Séminaire Lotharingien de Combinatoire **86B** (2022)
Article #73, 12 pp.

Proceedings of the 34th Conference on Formal Power Series and Algebraic Combinatorics (Bangalore)

Rotor-Routing Induces the Only Consistent Sandpile Torsor Structure on Plane Graphs

Ankan Ganguly*[1] and Alex McDonough† [2]

[1]*Department of Applied Mathematics, Brown University. Supported by the United States Army Research Office under grant number W911NF2010133.*
[2]*Department of Mathematics, University of California, Davis*

Abstract. A sandpile torsor algorithm is a map which associates each plane graph (*i.e.* planar embedding) with a free transitive action of its sandpile group on its spanning trees. We define a notion of consistency, which requires a torsor algorithm to be preserved with respect to a certain class of contractions and deletions. We then show that the rotor-routing sandpile torsor algorithm (which is equivalent to the Bernardi algorithm) is consistent. Furthermore, we demonstrate that there are only three other consistent algorithms, which all have the same structure as rotor-routing. This proves a conjecture of Klivans. The paper corresponding to this extended abstract can be found at [8].

Keywords: sandpile group, sandpile torsor, rotor-routing

1 Introduction

Let G be a finite connected multigraph. The *sandpile group* of G, which we denote $\mathrm{Pic}^0(G)$, is a finite abelian group given by the cokernal of the *graph Laplacian matrix* over the integers. A remarkable fact, which follows from Kirchhoff's matrix-tree theorem, is that the size of $\mathrm{Pic}^0(G)$ is equal to the number of *spanning trees* of G (see [4, Theorem 7.3]). The traditional proof of this result is non-bijective, and for many graphs, no automorphism invariant bijections exist unless G is given additional structure (see [15, Theorem 8.1]). This problem is often resolved by working with *ribbon graphs*.

Let χ specify a cyclic ordering on the edges adjacent to each vertex of G. The pair (G, χ) is called a *ribbon graph*. Holroyd *et al.* demonstrated that the *rotor-routing algorithm* induces a *free transitive group action* of $\mathrm{Pic}^0(G)$ on $\mathcal{T}(G)$, which depends on the ribbon structure χ and a choice of *sink* vertex [9]. In particular, this makes $\mathcal{T}(G)$ a $\mathrm{Pic}^0(G)$-*torsor*, so we will call such an action a *sandpile torsor action*.

On a MathOverflow post, Ellenberg asked if there exist certain classes of ribbon graphs for which natural sandpile torsor actions can be defined without requiring a

*ankan_ganguly@brown.edu
†amcd@ucdavis.edu

sink vertex [7]. Chan, Church, and Grochow resolved this question by showing that the rotor-routing sandpile torsor action on (G, χ) is independent of the sink vertex if and only if (G, χ) is a *plane graph* [5, Theorem 2].[1] With this result in mind, we define a *sandpile torsor algorithm* to be a function which assigns a sandpile torsor action to every plane graph (G, χ) independent of the choice of sink vertex.

Another sink-dependent sandpile torsor action is given by the *Bernardi algorithm* [3], which is distinct from rotor-routing on non-planar ribbon graphs. Baker and Wang showed that the restriction of the Bernardi algorithm to plane graphs is also a sandpile torsor algorithm [2, Theorem 5.1]. Furthermore, they showed that this sandpile torsor algorithm is equivalent to the rotor-routing algorithm (when restricted to plane graphs) [2, Theorem 7.1]. This surprising equivalence motivated the following conjecture by Klivans, which we prove in this paper.

Conjecture 1.1 ([10, Conjecture 4.7.17])**.** *For plane graphs, there is only one sandpile torsor structure.*

One of the challenges in resolving this conjecture is that one must introduce a reasonable definition of *sandpile torsor structure*. The rotor-routing sandpile torsor algorithm follows a simple set of rules that can be applied to any plane graph. With this in mind, we posit that any definition of *sandpile torsor structure* should include some notion of consistency of induced sandpile torsor actions between different plane graphs. However, it is difficult to find a suitable consistency condition because there is no general map between the sandpile groups of different graphs which preserves their structure. The key insight that motivated the writing of this paper is the formulation of a contraction-deletion based definition of consistency (see Definition 3.2 and Figure 1).

