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

FPSAC 2022

Alex McDonough (UC Davis)

Joint work with Ankan Ganguly (Brown University)

Full paper: arXiv:2203.15079

Animations: https://youtu.be/2StIAfnONMs

July 18, 2022
A *ribbon graph* $G$ (also called a *combinatorial map*) is a graph along with a choice of cyclic order of edges around each vertex (clockwise for this talk). Ribbon graphs are used to represent graph embeddings.

A *plane graph* is a ribbon graph with no edge crossings (a planar embedding). Of the ribbon graphs above, only the middle is a plane graph.
Single-Chip Rotor-Routing Algorithm (With Sink)

**Input:** a ribbon graph $G$, a spanning tree $T$, a *sink* vertex $s$, and a *chip* $c$ on any non-sink vertex.

1. Orient the edges of $T$ toward $s$. Every vertex $v \in V(G) \setminus s$ has a single outgoing edge called the *rotor at* $v$.

2. Rotate the rotor at $c$ and then move $c$ along it.

3. Repeat step 2 until $c$ reaches the sink, then remove $c$.

4. Forget the orientation of the rotors and let $T'$ be their edges.

**Output:** $T'$
Facts about Rotor-Routing

- Rotor-routing was introduced under the name “Eulerian Walkers Model” by Priezzhev, D. Dhar, A. Dhar, and Krishnamurthy in 1996. The following lemmas are implied by their results:

**Lemma**

The output $T'$ is always a spanning tree.

**Lemma**

If the single-chip rotor-routing algorithm is performed multiple times, the order of chips does not affect the final tree.

- The 2008 paper “Chip Firing and Rotor-Routing on Directed Graphs” by Holroyd, Levine, Mészáros, Peres, Propp, and Wilson is an excellent survey of rotor-routing and sandpile ideas.
Multiple-Chip Rotor-Routing Algorithm (With Sink)

**Input:** a ribbon graph $G$, a spanning tree $T$, a *sink* vertex $s$, and a collection $C$ of *chips* on non-sink vertices.

1. Orient the edges of $T$ toward $s$. Every vertex $v \in V(G) \setminus s$ has a single outgoing edge called the *rotor at* $v$.

2. Choose any $c \in C$. Rotate the rotor at $c$ and then move $c$ along it. If $c$ reaches the sink, remove it from $C$.

3. Repeat step 2 until $C = \emptyset$.

4. Forget the orientation of the rotors and let $T'$ be their edges.

**Output:** $T'$
Let $G$ be a finite connected graph with vertices $V(G)$.

A *degree 0 divisor* is an assignment of an integral number of “chips” to each vertex (allowing negative chips) so that there are 0 total chips.

The degree 0 divisors under pointwise addition form a group called $\text{Div}^0(G)$.

The *Laplacian matrix* $\Delta$ is $D - A$, where $D$ is the *degree matrix* of $G$ and $A$ is the *adjacency matrix* of $G$.

**Definition**

The *sandpile group* $S(G)$ is $\text{Div}^0(G)/\text{im}\mathbb{Z}(\Delta)$. 

See Clip 3
The Sandpile Group of a Graph

- Let \( G \) be a finite connected graph with vertices \( V(G) \).
- A *degree 0 divisor* is an assignment of an integral number of “chips” to each vertex (allowing negative chips) so that there are 0 total chips.
- The degree 0 divisors under pointwise addition form a group called \( \text{Div}^0(G) \).
- The *Laplacian matrix* \( \Delta \) is \( D - A \), where \( D \) is the *degree matrix* of \( G \) and \( A \) is the *adjacency matrix* of \( G \).

**Definition**

The *sandpile group* \( S(G) \) is \( \text{Div}^0(G)/\text{im}_\mathbb{Z}(\Delta) \).

**Theorem (sandpile matrix-tree theorem for graphs, Biggs 1999)**

The size of \( S(G) \) is the number of spanning trees of \( G \).
Sandpile Rotor-Routing Algorithm (With Sink)

**Input:** a ribbon graph $G$, a spanning tree $T$, a *sink* vertex $s$, and an element of the sandpile group $S \in S(G)$.

1. Orient the edges of $T$ toward $s$. Every vertex $v \in V(G) \setminus s$ has a single outgoing edge called the *rotor at* $v$. Let $D$ be any representative of $S$ such that $D(v) \geq 0$ for $v \neq s$. Let $C$ be a set of $D(v)$ chips at each $v \neq s$.

2. Choose any $c \in C$. Rotate the rotor at $c$ and then move $c$ along it. If $c$ reaches the sink, remove it from $C$.

3. Repeat step 2 until $C = \emptyset$

4. Forget the orientation of the rotors and let $T'$ be their edges.

**Output:** $T'$
Theorem (HLMPPW, 2008)

The algorithm in the previous slide is well defined.
Theorem (HLMPPW, 2008)

The algorithm in the previous slide is well defined. Furthermore, for any spanning trees $T$ and $T'$, there is exactly one $S \in S(G)$ that maps $T$ to $T'$.

