Identity Configurations of the Sandpile Group

William Chen
Illinois Mathematics and Science Academy
1500 W. Sullivan Road
Aurora, IL, 60506-1000, USA
email: chenwb@imsa.edu

under the direction of
Travis Schedler*

Submitted to Intel Science Talent Search

November 15, 2005

*Dept. of Mathematics, University of Chicago. *email:* trasched@math.uchicago.edu

Abstract

The abelian sandpile model on a connected graph yields a finite abelian group $\mathcal{G}$ of recurrent configurations which is closely related to the combinatorial Laplacian. We consider the identity configuration of the sandpile group on graphs with large edge multiplicities, called "thick" graphs. We explicitly compute the identity configuration for all thick paths using a recursion formula. We then analyze the thick cycle and explicitly compute the identity configuration for the three-cycle, the four-cycle, and certain types of symmetric cycles. The latter is a special case of a more general symmetry theorem we prove that applies to an arbitrary graph.

1 Introduction

The sandpile model was introduced by Bak, Tang, and Wiesenfeld [1] as an example of self-organized criticality and has been extensively studied in the context of statistical mechanics. It is the simplest and best-understood model of the phenomenon of self-organized criticality, which has been used to describe natural systems including earthquakes, forest fires, turbulent fluids, and punctutated equilibrium [2]. It also has applications for modeling fault tolerance and routing in internet computing [3]. The sandpile model is a variant of the chip-firing game, introduced independently by Björner, Lovász, and Shor [4]. Other similar models include the dollar game [5, 6], the oil game [6], and the Dirichlet game [3].

The structure of the sandpile model yields a finite abelian group, first studied by Dhar [7]. The identity configuration of this group is particularly interesting. For example, Creutz's paper [8] displays the complicated fractal patterns of the identity configuration on a square grid. This grid identity was further studied in [10, 11].

This paper presents several results on the identity configurations of graphs with arbitrary edge multiplicities, called "thick" graphs. In Section 2, we introduce the sandpile model and review preliminary results. In Section 3, we consider the problem of the identity configuration of the thick path on n vertices. In Section 4, we consider the related problem of the thick cycle.

2 The sandpile group

In this section we formally introduce the basic theory of the sandpile model and review important definitions and notation.

2.1 *The toppling rule*

We define the sandpile model on an finite connected graph $A = (V, E)$ with $|V| = n$. The graph A is called the *ambient space*. For this paper, we restrict ambient spaces to be undirected, but sandpiles may also be considered on directed graphs [12]. The ambient space may have multiple edges, but no loops. Recall that the *edge multiplicity* $e_{i,j}$ denotes the number of edges with endpoints i and

j. A distinguished vertex numbered n is designated as the *sink* and usually denoted by s; all other vertices are called *ordinary* and form a set $V_0 = V \backslash s$. Every vertex $i \in V_0$ is assigned a certain nonnegative number of grains of sand, called the *height*. This mapping of $V_0 \to \mathbb{N}^{n-1}$ is called a *configuration* and denoted by a vector $\boldsymbol{u} = (u_1, u_2, \ldots, u_{n-1})$. If $u_i < \deg(i)$ for $i \in V_0$, where $\deg(i)$ is the degree of i, then the vertex i is said to be *stable*. We also call a configuration *stable* if all of its vertices are stable.

If for any vertex i the inequality $u_i \geq \deg(i)$ holds, then i is unstable and *topples*. When vertex i topples, it passes one grain along every edge $\{i, j\}$ to j. That is, vertex i loses $\sum_{k=1}^{n} e_{i,k}$ grains and its adjacent vertices j gain $e_{i,j}$ grains each. In a *reverse toppling* of i, vertex i gains $\sum_{k=1}^{n} e_{i,k}$ grains and each adjacent vertex j loses $e_{i,j}$ grains. The sink cannot topple (unless noted otherwise) and we ignore the number of grains it holds. Since the sink collects grains, any sequence of topplings is of finite length.

Given a configuration $\boldsymbol{u}$, we now define $\sigma(\boldsymbol{u})$ as the stable configuration reached from $\boldsymbol{u}$ by a sequence of topplings and call it the *stabilization of* $\boldsymbol{u}$. The function σ is well defined by a confluence property, namely that the number of times each vertex is toppled is independent of the sequence in which the topplings are performed in $\boldsymbol{u}$ [3]. We can define a binary operation $\oplus$ on the set of stable configurations $\mathcal{M}$ by $\boldsymbol{u} \oplus \boldsymbol{v} = \sigma(\boldsymbol{u} + \boldsymbol{v})$, where addition is taken componentwise? pointwise. This gives $\mathcal{M}$ a monoid structure, which we will henceforth call the "sandpile monoid." The identity of $\mathcal{M}$ is the empty configuration $\mathbf{0}$.

We define a matrix corresponding to the toppling of vertices. The *combinatorial Laplacian* $L = (L_{ij})_{i,j \in V}$ is the $n \times n$ matrix defined as

$$L_{ij} = \begin{cases} \deg(i) & \text{if } i = j, \\ -e_{i,j} & \text{if } i \neq j, \end{cases} \tag{1}$$

The *toppling matrix* $\Delta = (\Delta_{ij})$ is the $(n-1) \times (n-1)$ matrix obtained from the Laplacian by deleting row n and column n, those corresponding to the sink. We let $\boldsymbol{\delta}_i$ denote the row vectors of Δ. A toppling of vertex i can be represented by $-\boldsymbol{\delta}_i$ and a reverse toppling by $\boldsymbol{\delta}_i$. For example, the configuration obtained upon toppling the vertex i in configuration $\boldsymbol{u}$ is $\boldsymbol{u} - \boldsymbol{\delta}_i$. The vectors $\boldsymbol{\delta}_i$

span a lattice,

$$\Lambda = \sum_{i=1}^{n-1} (\mathbb{Z}\delta_i). \tag{2}$$

The *score vector* of $\boldsymbol{u}$ is written as $\boldsymbol{\tau u} = ((\tau\boldsymbol{u})_1, (\tau\boldsymbol{u})_2, \ldots, (\tau\boldsymbol{u})_{n-1})$, where $(\tau\boldsymbol{u})_i$ denotes the number of times vertex i topples in the stabilization of $\boldsymbol{u}$. We define a map $\gamma : \mathbb{Z}^{n-1} \to \mathbb{Z}^{n-1}$ that takes a score vector $\boldsymbol{\tau}$ as input, given by:

$$\gamma(\boldsymbol{\tau}) = \left(\tau_1 \sum_{j=1}^{n-1}(-\Delta_{1,j}), \tau_2 \sum_{j=1}^{n-1}(-\Delta_{2,j}), \ldots, \tau_{n-1} \sum_{j=1}^{n-1}(-\Delta_{n-1,j}) \right). \tag{3}$$

Thus for a configuration $\boldsymbol{u}$, we have $\sigma(\boldsymbol{u}) = \boldsymbol{u} + \gamma(\boldsymbol{\tau u})$.