There are two main results in this paper. In Theorem 3.5, we show that rotor-routing is *consistent*. In Theorem 4.4 we show that there are only 3 other consistent sandpile torsor algorithms, which can be obtained from rotor-routing by reversing the cyclic order on the vertices, taking the inverse action, or both. We say that these algorithms all have the same *structure* as rotor-routing. One powerful tool that we introduce is Proposition 4.5, which implies that on any 2-connected graph, it is possible to transform one spanning tree to any other by repeatedly swapping leaf edges with edges not on the tree.

2 Background and Definitions

[1]Following the language of [2], we distinguish between a *plane graph*, which is a particular planar embedding, and a *planar graph* which is a graph where such an embedding exists.

2.1 Divisors, Ribbon Graphs, and the Sandpile Group

In this section, we use much of the notation from [5]. See also [6, 10] for more information on sandpile groups and chip-firing.

Let G be a finite, connected, undirected graph with vertices $V(G)$ and edges $E(G)$. We allow multiple edges but not loop edges. For an edge $e \in E(G)$, we write $i(e)$ for the unordered pair of vertices incident to e, which we write in set notation.

Definition 2.1. The group $\mathrm{Div}^0(G)$ of *degree-0 divisors* of G is given by

$$\mathrm{Div}^0(G) := \left\{ \sum_{v \in V(G)} n_v v \mid n_v \in \mathbb{Z}, \sum_{v \in V(G)} n_v = 0 \right\}.$$

For $D = \sum n_v v$, if there exists an $s \in V(G)$ such that $n_v \geq 0$ for $v \neq s$, then we say that $D \in \mathrm{Div}^0_s(G)$.

Note that while $\mathrm{Div}^0(G)$ is a group, $\mathrm{Div}^0_s(G)$ is a *monoid* because its nonzero elements lack inverses. The *Laplacian matrix* Δ of G is the symmetric matrix defined by $\Delta = \deg(G) - A$, where A is the adjacency matrix of G and $\deg(G)$ is the $|V(G)| \times |V(G)|$ diagonal matrix such that $\deg(G)_{vv} = \deg(v)$ for every $v \in V(G)$.

Definition 2.2. The *sandpile group* of G, denoted $\mathrm{Pic}^0(G)$, is the quotient

$$\mathrm{Pic}^0(G) := \mathrm{Div}^0(G) / \operatorname{im}_{\mathbb{Z}}(\Delta).$$

One way to explore properties of $\mathrm{Pic}^0(G)$ is in terms of "chip-firing", also called "the dollar game", which we describe below.

Each element $D := \sum_{v \in V(G)} n_v v \in \mathrm{Div}^0(G)$ may be viewed as a configuration of "chips" placed on the vertices of G, also allowing negative "debt" chips (where the total number of chips matches the total amount of debt). Note that elements of $\mathrm{Div}^0_s(G)$ have all of their debt on the sink. In this context, divisors are also called *chip configurations*. Given a chip configuration D, we may "fire" a vertex $v \in V(G)$ by removing $\deg(v)$ chips from v and placing one chip on each of the neighbors of v. That is, v "gifts" a chip to each of its neighbors. For our purposes, we allow a vertex to fire regardless of the number of chips, even if this puts the vertex in debt.

Each row of the Laplacian corresponds to a vertex firing. Thus, the image of the Laplacian describes an equivalence relation on chip configurations, where D_1 and D_2 are *firing equivalent* if D_2 can be obtained from D_1 by a sequence of vertex firings. This means that two chip configurations are firing equivalent precisely when they are both representatives of the same equivalence class of $\mathrm{Pic}^0(G)$.

Throughout this paper, we will put brackets around a divisor to indicate the corresponding equivalence class of $\mathrm{Pic}^0(G)$. As discussed in the previous paragraph, for

$D, D' \in \mathrm{Div}^0(G)$, $[D] = [D']$ if and only if these two divisors are firing equivalent. We consider $D, D' \in \mathrm{Div}_s^0(G)$ to be firing equivalent if they are firing equivalent as elements of $\mathrm{Div}^0(G)$.

Let $\mathcal{T}(G)$ be the set of spanning trees of G. For convenience, we will treat each $T \in \mathcal{T}(G)$ as a subset of $E(G)$ when discussing set containment and as a subgraph of G when contracting or deleting edges. The following is a version of Kirchhoff's celebrated matrix-tree theorem.