- In other words, rotor routing defines a free transitive action of $S(G)$ on the spanning trees of $G$. 
The algorithm in the previous slide is well defined. Furthermore, for any spanning trees $T$ and $T'$, there is exactly one $S \in S(G)$ that maps $T$ to $T'$.

- In other words, rotor routing defines a free transitive action of $S(G)$ on the spanning trees of $G$.

Question (Ellenberg, 2012)
When is the rotor-routing action preserved after changing the sink vertex?
Theorem (HLMPPW, 2008)

The algorithm in the previous slide is well defined. Furthermore, for any spanning trees $T$ and $T'$, there is exactly one $S \in S(G)$ that maps $T$ to $T'$.

- In other words, rotor routing defines a free transitive action of $S(G)$ on the spanning trees of $G$.

Question (Ellenberg, 2012)

When is the rotor-routing action preserved after changing the sink vertex?

Theorem (Chan-Church-Grochow, 2013)

The rotor-routing action is preserved regardless of sink vertex if and only if $G$ is a plane graph.
Sink-Free Rotor-Routing Algorithm

**Input:** a plane graph $G$, a spanning tree $T$, and an element of the sandpile group $S \in S(G)$.

1. Choose any $s \in V(G)$. Orient the edges of $T$ toward $s$. Every vertex $v \in V(G) \setminus s$ has a single outgoing edge called the *rotor at $v$*. Let $D$ be any representative of $S$ such that $D(v) \geq 0$ for $v \neq s$. Let $C$ be a set of $D(v)$ chips at each $v \neq s$.

2. Choose any $c \in C$. Rotate the rotor at $c$ and then move $c$ along it. If $c$ reaches the sink, remove it from $C$.

3. Repeat step 2 until $C = \emptyset$

4. Forget the orientation of the rotors and let $T'$ be their edges.

**Output:** $T'$. We write that $r_G([D], T) = T'$. 

Alex McDonough (UC Davis) (UC Davis)
Definition

A *sandpile torsor action* on a plane graph $G$ is a free transitive action of $S(G)$ on the spanning trees of $G$. 
A sandpile torsor action on a plane graph $G$ is a free transitive action of $S(G)$ on the spanning trees of $G$.

A sandpile torsor algorithm is a function which assigns a sandpile torsor action to every plane graph.
Definition

A sandpile torsor action on a plane graph $G$ is a free transitive action of $S(G)$ on the spanning trees of $G$.

Definition

A sandpile torsor algorithm is a function which assigns a sandpile torsor action to every plane graph.

- We saw that rotor-routing induces a sandpile torsor algorithm, but are there other natural algorithms?
in 2012, Baker and Wang used the *Bernardi process* to define another sandpile torsor algorithm.
in 2012, Baker and Wang used the *Bernardi process* to define another sandpile torsor algorithm.

**Theorem (Baker-Wang, 2012)**

*On plane graphs, the rotor-routing algorithm and Bernardi algorithm are equivalent.*
in 2012, Baker and Wang used the *Bernardi process* to define another sandpile torsor algorithm.

**Theorem (Baker-Wang, 2012)**

*On plane graphs, the rotor-routing algorithm and Bernardi algorithm are equivalent.*

Other descriptions were found for this algorithm (see Yuen 2017 and Kálmán-Lee-Tóthmérsz 2022+), but these are still identical to rotor-routing.
“Other” Sandpile Torsor Algorithms

- in 2012, Baker and Wang used the \textit{Bernardi process} to define another sandpile torsor algorithm.

\textbf{Theorem (Baker-Wang, 2012)}

\textit{On plane graphs, the rotor-routing algorithm and Bernardi algorithm are equivalent.}

- Other descriptions were found for this algorithm (see Yuen 2017 and Kálmán-Lee-Tóthmérész 2022+), but these are still identical to rotor-routing.

\textbf{Conjecture (Klivans, 2018)}

For plane graphs, there is only one sandpile torsor structure.
in 2012, Baker and Wang used the *Bernardi process* to define another sandpile torsor algorithm.

**Theorem (Baker-Wang, 2012)**

*On plane graphs, the rotor-routing algorithm and Bernardi algorithm are equivalent.*

Other descriptions were found for this algorithm (see Yuen 2017 and Kálmán-Lee-Tóthmérész 2022+), but these are still identical to rotor-routing.

**Conjecture (Klivans, 2018)**

*For plane graphs, there is only one sandpile torsor structure.*
Proposition (Ganguly-M., 2022+)

Rotor-routing produces 4 closely related sandpile torsor algorithms:

- clockwise rotor-routing,
- counterclockwise rotor-routing,
- inverse clockwise rotor-routing, and
- inverse counterclockwise rotor-routing.
Proposition (Ganguly-M., 2022+)

Rotor-routing produces 4 closely related sandpile torsor algorithms:
- clockwise rotor-routing,
- counterclockwise rotor-routing,
- inverse clockwise rotor-routing, and
- inverse counterclockwise rotor-routing.