2.2 *Recurrent configurations and the burning algorithm*

Certain recurrent configurations appear with nonzero probability as grains are randomly added to an ambient space A. We state this concept more formally.

Definition 2.1. A stable configuration $\boldsymbol{w}$ is called *recurrent* if for every stable configuration $\boldsymbol{u}$ there exists a configuration $\boldsymbol{v}$ such that $\boldsymbol{w} = \boldsymbol{u} \oplus \boldsymbol{v}$.

The set of these recurrent configurations is the minimal nonempty ideal of the sandpile monoid $\mathcal{M}$. This ideal is a finite abelian group $\mathcal{G} \cong \mathbb{Z}^{n-1}/\Lambda$ of order $\det(\Delta)$ called the *sandpile group.* This group is closely related to the combinatorial Laplacian; the torsion coefficients of the canonical cyclic group decomposition of $\mathcal{G}$ are equal to the diagonal entries of the Smith normal form of the Laplacian [9]. The following lemma illustrates the ubiquitous structure of $\mathcal{G}$:

Lemma 2.2. *Let $\mathcal{M}^-$ denote the monoid reduction $\langle \mathcal{M} \backslash \{\mathbf{0}\} \rangle$. If $A_0 = (V_0, E)$ is connected, there exists some $k \in \mathbb{N}$ such that $\sigma(k\boldsymbol{u}) \in \mathcal{G}$ for every $\boldsymbol{u} \in \mathcal{M}$.*

Proof. Let $\boldsymbol{t}_i$ be the standard basis vector $(0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{Z}^{n-1}$ corresponding to i.

We show that there exists $k' \in \mathbb{N}$ such that every vertex $j \in V_0$ has height at least 1 after some sequence of topplings of $k'\boldsymbol{t}_i$, using induction on the diameter $D(A)$ of A. If $D(A) = 1$, choose $k' = \deg(i) + 1$. Upon toppling i, each vertex will have height at least 1. If $D(A) > 1$, there exists

$k'' \in \mathbb{N}$ such that every vertex of distance less than $D(A) - 1$ has height at least 1 after some sequence of topplings of $k''\boldsymbol{t}_i$ by the induction hypothesis. Then set $k' = k''(\max_{j\in V_0} \deg(j) + 1)$. Every vertex of distance $D(A) - 1$ from i can topple once and the inductive step is complete.

Set $k = k'(\max_{j\in V_0} \deg(j) + 1)$. Then $\sigma(k\boldsymbol{t}_i) \in \mathcal{G}$ because there is a sequence of topplings of $k\boldsymbol{t}_i$ yielding a configuration $\boldsymbol{C}$ in which every vertex is unstable. Configuration $\boldsymbol{C}$ can thus be obtained from any stable configuration by adding some positive number of chips at every vertex. Therefore $\sigma(\boldsymbol{C}) = \sigma(k\boldsymbol{t}_i)$ is recurrent. Since $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_{n-1}$ generate $\mathcal{M}^-$, for any stable configuration $\boldsymbol{u} \notin \mathcal{G}$ there exists some constant $k \in \mathbb{N}$ such that $\sigma(k\boldsymbol{u}) \in \mathcal{G}$. □

There are a number of other characterizations of recurrence. In fact, Dhar [7] showed that a configuration $\boldsymbol{u}$ is recurrent if and only if there does not exist $S \subseteq V_0$ for which all vertices $i \in S$ have the property $u_i < \deg_S(i)$, where $\deg_S(i)$ denotes $\sum_{j\in V} e_{i,j}$, the degree of i in the subgraph of A induced by vertex set S.

From this characterization, Majumdar and Dhar [13] gave a polynomial-time algorithm to recognize recurrence in an undirected graph, called the *burning algorithm.* Given a stable configuration $\boldsymbol{u}$, "burn" any vertex i for which $u_i \geq \deg_S(i)$, where S is the subset of all unburnt ordinary vertices in V_0. The configuration $\boldsymbol{u}$ is recurrent if and only if $S = \{\emptyset\}$ eventually (all sites are burnt). Equivalently, add $\boldsymbol{u}$ to the configuration

$$\boldsymbol{\beta} = \boldsymbol{\delta}_1 + \boldsymbol{\delta}_2 + \cdots + \boldsymbol{\delta}_{n-1} \tag{4}$$

corresponding to toppling the sink vertex (imagine an infinite supply of grains at the sink). Every ordinary vertex will topple at most once. The configuration $\boldsymbol{u}$ is recurrent if and only if every vertex topples exactly once. In fact, the final state after adding $\boldsymbol{\beta}$ will be identical to the original by (4). A formal statement of both versions of the burning algorithm follows:

Algorithm 2.3 (Burning algorithm version 1). Given a configuration $\boldsymbol{u}$,

1. Set $S := V_0$.
2. If there exists a vertex $i \in V_0$ where $u_i \geq \deg_S(i)$, set $S := S\backslash i$.
3. Repeat 2. until there are no vertices $i \in V_0$ where $u_i \geq \deg_S(i)$. Then $\boldsymbol{u}$ is recurrent if and

only if $S = \{\emptyset\}$.

Algorithm 2.4 (Burning algorithm version 2). Given a configuration $\boldsymbol{u}$,

1. Take $\boldsymbol{u} \oplus \boldsymbol{\beta}$. We often refer to this step as "toppling the sink," since addition of $\boldsymbol{\beta}$ corresponds to toppling the sink vertex.

2. Then $\boldsymbol{u}$ is recurrent if and only if $\boldsymbol{u} \oplus \boldsymbol{\beta} = \boldsymbol{u}$.

We now define an equivalence relation on the set of configurations that offers a method to find recurrent configurations.