Theorem 2.3 (Sandpile Matrix-Tree Theorem for Graphs [4, Theorem 7.3])**.**

$$|\operatorname{Pic}^0(G)| = |\mathcal{T}(G)|.$$

Theorem 2.3 implies the existence of bijections between equivalence classes of $\mathrm{Pic}^0(G)$ and elements of $\mathcal{T}(G)$. However, in order to define sufficiently "natural" bijections, we will need to give G some additional structure (see [15] and [5] for discussion on the need for additional structure).

Let χ be a function which maps each $v \in V(G)$ to a cyclic order of edges around v. The pair (G, χ) is called a *ribbon graph*. Suppose that G can be drawn in a plane such that for every $v \in V(G)$, $\chi(v)$ gives the edges incident to v in counterclockwise order. Then, (G, χ) is called a *plane graph*. A graph G is called *planar* if there exists some χ such that (G, χ) is a plane graph.

Suppose $e \in E(G)$ with $i(e) = \{v, w\}$. We will write $\chi(v, e)$ for the edge after e that is incident to v with respect to χ.

Definition 2.4. Let (G, χ) and (G', χ') be two ribbon graphs. Then a *ribbon graph isomorphism* is a bijection, $\phi : V(G) \cup E(G) \to V(G') \cup E(G')$ such that $\phi(V(G)) = V(G')$ and for any $e \in E(G)$ such that $i(e) = \{v_1, v_2\}$,

1. $i(\phi(e)) = \{\phi(v_1), \phi(v_2)\}$;
2. $\phi(\chi(v_i, e)) = \chi'(\phi(v_i), \phi(e))$, for $i \in \{1, 2\}$.

A *ribbon graph automorphism* is a ribbon graph isomorphism from a ribbon graph to itself.

Lemma 2.5. *Any ribbon graph isomorphism from (G, χ) to (G', χ') induces an bijection from $\mathcal{T}(G)$ to $\mathcal{T}(G')$. Furthermore, ϕ induces an isomorphism ϕ^{Div} from $\mathrm{Div}^0(G)$ to $\mathrm{Div}^0(G')$ which also describes isomorphisms from $\mathrm{Div}_s^0(G)$ to $\mathrm{Div}_{\phi(s)}^0(G')$ and $\mathrm{Pic}^0(G)$ to $\mathrm{Pic}^0(G')$.*

2.2 Sandpile Torsor Actions and Sandpile Torsor Algorithms

Given a ribbon graph (G, χ) and a vertex $s \in V(G)$, we will work with a function $\alpha_{(G,\chi,s)} : \mathrm{Div}_s^0(G) \times \mathcal{T}(G) \to \mathcal{T}(G)$ which belongs to a specific class of *monoid actions*.

Definition 2.6. A function of the form $\widehat{\alpha}_{(G,\chi,s)}$ is called a *sink-dependent sandpile torsor action* if for any $T \in \mathcal{T}(G)$, and $D, D' \in \mathrm{Div}_s^0(G)$, the following properties are satisfied:

1. $\widehat{\alpha}_{(G,\chi,s)}(D,T) = \widehat{\alpha}_{(G,\chi,s)}(D',T)$ if and only if $[D] = [D']$.
2. $\widehat{\alpha}_{(G,\chi,s)}([0],T) = T$.
3. $\widehat{\alpha}_{(G,\chi,s)}(D+D',T) = \widehat{\alpha}_{(G,\chi,s)}(D, \widehat{\alpha}_{(G,\chi,s)}(D',T))$.
4. For any $T' \in \mathcal{T}(G)$, there exists some $D'' \in \mathrm{Div}_s^0(G)$ such that $\widehat{\alpha}_{(G,\chi,s)}(D'',T) = T'$.

Definition 2.7. The function $\widehat{\alpha}$ is called a *sink-dependent sandpile torsor algorithm* if for any ribbon graph (G,χ) and $s \in V(G)$, the following conditions hold:

1. $\widehat{\alpha}_{(G,\chi,s)}$ is a sink-dependent sandpile torsor action.
2. For any ribbon graph isomorphism ϕ from (G,χ) to (G',χ'), and any $D \in \mathrm{Div}_s^0(G)$,
$$\phi(\widehat{\alpha}_{(G,\chi,s)}(D,T)) = \widehat{\alpha}_{(G',\chi',\phi(s))}(\phi^{\mathrm{Div}}(D), \phi(T)).$$

Definition 2.8. For every *plane graph* (G,χ), let $\alpha_{(G,\chi)}$ be an action of $\mathrm{Pic}^0(G)$ on $\mathcal{T}(G)$. We call α a *sandpile torsor algorithm* if there is a sink-dependent sandpile torsor algorithm $\widehat{\alpha}$ such that for every *plane graph* (G,χ), $s \in V(G)$, $D \in \mathrm{Div}_s^0(G)$, and $T \in \mathcal{T}(G)$,
$$\alpha_{(G,\chi)}([D],T) = \widehat{\alpha}_{(G,\chi,s)}(D,T).$$
We call a specific action of the form $\alpha_{(G,\chi)}$ a *sandpile torsor action*.

