m_{vw}$ and $k \cdot m_{vw}$ for any $k \in \mathbb{N}$.

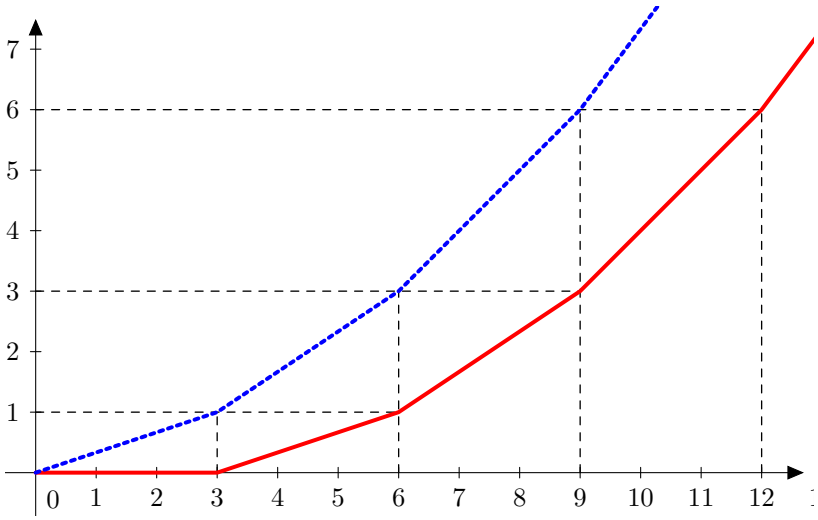

Figure 2: The cost function of the forward edge and the backward edge (dotted) obtained from an edge of G with multiplicity $m_{vw} = 3$.

As $C^{\rightarrow}_{vw}$ and $C^{\leftarrow}_{vw}$ are both piecewise linear integral convex cost functions, $\mathcal{N}$ satisfies the conditions of Proposition 2.17. Hence we can use the algorithm of Theorem 2.18 to calculate a min cost flow f and potential $\pi : V \to \mathbb{Z}$. Note that the sum of all positive excesses is less than U, hence no arcs are saturated by a min cost flow.

Remark 3.4. Note that the maximal capacity in the network is U and all capacities are bounded by $C' = \binom{U+1}{2}$. As $U \leq n \cdot M$, the bound given in Theorem 2.18 is polynomial in the input size. On the other hand, the $\log(U)$ factor means that we can only prove a weakly polynomial running time.

For any arc $\overrightarrow{vw} \in A_1$, let $f^{\rightarrow}(vw)$ and $f^{\leftarrow}(vw)$ denote the value of f on the forward arc and backward arc assigned to $\overrightarrow{vw}$, respectively. Note that, as f is an optimal cost flow, at least one of $f^{\rightarrow}(vw)$ and $f^{\leftarrow}(vw)$ is 0.

Let $f(vw) = f^{\rightarrow}(vw) - f^{\leftarrow}(vw)$ and $f_\pi(vw) = f(vw) - [\pi(w) - \pi(v)]m_{vw}$. By the optimality conditions of Proposition 2.17,

$$[\pi(v) - \pi(u)]m_{vw} \leq f(vw) \leq ([\pi(w) - \pi(v)] + 1)\, m_{vw}$$

hence $0 \leq f_\pi(vw) \leq m_{vw}$.

Now we are ready to define the desired orientation $\mathcal{O}$: for any arc $\overrightarrow{vw} \in A_1$, we reverse $f_\pi(vw)$ pieces of $\overrightarrow{vw}$, i.e. in $\mathcal{O}$ the edge vw has $\overrightarrow{m}_{vw} = m_{vw} - f_\pi(vw)$ pieces oriented from v to w and $\overleftarrow{m}_{vw} = f_\pi(vw)$ pieces oriented from w to v.

Claim 3.5. *The indegree vector of $\mathcal{O}$ is $\mathbf{d}^-_{\mathcal{O}}(v) = x - L\pi$.*

Proof. Consider any node $v \in V$. As f is a flow,

$$b(v) = x(v) - d^-_1(v) = \sum_{w \in \Gamma^+_1(v)} f(vw) - \sum_{w \in \Gamma^-_1(v)} f(wv). \tag{4}$$

By using the fact that $f(vw) = f_\pi(vw) + [\pi(w) - \pi(v)]\, m_{vw}$, we get

$$\begin{aligned}
&\sum_{w\in\Gamma_1^+(v)} f(vw) - \sum_{w\in\Gamma_1^-(v)} f(wv) = \\
= &\left(\sum_{w\in\Gamma_1^+(v)} f_\pi(vw) + \sum_{w\in\Gamma_1^+(v)} [\pi(w) - \pi(v)]\, m_{vw}\right) \\
- &\left(\sum_{w\in\Gamma_1^-(v)} f_\pi(wv) + \sum_{w\in\Gamma_1^-(v)} [\pi(v) - \pi(w)]\, m_{wv}\right) = \\
= &\sum_{w\in\Gamma_1^-(v)} f_\pi(vw) - \sum_{w\in\Gamma_1^-(v)} f_\pi(wv) + \sum_{w\in\Gamma(v)} [\pi(w) - \pi(v)]\, m_{vw}.
\end{aligned}$$