Definition 2.5. Let a *generalized configuration* be an element of $\mathbb{Z}^{n-1}$ which we interpret as a configuration allowing negative heights. For two generalized configurations $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}^{n-1}$, $\boldsymbol{u} \equiv \boldsymbol{v}$ if $\boldsymbol{u} - \boldsymbol{v} \in \Lambda$, i.e., $\boldsymbol{u}$ can be reached by $\boldsymbol{v}$ through a series of topplings or reverse topplings and vice versa.

Then we show that there exists exactly one stable recurrent configuration in each equivalence class. We will use the configuration $\boldsymbol{\beta}$, since $\boldsymbol{\beta} \equiv \mathbf{0}$ by (4).

Corollary 2.6 (of Lemma 2.2). *For any configuration $\boldsymbol{u} \in \mathcal{M}$ there exists $k \in \mathbb{N}$ such that $\boldsymbol{u} \oplus k\boldsymbol{\beta} \in \mathcal{G}$.*

Proof. Lemma 2.2 establishes the existence of a $k \in \mathbb{N}$ such that $k\boldsymbol{\beta} \in \mathcal{G}$. Then by Definition 2.1, $\boldsymbol{u} \oplus k\boldsymbol{\beta}$ is recurrent. □

Proposition 2.7. *For any generalized configuration $\boldsymbol{u} \in \mathbb{Z}^{n-1}$, there exists a unique stable recurrent configuration $\boldsymbol{v} \in \mathcal{G}$ such that $\boldsymbol{u} \equiv \boldsymbol{v}$.*

Proof. Existence is established by Corollary 2.6, since there exists $k \in \mathbb{N}$ such that $\boldsymbol{u} \oplus k\boldsymbol{\beta}$ is recurrent and $\boldsymbol{u} \oplus k\boldsymbol{\beta} \equiv \boldsymbol{u}$. Now assume there exist recurrent configurations $\boldsymbol{u}$ and $\boldsymbol{v}$ such that $\boldsymbol{u} - \boldsymbol{v} \in \Lambda$. But then $\boldsymbol{u} = \boldsymbol{v}$, so there is a unique stable configuration in every equivalence class. □

Let $\phi : \mathcal{M} \rightarrow \mathcal{G}$ denote the natural homomorphism mapping each element of $\mathcal{M}$ to its equivalent recurrent configuration. Then by Corollary 2.6 there exists a $k \in \mathbb{N}$ such that $k\boldsymbol{\beta} \oplus \boldsymbol{u}$ is recurrent. Since $k\boldsymbol{\beta} \oplus \boldsymbol{u} \equiv \boldsymbol{u}$, iterating additions of $\boldsymbol{\beta}$ will determine the image $\phi(\boldsymbol{u})$ by Algorithm 2.4. This gives the following algorithm for computing the identity configuration:

Algorithm 2.8 (Extended burning algorithm). Given a configuration $\boldsymbol{u}$, the image $\phi(\boldsymbol{u})$ can be determined by the following algorithm:

1. Set $k := 1$.

2. If $\sigma(\boldsymbol{u} + k\boldsymbol{\beta}) \neq \sigma(\boldsymbol{u} + (k-1)\boldsymbol{\beta})$, set $k := k+1$. As in Algorithm 2.4, this corresponds to toppling the sink vertex.

3. Repeat 2. until $\sigma(\boldsymbol{u} + k\boldsymbol{\beta}) = \sigma(\boldsymbol{u} + (k+1)\boldsymbol{\beta})$. Then $\sigma(\boldsymbol{u} + k\boldsymbol{\beta}) = \phi(\boldsymbol{u})$.

Moreover, note that the identity of $\mathcal{G}$ on the ambient space A (hereafter referred to as the *identity configuration* and denoted by $\boldsymbol{I}^A$) is the image of the empty configuration $\boldsymbol{0}$ under ϕ. Therefore Algorithm 2.8 provides a useful way to compute the identity configuration $\boldsymbol{I}^A$.

Another method to demonstrate that a given configuration $\boldsymbol{u}$ is the identity configuration is to check $\boldsymbol{u}$ for idempotence. Clearly if $\boldsymbol{u}$ is recurrent and $\boldsymbol{u} \oplus \boldsymbol{u} = \boldsymbol{u}$, then $\boldsymbol{u}$ is the identity configuration. We show that in fact the only nonzero idempotent of the sandpile monoid $\mathcal{M}$ is the identity configuration if the subgraph of A induced by V_0 is connected, which eliminates the need to test for recurrence in most cases.

Lemma 2.9. *Let $A_0 = (V_0, E)$ be the subgraph of A obtained after deletion of the sink. If A_0 is connected, then there is exactly one idempotent in $\mathcal{M}^-$, namely the identity of the sandpile group $\mathcal{G}$.*

Proof. The quotient group $\mathcal{M}^-/\mathcal{G}$ is nilpotent by Lemma 2.2. Since $\mathcal{G} \triangleleft \mathcal{M}$, a stable configuration $\boldsymbol{u} \notin \mathcal{G}$ cannot be idempotent. The only idempotent in $\mathcal{G}$ is the identity configuration, and the result follows. □

3 Identity configuration of the thick path

We are interested in the identity configuration of the general thick path. Let $P_{n,\boldsymbol{e}}$ denote the thick path on n vertices with a sink on one end and edge multiplicities $\boldsymbol{e} = \{e_{1,2}, e_{2,3}, \ldots, e_{n-1,s}\}$. For simplicity we will usually denote $P_{n,\boldsymbol{e}}$ as simply P_n; the set of edge multiplicities used should be clear from the context.

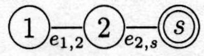

Figure 1: The thick path P_3.

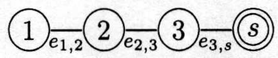

Figure 2: The thick path P_4.

First, we consider the undirected thick path P_3 (Figure 1) on three vertices with sink s and vertices labeled 1 and 2.