Remark 2.9. If we were to replace *plane graph* with *ribbon graph* in Definition 2.8, then it would follow that no sandpile torsor algorithms exist (see [12, Figure 1]).

Notice that a sandpile torsor algorithm α is uniquely defined by a sink-dependent sandpile torsor algorithm $\widehat{\alpha}$ such that for every plane graph (G,χ), $s, s' \in V(G)$, $D \in \mathrm{Div}_s^0(G)$, $D' \in \mathrm{Div}_{s'}^0(G)$, and $T \in \mathcal{T}(G)$,
$$\widehat{\alpha}_{(G,\chi,s)}(D,T) = \widehat{\alpha}_{(G,\chi,s')}(D',T) \text{ if and only if } [D] = [D']. \tag{2.1}$$

Two well-known sink-dependent sandpile torsor algorithms are the *rotor-routing algorithm* and the *Bernardi algorithm*. These algorithms are distinct in general, and their constructions appear unrelated [2, Figure 9]. However, both algorithms satisfy (2.1) [5, 2], and they both define the same sandpile torsor algorithm [2, Theorem 7.1]. This surprising equivalence inspired Conjecture 1.1, which asks if this sandpile torsor algorithm is in some sense unique. We will not explicitly work with the Bernardi algorithm in this paper, but we encourage the curious reader to see [2] for its construction, as well as [16] for an alternate perspective when restricting to plane graphs.

2.3 Rotor-Routing

Let (G, χ) be a ribbon graph and $s \in V(G)$.

Definition 2.10. A *rotor configuration with sink s* is an assignment of an incident edge to every vertex of G except for s. Given a rotor configuration ρ and $x \in V(G) \setminus s$, we write $\rho\langle x\rangle$ for the edge assigned to x.

We will call these edges *rotors*. It is useful to visualize a rotor $\rho\langle x\rangle$ as a directed edge adjacent to x and oriented away from x. Given a fixed sink $s \in V(G)$, any tree $T \in \mathcal{T}(G)$ can be uniquely represented by a rotor configuration T_s such that for all $x \in V(G) \setminus \{s\}$, $T_s\langle x\rangle$ is the unique edge in T incident to x along the path from x to s. Then, T_s describes an orientation of the edges of T such that each edge is directed toward s. Given a sandpile element of the form $c - s \in \mathrm{Div}^0_s(G)$ for some $c, s \in V(G)$ (which we will call the chip and sink vertices respectively), Algorithm 1 describes a sequence of rotor configurations in which at each step, a single rotor rotates according to the cyclic order χ (on a plane graph, this would be a counterclockwise rotation). The output rotor configuration will be of the form T'_s for some other spanning tree $T' \in \mathcal{T}(G)$, which we will use to construct a sink-dependent sandpile torsor algorithm r.

Algorithm 1 Single-chip rotor-routing

```
    Input: A tree $T \in \mathcal{T}(G)$ and a divisor of the form $c - s \in \mathrm{Div}^0_s(G)$.
    Output: The rotor configuration $\rho$ with sink vertex $s$.
Initialize the rotor configuration $\rho = T_s$.
Initialize a "traveling" vertex $x$ at $c$. We call $x$ a chip.
while $x \neq s$ do
    Update $\rho$: $\rho\langle x\rangle = \chi(x, \rho\langle x\rangle)$.      ▷ rotate the rotor at $x$ by one position.
    Replace $x$ with the other vertex incident to $\rho\langle x\rangle$.      ▷ move $x$ across the rotor.
end while
return $\rho$.
```

Algorithm 1 was first explored by Priezzhev *et al.* under the name *Eulerian walkers model* [14]. By [9, Lemma 3.10], the final rotor ρ in Algorithm 1 is acyclic which implies the existence of a unique tree $T' \in \mathcal{T}(G)$ such that $T'_s = \rho$. Then, for the inputs $c, s \in V(G)$ and $T \in \mathcal{T}(G)$, we can define the mapping $r_{(G,\chi,s)}(c - s, T) := T'$ where $T'_s = \rho$ is the output of Algorithm 1.