Reformulation and equation 4 gives:

$$d_1^-(v) + \sum_{w\in\Gamma_1^+(v)} f_\pi(vw) - \sum_{w\in\Gamma_1^-(v)} f_\pi(wv) = x(v) - \sum_{w\in\Gamma(v)} [\pi(w) - \pi(v)]\, m_{vw}.$$

Notice that by the definition of $\mathcal{O}$, the left hand side is exactly $\mathbf{d}^-_{\mathcal{O}}$. Hence

$$d^-_{\mathcal{O}}(v) = x(v) - \sum_{w\in\Gamma(v)} [\pi(v) - \pi(w)]\, m_{vw} = x(v) - L\pi(v)$$

for any v, which is exactly the statement of our claim. □

Claim 3.5 implies $\mathbf{d}^-_{\mathcal{O}} \sim x$, so $\mathcal{O}$ fulfills our requirements. □

Remark 3.6. The running time of the algorithm is dominated by the integral convex cost flow algorithm, which is polynomial by Remark 3.4.

Remark 3.7. Note that the idea of using flow algorithms has already appeared in Backman's paper [3, Algorithm 7.7]. The main difference in comparison to his work is that while he uses alternating phases of MFMC subroutines and cut reversing, we can compress these into only one flow cost minimizing subroutine. In Backman's algorithm, the number of phases is only pseudopolynomial, so this improvement is essential in the case of large edge multiplicities.

Another interesting feature of our algorithm is that the dual variable π has a meaning as a firing vector of an unconstrained chip-firing game.

Proof of Theorem 3.1. The algorithm consists of two main parts:

(P1) Compute an orientation $\mathcal{O}$ with $\mathbf{d}^-_{\mathcal{O}} \sim x$. This can be done in polynomial time by 3.3.

(P2) Decide whether the sink-reversal game started from $\mathcal{O}$ is terminating or non-terminating. This can be done in $O(|E|)$ by Proposition 2.14.

By Lemma 2.7 and Claim 2.8, x is terminating if and only if the sink-reversal game started from $\mathcal{O}$ is terminating. □

4 Complements

Let us recall a theorem from [6] and reformulate it for multigraphs.

Theorem 4.1. *[6, Theorem 3.3] Consider a multigraph on n nodes with the sum of edge multiplicites being M and let $x \in \mathrm{Chip}(G)$.*

(a) If $\deg(x) > 2M - n$, then x is non-terminating.

(b) If $M \leq \deg(x) \leq 2M - n$, then x can be either terminating or non-terminating.

(c) If $\deg(x) < M$, then x is terminating.

It tells us that the special case solved in this paper is the minimal nontrivial case. Note that at the other end, things are much simpler.

Claim 4.2. *Let $x \in \mathrm{Chip}(G)$ with $\deg(x) = 2M - n$. Then x is terminating if and only if $x \sim \mathbf{d}_G - \mathbf{1}$, where $\mathbf{d}_G - \mathbf{1}$ denotes the chip-distribution with $d(v) - 1$ chips on every node v.*

Proof. The only stable configuration with $\deg(x) = 2M - n$ is $\mathbf{d}_G - \mathbf{1}$. Lemma 2.7 completes the proof. □

Remark 4.3. Linear equivalence is decidable in polynomial time, see [14] for details.

Backman [3, Theorem 4.10] gave a generalization of Theorem 3.2. By reformulating it to the theory of chip-firing games, we get the following theorem.

Theorem 4.4 (Reformulation of Theorem 4.10. of [3])**.** *(1) If a chip-distribution x with $\deg(x) \geq M$ is terminating, then it is linearly equivalent to the in-degree vector of a sink-free partial orientation.*
(2) If a chip-distibution x with $\deg(x) \geq M$ is non-terminating, then it is linearly equivalent to a chip-distribution y such that $y \geq \mathbf{d}^-_{\mathcal{O}}$, where $\mathcal{O}$ is an acyclic partial orientation.

Backman also gives an algorithm for computing these orientations. His algorithm is polynomial for simple graphs but not polynomial for multigraphs. It can be the subject of some future work to upgrade this algorithm to run in polynomial time.

4.1 Graph divisor theory

In [4], Baker and Norine established graph divisor theory (or Riemann–Roch-theory of graphs) as a discrete analogue of the classical Riemann–Roch-theory of algebraic curves. *Graph divisors* are integer-valued functions on the node set of a graph. Non-negative valued divisors are called *effective*. The basic question about a divisor is whether it is linearly equivalent to an effective divisor (we will shortly call such divisors *equi-effective*). As pointed out by Baker and Norine in [4, Section 5.5], there is a dual relationship between chip-firing games and graph divisor theory, which is described by the following proposition:

Proposition 4.5. *[4, Corollary 5.4] $x \in \mathrm{Chip}(G)$ is terminating if and only if the divisor $D = \mathbf{d_G} - \mathbf{1} - x$ (i.e. $D(v) = d(v) - 1 - x(v)$ for all node v) is equi-effective.*

Notice that non-terminating distributions of M chips correspond to divisors of degree $M - n$ that are not equi-effective. These divisors are analogous to *non-special divisors* of Riemann surfaces [4, Remark 2.5].