Claim 3.1. *The identity configuration $\boldsymbol{I}^{P_3}$ of P_3 is $\boldsymbol{u}^{P_3} = (u_1^{P_3}, u_2^{P_3})$, where $u_1^{P_3} = 0$ and $u_2^{P_3} = \left\lfloor \frac{e_{1,2}+e_{2,s}-1}{e_{2,s}} \right\rfloor e_{2,s}$, i.e., the largest multiple of $e_{2,s}$ less than $e_{1,2} + e_{2,s}$.*

Proof. We show that $\boldsymbol{u}^{P_3}$ is idempotent. Considering

$$\boldsymbol{u}^{P_3} + \boldsymbol{u}^{P_3} = \sigma\left(\left(0, 2\left\lfloor \frac{e_{1,2}+e_{2,s}-1}{e_{2,s}} \right\rfloor e_{2,s}\right)\right), \tag{5}$$

we observe that grains transferred to vertex 1 can be transferred back to vertex 2. Clearly, the value of $u_1^{P_3}$ will remain 0. The value of $u_2^{P_3}$ decreases by $e_{1,2} + e_{2,s}$ every time vertex 2 is toppled, but grains sent to vertex 1 are restored. This means we can say that the value of $u_2^{P_3}$ decreases by a net amount of $e_{2,s}$ every time vertex 2 is toppled. Eventually, we have

$$\boldsymbol{u}^{P_3} + \boldsymbol{u}^{P_3} = \left(0, \left\lfloor \frac{e_{1,2}+e_{2,s}-1}{e_{2,s}} \right\rfloor e_{2,s}\right) = \boldsymbol{u}^{P_3}. \tag{6}$$

Since $\boldsymbol{u}^{P_3} \neq \mathbf{0}$, Lemma 2.9 completes the proof. □

Next, we consider the thick path P_4 on four vertices, as shown in Figure 2. Notice the similarities in the formula of $\boldsymbol{I}^{P_4}$ to that of $\boldsymbol{I}^{P_3}$.

Claim 3.2. *The identity configuration $\boldsymbol{I}^{P_4}$ of P_4 is $\boldsymbol{u}^{P_4} = (u_1^{P_4}, u_2^{P_4}, u_3^{P_4})$, where $u_1^{P_4} = 0$, $u_2^{P_4} = \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor e_{2,3}$, and $u_3^{P_4} = \lambda e_{3,s} - u_2^{P_4}$, where λ is the largest integer such that $\lambda e_{3,s} - u_2^{P_4} < e_{2,3} + e_{3,s}$, i.e., $\lambda = \left\lfloor \frac{e_{2,3}+e_{3,s}+u_2^{P_4}-1}{e_{3,s}} \right\rfloor$.*

Proof. We show idempotence by considering

$$\boldsymbol{u}^{P_4} + \boldsymbol{u}^{P_4} = \sigma((0, 2u_2^{P_4}, 2u_3^{P_4})). \tag{7}$$

Vertex 1 will transfer any grains it receives in the stabilization process back to vertex 2 as in the proof of 3.1. Additionally, the number of grains on vertex 2 will decrease by $e_{2,3}$ every time it is toppled. While vertex 2 is stabilizing, vertex 3 receives a net total of $u_2^{P_4}$ grains. The grains on vertex 3 are lost $e_{3,s}$ at a time, which accounts for our choice of $u_3^{P_4}$. Then Lemma 2.9 completes the proof. □

Now referring back to the two previous graphs P_3 and P_4, we construct a generalization for identity configuration $\boldsymbol{I}^{P_n}$ of the thick path on n vertices (including the sink). Inductively, we have the following recursion lemma:

Lemma 3.3. *Given two thick paths P_n and P_{n+1}, where $e_{i,j}$ in P_n is equal to $e_{i,j}$ in P_{n+1}:*

$$I_i^{P_n} = I_i^{P_{n+1}} \tag{8}$$

for i ranging from 1 to $n-1$.

Proof. We can think of the vertex n in the thick path on $n+1$ vertices as the sink in the thick path on n vertices, collecting grains. As in the implementation of the Algorithm 2.4, any toppling of vertex n will leave the vertices to the left unchanged after toppling. □

Now we find the identity configuration of the general thick path on n vertices.

Theorem 3.4. *The identity configuration $\boldsymbol{I}^{P_n}$ of the thick path on n vertices is $\boldsymbol{u}^{P_n}$, where*

$$u_k^{P_n} = \begin{cases} 0 & \text{if } k = 1, \\ \lambda_k e_{k,k+1} - \sum_{i=1}^{k-1} u_i^{P_n} & \text{if } k > 1, \end{cases} \tag{9}$$

where λ_k is the largest integer such that $\lambda_k e_{k,k+1} - \sum_{i=1}^{k-1} u_i^{P_n} < e_{k-1,k} + e_{k,k+1}$.

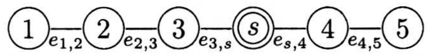

Figure 3: Another thick path on six vertices.

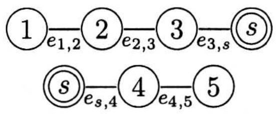

Figure 4: This graph decomposed into P_4 and P_3.

Proof. Upon stabilizing $\boldsymbol{u}^{P_n} + \boldsymbol{u}^{P_n}$, vertex $n-1$ eventually collects a total of $\sum_{i=1}^{n-2} u_i^{P_n}$ grains as the configuration stabilizes. Then, since vertex $n-1$ loses $e_{n-1,s}$ grains every time it topples, $u_{n-1}^{P_n}$ must equal the largest $\lambda_{n-1}e_{n-1,s} - \sum_{i=1}^{n-2} u_i^{P_n}$ less than $e_{n-2,n-1} + e_{n-1,s}$. Using Claim 3.1 for the initial case $k = 1$ and Lemma 3.3 to establish the recursion, we have the desired generalization. □

Now we address the question of the thick path on n vertices where the sink is not an end vertex. An example is shown in Figure 3. Note that such a path can be formed by taking the disjoint union of two paths where the sink is an end vertex and identifying their sinks, as in Figure 4. Idempotence is clearly not affected.

Remark 3.5. Using the same principle, we can also find the identity configuration of the "spider" graph formed by the disjoint union of several paths where the sinks are identified. More generally, for any arbitrary graph, the identity configuration can be computed separately on each connected component of $A_0 = (V_0, E)$.

By investigating the thick path, we characterized its identity configuration in Theorem 3.4. This result also extends to the identity configurations of several families of graphs with arbitrary edge multiplicities, including the spider graph.

4 Identity configuration of the thick cycle

We now consider the identity configuration of the thick cycle. Unless noted otherwise, we number the vertices in order, starting from a vertex distance 1 from the sink, such that the sink vertex is numbered n. Let $C_{n,\boldsymbol{e}}$ denote the thick cycle on n vertices with edge multiplicities

$e = \{e_{1,2}, e_{2,3}, \ldots, e_{n-1,n}, e_{n,1}\}$. As with the thick path we will usually write $P_{n,e}$ as simply P_n; the set e of edge multiplicities used should be clear from the context.