Holroyd *et al.* showed how this model could be used to define a sink-dependent sandpile torsor action on any ribbon graph (G, χ). Fix any $D = \sum_{v \in V(G)} n_v v \in \mathrm{Div}^0_s(G)$ and let $N := -n_s$. Note that $D = \sum_{i=1}^{N}(c_i - s)$ for some sequence of vertices $(c_1, \ldots, c_N)$. Let $T^0 = T$, and for each $i \in [1, N]$, let $T^i = \widehat{r}_{(G,\chi,s)}(c_i - s, T^{i-1})$. Then, we define

$\widehat{r}_{(G,\chi,s)}(D,T) := T^N$. By [9, Corollary 2.6], T^N does not depend on the ordering of $(c_1,\dots,c_N)$. Thus, $\widehat{r}_{(G,\chi,s)}$ is a well-defined monoid action of $\mathrm{Div}^0_s(G)$ on $\mathcal{T}(G)$.

Theorem 2.11 ([11, Theorem 2.5]). *The rotor-routing algorithm $\widehat{r}$ constructed above is a sink-dependent sandpile torsor algorithm.*

Chan, Church, and Grochow proved that, on plane graphs, the rotor-routing action does not depend on the choice of sink vertex, which implies the following result:

Theorem 2.12. *[5, Theorem 2] The rotor-routing algorithm $\widehat{r}$ satisfies (2.1). In particular, it defines a sandpile torsor algorithm r.*

3 Rotor-Routing is Consistent

In this section, we introduce the notion of *consistency* of sandpile torsor algorithms and show that the rotor-routing algorithm is consistent.

Throughout this section, (G,χ) is a plane graph, $f \in E(G)$, and $i(f) = \{c,s\}$. For $e \in E(G)$, we write $G \setminus e$ for the graph we obtain from G by *deleting* edge e and G/e for the graph we obtain by *contracting* the edge e. After contracting e, we also remove all loop edges because these cannot occur on any spanning tree and have no effect on the rotor-routing algorithm.

Definition 3.1. Let (G,χ) be a ribbon graph with $e \in E(G)$, where $i(e) = \{x,y\}$.

- Define $\chi \setminus e$ to be equal to χ, except with e removed from $\chi(x)$ and $\chi(y)$.
- Suppose that after removing edges parallel to e, $\chi(x) = (e,e_1,e_2,\dots,e_p)$ and $\chi(y) = (e,\widehat{e}_1,\widehat{e}_2,\dots,\widehat{e}_l)$. Define χ/e to be equal to χ except that on the contracted edge z, we have
$$(\chi/e)(z) := (e_1,e_2,\dots,e_p,\widehat{e}_1,\widehat{e}_2,\dots,\widehat{e}_l).$$

Notice that $(G\setminus e,\chi\setminus e)$ and $(G/e,\chi/e)$ are both plane graphs. If (G',χ') can be obtained from (G,χ) through deletion and contraction, we say (G',χ') is a *minor* of (G,χ).

Definition 3.2. A sandpile torsor algorithm α is *consistent* if for every plane graph (G,χ), every choice of $f \in E(G)$, and every choice of $T \in \mathcal{T}(G)$, the following three properties hold (where we define $\{c,s\} = i(f)$):

1. For any $e \in E(G)$ such that $i(e) \neq \{c,s\}$, if $e \in T \cap \alpha_{(G,\chi)}([c-s],T)$, then
$$\alpha_{(G,\chi)}([c-s],T)\setminus e = \alpha_{(G/e,\chi/e)}([c-s],T\setminus e).$$
2. For any $e \in E(G)$, if $e \notin T \cup \alpha_{(G,\chi)}([c-s],T)$, then
$$\alpha_{(G,\chi)}([c-s],T) = \alpha_{(G\setminus e,\chi\setminus e)}([c-s],T).$$

3. For any $e \in E(G) \setminus f$, if there is a cut vertex x such that all paths from e to f pass through x, then
$$e \in T \text{ if and only if } e \in \alpha_{(G,\chi)}([c-s], T).$$

The first two conditions of Definition 3.2 are illustrated by Figure 1. The third condition says that we can treat two subgraphs joined at a cut vertex independently.