Definition

Two sandpile torsor algorithms have the same *structure* if they differ by inverting the action and/or the ribbon structure.
Proposition (Ganguly-M., 2022+)

Rotor-routing produces 4 closely related sandpile torsor algorithms:
- clockwise rotor-routing,
- counterclockwise rotor-routing,
- inverse clockwise rotor-routing, and
- inverse counterclockwise rotor-routing.

Definition

Two sandpile torsor algorithms have the same *structure* if they differ by inverting the action and/or the ribbon structure.

- To prevent simple but contrived counterexamples to Klivans’ conjecture, we want our algorithm to act *consistently* across different plane graphs.
A Consistency Condition

Theorem (Ganguly-M., 2022+)

Let $G$ be a plane graph with a spanning tree $T$, and incident vertices $c$ and $s$. Let $T' = r_G([c - s], T)$.

1. For any $e \in E(G)$ (not incident to both $c$ and $s$), if $e \in T \cap T'$, then
   \[
   r_G([c - s], T) \setminus e = r_{G/e}([c - s], T \setminus e). \]
   ![See Clip 6]

2. For any $e \in E(G)$, if $e \notin T \cup T'$, then
   \[
   r_G([c - s], T) = r_{G\setminus e}([c - s], T). \]
   ![See Clip 7]

3. For any $e \in E(G)$, if there is a cut vertex $x$ such that all paths from $e$ to $c$ or $s$ pass through $x$, then
   \[
   e \in T \iff e \in T'. \]
   ![See Clip 8]
Definition

A sandpile torsor algorithm is *consistent* if it satisfies the 3 properties on the previous slide.
A sandpile torsor algorithm is **consistent** if it satisfies the 3 properties on the previous slide.

**Theorem (Ganguly-M., 2022+)**

Every consistent sandpile torsor algorithm has the same structure as rotor-routing (i.e. it is unique up to two $\mathbb{Z}_2$ actions).
Definition

A sandpile torsor algorithm is consistent if it satisfies the 3 properties on the previous slide.

Theorem (Ganguly-M., 2022+)

Every consistent sandpile torsor algorithm has the same structure as rotor-routing (i.e. it is unique up to two $\mathbb{Z}_2$ actions).

- To prove this, we first prove that it suffices to consider a subset of situations where rotor-routing takes just one step.
- We then use induction to reduce to 4 special cases.
- Resolving these cases requires a variety of methods and a great deal of work.
In 2017, Backman, Baker, and Yuen showed how to generalize the Bernardi action to *regular matroids*.

Instead of a ribbon structure, they require *acyclic circuit and cocircuit signatures*.

The definitions of consistency and sandpile torsor structure generalize naturally to regular matroids.

**Conjecture**

- The Backman-Baker-Yuen algorithm is consistent.
- All consistent sandpile torsor algorithms on regular matroids have the same structure.
Thanks for Listening!
Sources

- Spencer Backman, Matthew Baker, and Chi Ho Yuen. Geometric bijections for regular matroids, zonotopes, and Ehrhart theory. In *Forum of Mathematics, Sigma*, volume 7. Cambridge University Press, 2019.

- Norman L Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9(1):25–45, 1999.

- Matthew Baker and Yao Wang. The Bernardi process and torsor structures on spanning trees. *International Mathematics Research Notices*, 2017.

- Melody Chan, Thomas Church, and Joshua A Grochow. Rotor-routing and spanning trees on planar graphs. *International Mathematics Research Notices*, 2015(11), 2014.

- Scott Corry and David Perkinson. *Divisors and sandpiles*, volume 114. American Mathematical Soc., 2018.
Sources II

Jordan Ellenberg.
What is the sandpile torsor?
MathOverflow, 2011.

Emeric Gioan.
Circuit-cocircuit reversing systems in regular matroids.
*Annals of Combinatorics*, 12:171–182, 2008.

Ankan Ganguly and Alex McDonough.
Rotor-routing induces the only consistent sandpile torsor structure on plane graphs.
*arXiv preprint arXiv:2203.15079*, 2022.

Alexander E Holroyd, Lionel Levine, Karola Mészáros, Yuval Peres, James Propp, and David B Wilson.
Chip-firing and rotor-routing on directed graphs.
In *In and Out of Equilibrium 2*, pages 331–364. Springer, 2008.

Caroline Klivans.
*The Mathematics of Chip firing*.
Chapman & Hall, 2018.
Tamás Kálmán, Seunghun Lee, and Lilla Tóthmérész.  
The sandpile group of a trinity and a canonical definition for the planar bernardi action, 2019.

Itamar Landau and Lionel Levine.  
The rotor–router model on regular trees.  
*Journal of Combinatorial Theory, Series A, 116(2):421–433, 2009.*

Chi Ho Yuen.  
*Geometric Bijections of Graphs and Regular Matroids.*  
PhD thesis, Georgia Tech, 2018.