4.1 *Symmetric cycles*

We study the identity configuration of the symmetric cycle using Theorem 3.4. First, we prove a general result on joining several identical graphs.

Definition 4.1. Let $G = (V, E)$ be an ambient space with $G_1, G_2, \ldots, G_k$ as identical copies. Given a set $S_0 \subseteq V_0$, let $S = S_0 \cup \{\text{sink}\}$. Then $G^k(S)$ denotes the graph formed by taking the disjoint union $\bigsqcup_{i=1}^{k} G_i$ and, for each vertex $j \in S$, identifying all k copies of j as a single vertex (thereby joining edges). In particular, we are interested in $G^2(S)$, the *double* of G.

Theorem 4.2. *The identity configuration $\boldsymbol{I}^{G^k(S)}$ of $G^k(S)$ is $\boldsymbol{u}^{G^k(S)}$, where*

$$u_i^{G^k(S)} = \begin{cases} kI_i^G & \text{if } i \in S, \\ I_i^G & \text{otherwise .} \end{cases} \tag{10}$$

Proof. Let $\boldsymbol{\tau}$ denote the score vector of $\boldsymbol{I}^G + \boldsymbol{I}^G$. We take

$$\boldsymbol{u}^{G^k(S)} + \boldsymbol{u}^{G^k(S)} + \gamma(t_{v_1} + t_{v_2} + \cdots + t_{v_m}). \tag{11}$$

The height of each vertex i is then

$$u_i^{G^k(S)} + u_i^{G^k(S)} - \tau_i \deg(i) + \sum_{j \in V_0} \tau_i e_{i,j} = u_i^{G^k(S)}. \tag{12}$$

The resulting configuration is $\boldsymbol{u}^{G^k(S)}$ and thus $\boldsymbol{u}^{G^k(S)} \equiv \boldsymbol{0}$. We then show $\boldsymbol{u}^{G^k(S)}$ is recurrent using Algorithm 2.3. Since $\boldsymbol{I}^G$ is recurrent, there is a burning sequence of distinct vertices $v_1, v_2, \ldots, v_n$ for which all vertices are eventually burned. During each time step i we can burn all copies of vertex v_i in $G^k(S)$. Thus $\boldsymbol{u}^{G^k(S)}$ is recurrent and hence $\boldsymbol{u}^{G^k(S)} = \boldsymbol{I}^{G^k(S)}$. □

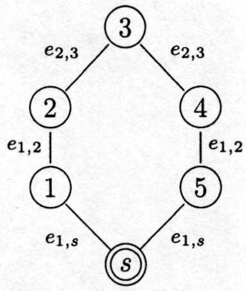

Figure 5: The symmetric hexagon C_6.

Corollary 4.3. *The identity configuration $\boldsymbol{I}^{G^2(S)}$ of the double graph $G^2(S)$ is $\mathbf{u}^{G^2(S)}$, where*

$$u_i^{G^2(S)} = \begin{cases} 2I_i^G & \text{if } i \in S, \\ I_i^G & \text{otherwise }. \end{cases} \tag{13}$$

Proof. This corollary is a special case of Theorem 4.2 where $k = 2$. ☐

Remark 4.4. Note that we can extend Theorem 4.2 to the disjoint union of graphs $G_1, G_2, \ldots, G_k$ where the score vector of $\boldsymbol{I}^{G_i} + \boldsymbol{I}^{G_i}$ is the same for all i, with some identification of the sets of vertices in each (also one can identify only some of the corresponding vertices, not only all of them or none).

We formally state a relationship between the identities of cycle and path graphs, namely that several paths can be joined together to form a cycle. From this relationship we derive a formula for the identity configuration of certain cycles.

Definition 4.5. A cycle C_n is *symmetric* if the edge multiplicities are equal between each pair of vertices of distance i and $i+1$ from the sink for all i from 0 to $\lfloor n/2 \rfloor$.

Remark 4.6. A symmetric thick cycle C_n is the double of a thick path $P_{n/2}$ if n is even. The symmetric C_6 is given as an example in Figure 5.

Theorem 4.7. *Let C_n be a symmetric thick cycle. If n is even, the identity configuration $\mathbf{u}^{C_n}$ is*

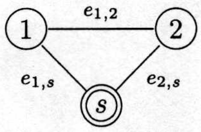

Figure 6: The thick three-cycle C_3.

given by

$$u_k^{C_n} = \begin{cases} 0 & \text{if } k = n/2, \\ \lambda_k e_{k,k-1} - \sum_{i=n/2}^{k} u_i^{C_n} & \text{if } k > n/2, \\ u_{n-k}^{C_n} & \text{if } k < n/2, \end{cases} \tag{14}$$

where λ_k *is the largest integer such that* $\lambda_k e_{k,k+1} - \sum_{i=1}^{k-1} u_i^{C_n} < e_{k-1,k} + e_{k,k+1}$.

If n *is odd, the identity configuration* $\boldsymbol{u}^{C_n}$ *is* $\boldsymbol{u}^{C_n}$

$$u_k^{C_n} = \begin{cases} \left\lfloor \frac{e_{k,k-1} + e_{k,k+1} - 1}{e_{k,k-1}} \right\rfloor e_{k,k-1} & \text{if } k = \lfloor n/2 \rfloor \text{ or } \lceil n/2 \rceil, \\ \lambda_k e_{k,k-1} - \sum_{i=\lceil n/2 \rceil}^{k} u_i^{C_n} & \text{if } k > \lceil n/2 \rceil, \\ u_{n-k}^{C_n} & \text{if } k < \lfloor n/2 \rfloor, \end{cases} \tag{15}$$

where λ_k *is the largest integer such that* $\lambda_k e_{k,k+1} - \sum_{i=1}^{k-1} u_i^{C_n} < e_{k-1,k} + e_{k,k+1}$.

Proof. We first prove the result for even n. Since C_n is symmetric, it can be expressed as the double of a thick path $P_{n/2}$. The result follows from Lemma 4.3 and Theorem 3.4.

Now if n is odd, we insert a vertex v between vertices $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$. By symmetry, if vertex $\lfloor n/2 \rfloor$ topples, vertex $\lceil n/2 \rceil$ also topples and vice versa. This means that vertex v does not affect the transfer of grains between vertices $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ and $u_v = 0$ for any stable configuration $\boldsymbol{u}$. Now our graph becomes an even cycle and the result follows from the proof for even n. □

4.2 *The general thick three-cycle and four-cycle*

We begin with the general thick three-cycle C_3 (Figure 6). Without loss of generality, $e_{1,s} \leq e_{2,s}$.