Proposition 3.3. *Condition 3 of Definition 3.2 is satisfied by the rotor-routing algorithm.*

By requiring consistent sandpile torsor algorithms to satisfy Condition 3, we simplify the proofs of many of the results in Section 4. However, we are not convinced that this condition is necessary.

Question 3.4. *Is Condition 3 of Definition 3.2 implied by the other two?*

Theorem 3.5. *The rotor-routing sandpile torsor algorithm is consistent.*

We reduce this result to 6 specific cases. If an edge e is in both T and $T' := r_{(G,\chi)}([c - s], T)$, then the rotor at e may be untouched during rotor-routing, it may complete a full rotation, or it could be oriented in opposite directions in T_s and T'_s. When the edge e is in neither tree, the travelling vertex may never have crossed e, it may have crossed e in both directions, or it may have only crossed e in one direction. For some of these cases, the path of the traveling vertex will change after deleting or contracting e, but we show that it still crosses the same edges, just in a different order. The planarity condition comes into play for the final case (where the edge is crossed in one direction). We apply tools developed by Chan, Church, and Grochow to show that this case is never realized [5].

4 Uniqueness of Consistent Torsor Algorithms

In this section, we classify the consistent sandpile torsor algorithms. In particular, we show that every consistent sandpile torsor algorithm must either be rotor-routing, or one of three related algorithms which we say have the same *structure* as rotor-routing.

For any ribbon graph (G,χ), let $\overline{\chi}$ be the reverse cyclic order around each vertex. Notice that if (G,χ) is a plane graph, then $(G,\overline{\chi})$ is a plane graph as well (simply reflect the planar embedding of (G,χ) to get a planar embedding of $(G,\overline{\chi})$).

Definition 4.1. Suppose α is a sandpile torsor algorithm. Define $\overline{\alpha}$, α^{-1}, and $\overline{\alpha}^{-1}$ such that for any plane graph (G,χ), $S \in \mathrm{Pic}^0(G)$, and $T \in \mathcal{T}(G)$, we have:

$$\alpha_{(G,\chi)}(S,T) = \overline{\alpha}_{(G,\overline{\chi})}(S,T) = \alpha^{-1}_{(G,\chi)}(-S,T) = \overline{\alpha}^{-1}_{(G,\overline{\chi})}(-S,T).$$

If α is the rotor-routing algorithm, then $\overline{\alpha}$ reverses the direction in which the rotors turn, α^{-1} switches the role of the chip and sink, and $\overline{\alpha}^{-1}$ makes both of these changes.

Proposition 4.2. *If α is a consistent sandpile torsor algorithm, then $\overline{\alpha}$, α^{-1} and $\overline{\alpha}^{-1}$ are distinct consistent sandpile torsor algorithms.*

Definition 4.3. Sandpile torsor algorithms α and β have the same *sandpile torsor structure* if $\beta \in \{\alpha, \overline{\alpha}, \alpha^{-1}, \overline{\alpha}^{-1}\}$.

Our goal will be to prove the following version of Conjecture 1.1.

Theorem 4.4. *Every consistent sandpile torsor algorithm has the same sandpile torsor structure as the rotor-routing algorithm.*

To show that a sandpile torsor algorithm α is equivalent to rotor-routing, it suffices to check that α and r are equivalent for all $T \in \mathcal{T}(G)$ on a generating set of $\mathrm{Pic}^0(G)$. In fact, we can further simplify the task by applying the following result.

Proposition 4.5. *Suppose that (G, χ) is a 2-connected ribbon graph, $T, \widehat{T} \in \mathcal{T}(G)$, and $s \in V(G)$. It is possible to transform T_s to $\widehat{T}_s$ by only rotating rotors with indegree 0 (with respect to the current rotor configuration).*

We prove this proposition by defining a partial order on spanning trees such that $\widehat{T}$ is the unique minimal element. We then show that for any $T \in \mathcal{T}(G) \setminus \widehat{T}$, it is always possible to rotate rotors with indegree 0 to reach some T' that is smaller than T with respect to our partial order. This is further discussed in Figure 2.

Definition 4.6. Let $T \in \mathcal{T}(G)$ and $c, s \in V(G)$. The pair $(c - s, T)$ is called a *source-turn pair* if c is only incident to a single edge of T, and $i(\chi(c, T_s\langle c\rangle)) = \{c, s\}$.