In graph divisor theory, the standard way of deciding whether a divisor is equivalent to an effective divisor is by v_0-reducing it.

Definition 4.6. [18] For a fixed $v_0 \in V$, a divisor D is *v_0-reduced* if both of the following conditions hold:

- $D(v) \geq 0$ for all $v \in V \setminus \{v_0\}$.
- For any $\emptyset \neq S \subseteq V \setminus \{v_0\}$ there is some $v \in S$ such that $D(v)$ is less than the number of edges between v and $V \setminus S$.

Note that v_0-reduced divisors are also called as *G-parking functions*.

The following proposition, which is a combination of [4, Proposition 3.1.] and [18, Theorem 3.18.] shows how v_0-reduction decides whether a divisor is equi-effective.

Proposition 4.7. *Fix a node v_0. Then for every divisor D of G, there exists a unique v_0-reduced divisor D' such that $D' \sim D$. D is equi-effective if and only if $D'(v_0) \geq 0$.*

In [18, Section 5.1], an algorithm is given for v_0-reducing a divisor of a multigraph. The algorithm is polynomial for simple graphs but only pseudopolynomial for multigraphs.

Problem 4.8. [12] Is there a polynomial algorithm for v_0-reducing divisors in multigraphs?

Remark 4.9.

1. An affirmative answer would prove Conjecture 1.1.
2. It would be enough to give a v_0-reducing algorithm for divisors with a fixed degree of k. We can arbitrarily choose a node v_0, subtract $k - \deg(D)$ from $D(v_0)$, v_0-reduce the resulting divisor. Then by adding back $k - \deg(D)$ to v_0, we get the desired v_0-reduced divisor $D' \sim D$.

Acknowledgements

The author would like to say thanks to András Frank for suggesting the idea of considering min cost flows; to Dion Gijswijt for calling the author's attention to the multigraphs case by Problem 4.8; to Viktor Kiss for Example 2.4. Special thanks to Lilla Tóthmérész for fruitful discussions about the topic and many valuable comments about the manuscript.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] Y. An, M. Baker, G. Kuperberg, and F. Shokrieh. Canonical representatives for divisor classes on tropical curves and the matrix-tree theorem. *Forum of Mathematics, Sigma*, 2:e24 (25 pages), 2014.

[3] S. Backman. Riemann-Roch theory for graph orientations. *Preprint*, `http://arxiv.org/abs/1401.3309`, 2015.

[4] M. Baker and S. Norine. Riemann–Roch and Abel–Jacobi theory on a finite graph. *Adv. Math.*, 215(2):766–788, 2007.

[5] A. Björner and L. Lovász. Chip-firing games on directed graphs. *J. Algebraic Combin.*, 1(4):305–328, 1992.

[6] A. Björner, L. Lovász, and P. W. Shor. Chip-firing games on graphs. *European J. Combin.*, 12(4):283–291, 1991.

[7] B. Bond and L. Levine. Abelian networks I. Foundations and examples. *SIAM J. Discrete Math.*, 30(2):856–874., 2016.

[8] D. Erdős, A. Frank, and K. Kun. Sink-stable sets of digraphs. *SIAM J. Discrete Math.*, 28(4):1651–1674, 2014.

[9] K. Eriksson. No polynomial bound for the chip firing game on directed graphs. *Proc. Amer. Math. Soc.*, 112(4):1203–1205, 1991.

[10] M. Farrell and L. Levine. Coeulerian graphs. *Proc. Amer. Math. Soc.*, 144:2847–2860, 2016.

[11] A. Frank. *Connections in Combinatorial Optimization*. Oxford University Press, 2011.

[12] D. Gijswijt. Private communication. 2013.

[13] E. Gioan. Enumerating degree sequences in digraphs and a cycle– cocycle reversing system. *European J. Combin.*, 28(4):1351–1366, 2007.

[14] B. Hujter, V. Kiss, and L. Tóthmérész. On the complexity of the chip-firing reachability problem. *to appear: Proc. Amer. Math. Soc.*, 2017.

[15] B. Hujter and L. Tóthmérész. Chip-firing based methods in the Riemann–Roch theory of directed graphs. EGRES TR 16-01.

[16] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency. Vol. A. , Paths, flows, matchings. chapters 1-38*. Algorithms and combinatorics. Springer-Verlag, Berlin, Heidelberg, New York, N.Y., et al., 2003.

[17] G. Tardos. Polynomial bound for a chip firing game on graphs. *SIAM J. Discrete Math.*, 1(3):397–398, 1988.

[18] J. van Dobben de Bruyn. Reduced divisors and gonality in finite graphs. Bachelor's thesis, Mathematisch Instituut, Universiteit Leiden, 2012.