Claim 4.8. *The identity configuration $\boldsymbol{I}^{C_3}$ of the thick three-cycle C_3 is $\boldsymbol{u}^{C_3}$ where*

$$\boldsymbol{u}^{C_3} = \left(\left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor e_{2,s}, \left\lfloor \frac{e_{1,2} + e_{1,s} - 1}{e_{2,s}} \right\rfloor e_{1,s} \right). \tag{16}$$

Proof. We show recurrence by the Algorithm 2.4. "Toppling the sink" allows vertex 1 to immediately topple. Then vertex 2 can topple, since it received $e_{1,2} + e_{2,s}$ grains. Now we demonstrate idempotence by considering

$$\sigma\left(\left(2\left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor e_{2,s}, 2\left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor e_{1,s} \right)\right). \tag{17}$$

Toppling the vertices in succession decreases the number of grains on vertex 1 by $e_{1,s}$ and on vertex 2 by $e_{2,s}$, so $\boldsymbol{u}^{C_3} + \boldsymbol{u}^{C_3} = \boldsymbol{u}^{C_3}$ after $\left\lfloor \frac{e_{1,2}+e_{2,s}-1}{e_{2,s}} \right\rfloor$ topplings of each vertex. Lemma 2.9 completes the proof.

Alternatively, we can find this result using Algorithm 2.8. The configuration $\boldsymbol{u}^{C_3}$ is in the class of the identity since $\boldsymbol{u}^{C_3} = \left\lfloor \frac{e_{1,2}+e_{2,s}-1}{e_{2,s}} \right\rfloor \boldsymbol{\beta}$. Since $\boldsymbol{u}^{C_3}$ is recurrent, it must be the identity configuration. □

Remark 4.9. The thick path P_3 is a special case of C_3, where $e_{1,s} = 0$. This is reflected in the identity configuration of P_3, where

$$\boldsymbol{I}^{P_3} = \left(\left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor 0, \left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor e_{2,s} \right) \tag{18}$$

$$= \left(0, \left\lfloor \frac{e_{1,2} + e_{2,s} - 1}{e_{2,s}} \right\rfloor e_{2,s} \right), \tag{19}$$

as in Claim 3.1. In general, $\boldsymbol{I}^{P_n}$ is a special case of $\boldsymbol{I}^{C_n}$.

We now find a general formula for the identity of the thick four-cycle C_4 shown in Figure 7. We will make use of the score vector $\boldsymbol{\tau}$ of $\boldsymbol{I}^{C_4} + \boldsymbol{I}^{C_4}$. Notice that since $\boldsymbol{I}^G + \boldsymbol{I}^G + \gamma(\boldsymbol{\tau}) = \boldsymbol{I}^G$, we have

$$\boldsymbol{I}^G = -\gamma(\boldsymbol{\tau}). \tag{20}$$

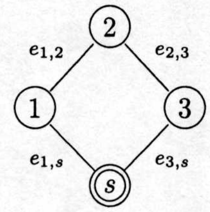

Figure 7: The thick four-cycle C_4.

For the square in particular, this means

$$\boldsymbol{I}^{C_4} = (\tau_1(e_{1,s} + e_{1,2}) - \tau_2 e_{1,2}, \tau_2(e_{1,2} + e_{2,3}) - \tau_1(e_{1,2}) - \tau_2(e_{2,3}), \tau_3(e_{2,3} + e_{3,s}) - \tau_2 e_{2,3}). \quad (21)$$

Lemma 4.10. *Let $\boldsymbol{\tau}$ denote the score vector of $\boldsymbol{I}^{C_4} + \boldsymbol{I}^{C_4}$. Then*

$$\tau_2 = \tau_1 \geq \tau_3 \text{ or } \tau_2 = \tau_3 \geq \tau_1. \quad (22)$$

Proof. Since I^{C_4} is stable, the following inequalities hold:

$$\tau_1 e_{1,s} + (\tau_1 - \tau_2) e_{1,2} < e_{1,s} + e_{1,2}, \quad (23)$$

$$(\tau_2 - \tau_1) e_{1,2} + (\tau_2 - \tau_3) e_{2,3} < e_{1,2} + e_{2,3}, \quad (24)$$

$$\tau_3 e_{3,s} + (\tau_3 - \tau_2) e_{2,3} < e_{2,3} + e_{3,s}. \quad (25)$$

If $\tau_1 > \tau_2$, then (23) is violated since $\tau_1 - \tau_2 \geq 1$. An analogous argument holds if $\tau_3 > \tau_2$, using (25). Therefore $\tau_2 \geq \tau_1$ and $\tau_2 \geq \tau_3$. If $\tau_2 > \tau_1$ and $\tau_2 > \tau_3$, then (24) is violated since $\tau_2 - \tau_1 \geq 1$ and $\tau_2 - \tau_3 \geq 1$. We conclude that $\tau_2 = \tau_1 \geq \tau_3$ or $\tau_2 = \tau_3 \geq \tau_1$. □

Theorem 4.11. *Let $\boldsymbol{\tau}$ denote the score vector of $\boldsymbol{I}^{C_4} + \boldsymbol{I}^{C_4}$. Then $\tau_1 = \tau_2 = \tau_3$ if and only if $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor = \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$. Moreover,*

$$\boldsymbol{\tau} = \left(\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor, \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor, \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor \right). \quad (26)$$

Proof. We first show that $\tau_1 = \tau_2 = \tau_3$ if $\left\lfloor \frac{e_{1,s}+e_{1,2}-1}{e_{1,s}} \right\rfloor = \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$. Using Algorithm 2.8, we set $k = \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor = \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$. We claim that $k\boldsymbol{\beta}$ is the identity configuration. Clearly $k\boldsymbol{\beta}$ is stable, since our choice of k guarantees that $ke_{1,s} < e_{1,s}+e_{1,2}$ and that $ke_{3,s} < e_{2,3}+e_{3,s}$. Adding $\boldsymbol{\beta}$ to $k\boldsymbol{\beta}$ allows vertices 1 and 3 to topple. Then vertex 2 receives $e_{1,2}+e_{2,3}$ and subsequently topples. Therefore $k\boldsymbol{\beta}$ is the identity configuration with $\boldsymbol{\tau} = (k, k, k)$, proving (26).