Notice that if $(c - s, T)$ is a source-turn pair, then $T' := r_{(G,\chi)}([c - s], T)$ rotates the indegree 0 rotor at c by one position without changing any other rotors on the graph. Furthermore, setting s' to be the unique vertex such that $\{c, s'\} = i(\chi(c, T'_s\langle c\rangle))$, we have that $(c - s', T')$ is also a source-turn pair. Therefore, repeated application of the rotor-routing algorithm can be used to freely rotate the rotor at c without changing any other rotors on the graph. This implies the following corollary of Proposition 4.5.

Corollary 4.7. *Let (G, χ) be a 2-connected plane graph. If $\alpha_{(G,\chi)}([c - s], T) = r_{(G,\chi)}([c - s], T)$ whenever $(c - s, T)$ is a source-turn pair, then $\alpha_{(G,\chi)} = r_{(G,\chi)}$.*

We now sketch a proof of Theorem 4.4. Let α be a consistent sandpile torsor algorithm and (G, χ) be a 2-connected plane graph. Suppose for the sake of induction that $\alpha_{(G',\chi')} = r_{(G',\chi')}$ for all 2-connected minors (G', χ') of (G, χ). By Corollary 4.7, the result follows if we show that $\alpha_{(G,\chi)}([c - s], T) = r_{(G,\chi)}([c - s], T)$ for every source-turn pair $(c - s, T)$.

Let $(c - s, T)$ be an arbitrary source-turn pair, and suppose that $\alpha_{(G,\chi)}([c - s], T) = \widehat{T} \neq r_{(G,\chi)}([c - s], T)$. By applying properties of Definition 3.2, one can show that

$$E(G) \setminus (T \Delta \widehat{T}) \subseteq \{f, g\}, \tag{4.1}$$

where Δ denotes the symmetric difference operator. This leaves us with 4 cases to consider corresponding to the 4 subsets of $\{f, g\}$. The bulk of our argument is spent resolving these cases using a variety of contraction/deletion tricks, properties of $\mathrm{Pic}^0(G)$, tools developed by Chan, Church, and Grochow, and other techniques (see [8]).

5 Generalization to Regular Matroids

Regular matroids also have associated *sandpile groups* [13]. Given a regular matroid M and a choice of *acyclic circuit and cocircuit signatures*, Backman, Baker, and Yuen defined an explicit free transitive action of the sandpile group of M on its *bases* [1]. We call such an action a *matroidal sandpile torsor algorithm*. We can naturally generalize the notions of *consistency* and *sandpile torsor structure* to this context (see [8, Section 6]). We conclude with the natural extension of Conjecture 1.1 to regular matroids.

Conjecture 5.1. *For regular matroids, there is only one sandpile torsor structure.*

Acknowledgements

The authors would like to thank Caroline Klivans for initially suggesting the project and for helpful discussion as well as Richard Kenyon for suggesting we look for properties involving contractions and deletions. Finally, we thank Cyrus Peterpaul for many fruitful discussions on this topic.

References

[1] S. Backman, M. Baker, and C. H. Yuen. "Geometric bijections for regular matroids, zonotopes, and Ehrhart theory". *Forum Math. Sigma* **7** (2019).

[2] M. Baker and Y. Wang. "The Bernardi Process and Torsor Structures on Spanning Trees". *Int. Math. Res. Not. IMRN* (2017). DOI.

[3] O. Bernardi. "Tutte polynomial, subgraphs, orientations and sandpile model: new connections via embeddings". *Electron. J. Combin.* **15**.R109 (2008), p. 1.

[4] N. L. Biggs. "Chip-firing and the critical group of a graph". *J. Algebraic Combin.* **9**.1 (1999), pp. 25–45.

[5] M. Chan, T. Church, and J. A. Grochow. "Rotor-routing and spanning trees on planar graphs". *Int. Math. Res. Not. IMRN* **2015**.11 (2014).

[6] S. Corry and D. Perkinson. *Divisors and sandpiles*. Vol. 114. American Mathematical Soc., 2018.

[7] J. Ellenberg. "What is the sandpile torsor?" MathOverflow. 2011. Link.

[8] A. Ganguly and A. McDonough. "Rotor-routing induces the only consistent sandpile torsor structure on plane graphs". 2022. arXiv:2203.15079.

[9] A. E. Holroyd, L. Levine, K. Mészáros, Y. Peres, J. Propp, and D. B. Wilson. "Chip-firing and rotor-routing on directed graphs". *In and Out of Equilibrium 2*. Springer, 2008, pp. 331–364.