Conversely, assume $\tau_1 = \tau_2 = \tau_3 = k$. Applying (21), we have

$$\begin{aligned} \boldsymbol{I}^{C_4} &= -\gamma(\tau) && (27) \\ &= (ke_{1,s}, 0, ke_{3,s}) && (28) \end{aligned}$$

Then $\boldsymbol{I}^{C_4} = k\boldsymbol{\beta}$. In order for $\boldsymbol{I}^{C_4}$ to be both stable and recurrent,

$$k = \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor = \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor, \tag{29}$$

which completes the proof. □

Lemma 4.12. *The inequality* $\tau_1 = \tau_2 > \tau_3$ *holds if and only if* $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor > \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$. *Similarly,* $\tau_3 = \tau_2 > \tau_1$ *if and only if* $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor < \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$.

Proof. Consider the case where $\tau_1 = \tau_2 > \tau_3$. We compute the identity configuration using Algorithm 2.8 starting from the empty configuration $\mathbf{0}$. There exists $k \in \mathbb{Z}$ such that $k\boldsymbol{\beta}$ is the identity configuration. Let $\boldsymbol{\tau}_{k\boldsymbol{\beta}} = ((\tau_{k\boldsymbol{\beta}})_1, (\tau_{k\boldsymbol{\beta}})_2, (\tau_{k\boldsymbol{\beta}})_3)$ denote the score vector of $k\boldsymbol{\beta}$. Then $\tau_i = k-(\tau_{k\boldsymbol{\beta}})_i$ for any vertex $i \in V_0$. In order for $\tau_1 > \tau_3$ to hold, $\tau_1 = k-(\tau_{k\boldsymbol{\beta}})_1 > \tau_3 = k-(\tau_{k\boldsymbol{\beta}})_3$ or $(\tau_{k\boldsymbol{\beta}})_1 < (\tau_{k\boldsymbol{\beta}})_3$. This means that $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor > \left\lfloor \frac{e_{2,3}+e_{3,s}-1}{e_{3,s}} \right\rfloor$ since we want vertex 3 to topple first in the implementation of Algorithm 2.8. The proof is analogous for the case where $\tau_3 = \tau_2 > \tau_1$. □

At this point we work only in the case where $\tau_1 = \tau_2 > \tau_3$, noting by symmetry that the case where $\tau_3 = \tau_2 > \tau_1$ is essentially the same. We also divide this case into two subcases: either vertex 2 can be toppled second (after vertex 3) or vertex 2 cannot be toppled second in the implementation of Algorithm 2.8 starting from the empty configuration $\mathbf{0}$.

Lemma 4.13. *If* $\tau_1 = \tau_2 > \tau_3$*, then the following statements are equivalent:*

(i) Vertex 2 can be toppled second (after vertex 3).

(ii) The inequality $\left\lfloor \frac{\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{3,s}}{e_{2,3}+e_{3,s}} \right\rfloor > \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor$ *holds.*

(iii) Both $e_{1,2} \leq (\tau_2 - \tau_3)e_{2,3}$ *and* $e_{2,3} \leq \tau_3 e_{3,s} + (\tau_3 - \tau_2)e_{2,3}$ *hold.*

Proof. The left-hand side of statement (ii) can be interpreted as the the number of times of vertex 3 can be toppled after "toppling the sink" $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor$ times in the implementation of Algorithm 2.8 starting from **0**. The right-hand side can be interpreted as the maximum number of times vertex 3 can topple while keeping vertex 2 stable. After "toppling the sink" $\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor$ times using Algorithm 2.8, vertex 1 is still stable, but any further topplings of the sink would make vertex 1 unstable. Equivalence between statements (i) and (ii) follows.

We show that statement (iii) is equivalent to statement (i). Statement (i) is equivalent to the condition that "toppling the sink" (or adding a copy of $\boldsymbol{\beta}$) makes vertex 3 unstable, and upon toppling it, vertex 2 becomes unstable. We show this is equivalent to the two inequalities listed being satisfied.

The values $(\tau_2 - \tau_3)e_{2,3}$ and $\tau_3 e_{3,s} + (\tau_3 - \tau_2)e_{2,3}$ are $I_2^{C_4}$ and $I_3^{C_4}$, respectively. Since $\boldsymbol{I}^{C_4}$ is recurrent and vertex 3 can be toppled upon one toppling of the sink in Algorithm 2.4, we have

$$e_{2,3} + e_{3,s} \leq I_3^{C_4} + e_{3,s} \tag{30}$$

$$e_{2,3} \leq I_3^{C_4}. \tag{31}$$

Similarly, we require vertex 2 to topple upon the toppling of vertex 3. We have

$$e_{1,2} + e_{2,3} \leq I_2^{C_4} + e_{2,3} \tag{32}$$

$$e_{1,2} \leq I_2^{C_4} \tag{33}$$

Then by (21), the inequalities $e_{1,2} \leq (\tau_2 - \tau_3)e_{2,3}$ and $e_{2,3} \leq \tau_3 e_{3,s} + (\tau_3 - \tau_2)e_{2,3}$ hold. □

Theorem 4.14. *If the statements of Lemma 4.13 are satisfied, then*

$$\tau_1 = \tau_2 = \left\lceil \frac{\left(\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1\right) e_{2,3}}{e_{3,s}} \right\rceil + \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor \tag{34}$$

and

$$\tau_3 = \left\lceil \frac{\left(\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1\right) e_{2,3}}{e_{3,s}} \right\rceil. \tag{35}$$

Proof. In the computation of the identity from Algorithm 2.8, vertex 3 must topple $\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1$ times for vertex 2 to become unstable. Then vertex 2 topples exactly once and subsequently vertex 1 topples exactly once. From this we deduce that $\tau_1 - \tau_3 = \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor$. In order for the identity configuration to be stable and recurrent,

$$e_{2,3} \leq I_3^{C_4} < e_{2,3} + e_{3,s}. \tag{36}$$

Using (21), we have

$$e_{2,3} \leq \tau_3 e_{3,s} - \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor e_{2,3} < e_{2,3} + e_{3,s} \tag{37}$$

$$\left(\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1\right) e_{2,3} \leq \tau_3 e_{3,s} - \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor e_{2,3} < \left(\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1\right) e_{2,3} + e_{3,s} \tag{38}$$

$$\tau_3 = \left\lceil \frac{\left(\left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor + 1\right) e_{2,3}}{e_{3,s}} \right\rceil. \tag{39}$$

The formula for τ_1 follows from the equation $\tau_1 - \tau_3 = \left\lfloor \frac{e_{1,2}+e_{2,3}-1}{e_{2,3}} \right\rfloor$. □

We now deal with the other subcase if $\tau_1 = \tau_2 > \tau_3$, namely where vertex 2 cannot topple before vertex 1 in the implementation of Algorithm 2.8 starting from $\mathbf{0}$.