[10] C. Klivans. *The Mathematics of Chip firing*. Chapman and Hall, 2018.

[11] I. Landau and L. Levine. "The rotor–router model on regular trees". *J. Combin. Theory Ser. A* **116**.2 (2009), pp. 421–433.

[12] A. McDonough. "Determining genus from sandpile torsor algorithms". *Discrete Math. Theoret. Comput. Sci.* **23**.1 (Jan. 2021). Link.

[13] C. Merino. "Matroids, the Tutte polynomial and the chip firing game." PhD thesis. University of Oxford, 1999.

[14] V. B. Priezzhev, D. Dhar, A. Dhar, and S. Krishnamurthy. "Eulerian walkers as a model of self-organized criticality". *Physical Review Letters* **77**.25 (1996), p. 5079.

[15] D. G. Wagner. "The critical group of a directed graph". 2000. arXiv:math/0010241.

[16] C. H. Yuen. "Geometric bijections between spanning trees and break divisors". *J. Combin. Theory Ser. A* **152** (2017), pp. 159–189.

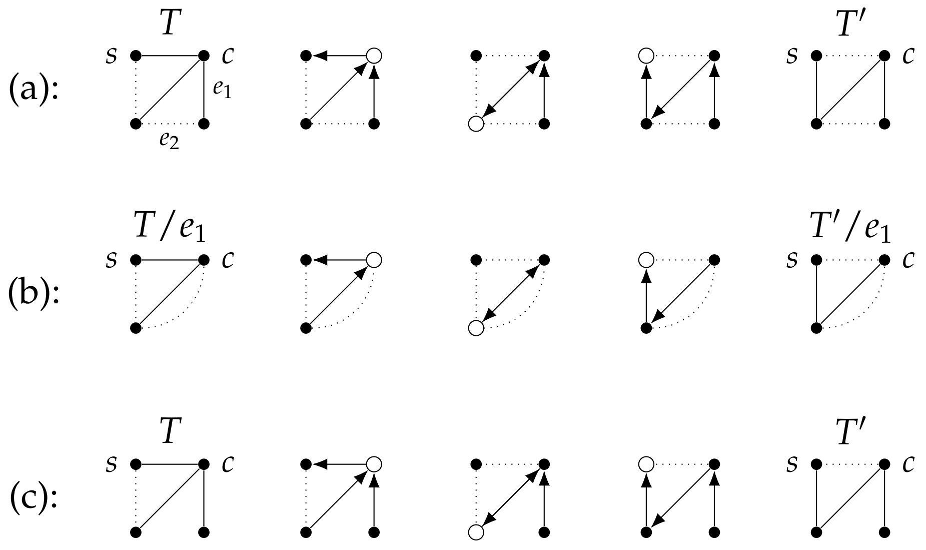

Figure 1: An example of the rotor-routing algorithm demonstrating that rotor-routing is consistent on a graph (G, χ), where χ denotes counterclockwise rotation. We denote the chip x by a hollow vertex. Figure 1(a) illustrates, using the rotor-routing algorithm, that $T' = r_{(G,\chi)}([c-s], T)$. Figure 1(b) illustrates how the rotor-routing algorithm commutes with contraction: $T'/e_1 = r_{(G/e_1,\chi)}([c-s], T/e_1)$. Figure 1(c) illustrates how the rotor-routing algorithm commutes with deletion: $T' = r_{(G\setminus e_2,\chi)}([c-s], T)$.

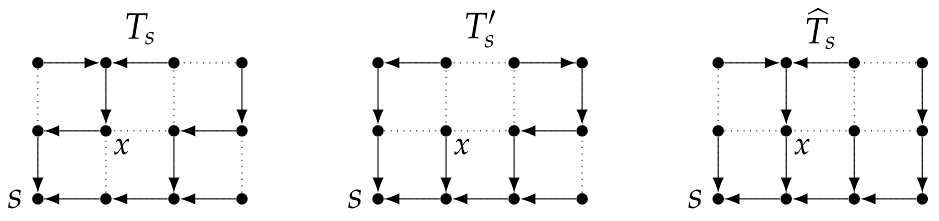

Figure 2: All of the indegree 0 rotors in T_s are in the same final position as they are in $\widehat{T}_s$, so their are no immediate moves to get "closer" to $\widehat{T}_s$. However, it is possible to reach T'_s which has rotor x in the correct position after only moving rotors that are further from s than x (with respect to $\widehat{T}$). This general method is used for Proposition 4.5.