Theorem 4.15. *If the statements of Lemma 4.13 are not satisfied, then*

$$\tau_1 = \tau_2 = \left\lfloor \frac{e_{1,s}+e_{1,2}-1}{e_{1,s}} \right\rfloor \tag{40}$$

and

$$\tau_3 = \left\lceil \frac{\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{2,3}}{e_{2,3}+e_{3,s}} \right\rceil. \tag{41}$$

Proof. If the statements of Lemma 4.13 are not satisfied, then $e_{1,2} \leq I_1^{C_4} < e_{1,s} + e_{1,2}$ since vertex 1 must topple first upon "toppling the sink" using Algorithm 2.8. By (21), we have $e_{1,2} \leq \tau_1(e_{1,s} + e_{1,2}) - \tau_2 e_{1,2} = \tau_1 e_{1,s} < e_{1,s} + e_{1,2}$. Dividing all sides of the inequality by $e_{1,s}$, it follows that $\tau_1 = \left\lfloor \frac{e_{1,s}+e_{1,2}-1}{e_{1,s}} \right\rfloor$.

We also have the inequality $0 \leq I_3^{C_4} < e_{2,3} + e_{3,s}$. Using (21), we have

$$0 \leq t_3(e_{2,3}+e_{3,s}) - \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{2,3} < e_{2,3}+e_{3,s} \tag{42}$$

$$\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{2,3} \leq t_3(e_{2,3}+e_{3,s}) < \left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{2,3} + (e_{2,3}+e_{3,s}). \tag{43}$$

$$\tau_3 = \left\lceil \frac{\left\lfloor \frac{e_{1,2}+e_{1,s}-1}{e_{1,s}} \right\rfloor e_{2,3}}{e_{2,3}+e_{3,s}} \right\rceil \tag{44}$$

and the proof is complete. □

An explicit formula for the identity configuration $\boldsymbol{I}^{C_4}$ follows immediately from (21) by the results of Theorem 4.11, Theorem 4.14, and Theorem 4.15.

5 Conclusion

This work presents results on the identity configuration of the sandpile model and gives insight into its structure. The graphs considered are the path P_n and cycle C_n with arbitrary edge multiplicities. The results obtained in this work open numerous questions into the identity configuration. For example, we may be able to generalize the methods used in finding the identity configuration of C_4 to C_n for arbitrary $n \geq 4$. There is also room for further study using different ambient spaces.

The identity configuration of the rectangular grid mentioned in Section 1 has generated much interest, but relatively little is known about it. Dhar [9] observed two fascinating properties of this configuration. First, there is a square in the central area of the identity configuration on a $2n \times 2n$

grid where all vertices have height 2. Second, the identity configuration of the $(2n+1) \times (2n+1)$ grid is identical to that of the $2n \times 2n$ grid except for a central "cross." These conjectures remain unproven, but important steps have been taken towards proofs. Le Borgne and Rossin [10] prove the existence of a central rectangular region of height 2 in grids of certain dimensions, and Dartois and Magnien [11] analyze the computation of the grid identity using Algorithm 2.8 and offer direction for a proof of Dhar's cross observation.

Another interesting problem is the computation of the identity configuration of the sandpile group of the thick complete graph K_n with arbitrary edge multiplicities, since every graph on n vertices is a special case of K_n. Similarly, the directed ambient space A considered in [12] is more general than the undirected case.

6 Acknowledgments

This work was mainly conducted during the University of Chicago's Research Experience for Undergraduates (REU) program under László Babai. I am indebted to Fred Howard, Alexander Frankel, and Yvan Le Borgne for providing useful computer programs. I would also like to thank Michael Creutz, Clémence Magnien, Arnaud Dartois, and Timothy Credo for their correspondence. Most of all I am grateful to Travis Schedler and László Babai for proposing and directing this research.

References

[1] P. Bak, C. Tang, K. Wiesenfeld: "Self-organized criticality: an explanation for the $1/f$ noise." *Phys. Rev. Lett.* **59** (1987): 381.

[2] P. Bak: *How Nature Works: The Science of Self-Organized Criticality.* Oxford: Oxford Univ. Press, 1997.

[3] F. R. K. Chung, R. B. Ellis: "A chip-firing game and Dirichlet eigenvalues." *Discrete Math.* **257** (2002): 341-355.

[4] A. Björner, L. Lovász, P. W. Shor: "Chip-firing Games on Graphs." *Europ. J. Combinatorics* **12**. (1991): 283-291.

[5] N. L. Biggs: "Chip firing and the critical group of a graph." *J. Alg. Combin.* **9**: 25-45.

[6] J. van den Heuvel: "Algorithmic aspects of a chip-firing game." *Combinatorics, Probability, and Computing* **10** (2001): 505-529.

[7] D. Dhar, S. N. Majumdar: "Abelian sandpile model on the Bethe lattice." *J. Phys.* **A 23** (1990): 4333-4350.

[8] M. Creutz: "Abelian Sandpiles." *Computers in Physics* **5** (1991): 198-203.

[9] D. Dhar, P. Ruelle, S. Sen, D. Verma: "Algebraic aspects of Abelian sandpile models." *J. Phys.* **A 28** (1995): 805-831.

[10] Y. Le Borgne, D. Rossin: "On the identity of the sandpile group." *Discrete Math.* **256** (2002): 775-790.

[11] A. Dartois, C. O. Magnien: "Results and conjectures on the Sandpile Identity on a lattice." *Disc. Math. and Theoret. Comput. Sci.* **AB** (2004): 89-102.

[12] A. Björner, L. Lovász: "Chip-firing games on directed graphs." *J. Alg. Combinatorics* **1** (1992): 305-328.

[13] S. N. Majumdar, D. Dhar: "Equivalence between the Abelian sandpile model and the $q \to 0$ limit of the Potts model." *Physica* **A 185** (1992): 129-145.