Compatible Recurrent Identities of the Sandpile Group and Maximal Stable Configurations

Rupert Li

under the direction of

Yibo Gao
Department of Mathematics
Massachusetts Institute of Technology

November 18, 2019

Abstract

In the abelian sandpile model, recurrent chip configurations are of interest as they are a natural choice of coset representatives under the quotient of the reduced Laplacian. We investigate graphs whose recurrent identities with respect to different sinks are compatible with each other. The maximal stable configuration is the simplest recurrent chip configuration, and graphs whose recurrent identities equal the maximal stable configuration are of particular interest, and are said to have the complete maximal identity property. We prove that given any graph G one can attach trees to the vertices of G to yield a graph with the complete maximal identity property. We conclude with several intriguing conjectures about the complete maximal identity property of various graph products.

1 Introduction

The **abelian sandpile model** has been convergently discovered numerous times by researchers operating independently of each other, such as in [9] as an example of self-organized criticality, which was introduced in [1]. The automaton model was invented to model stacked items at sites, which would topple if a critical height was reached, and send the items to adjacent sites; for example, using a lattice graph to model a plane, the abelian sandpile model could mimic a pile of sand on a flat surface as it collapses under gravity to reach a certain stable state, and the destabilizing behavior of adding additional sand analyzed. The analysis of these models have led to the generalization of it for arbitrary locations and compositions of sites.

The abelian sandpile model, also referred to as the **chip-firing game**, on a directed graph G consists of a collection of chips at each vertex of G. If a vertex v has at least as many chips as its outdegree, then it can fire, sending one chip along each outgoing edge to its neighboring vertices. This continues until no vertex can fire. Articles on chip-firing in the literature include [2–5, 14, 15].

In Section 2, we establish the basic theory surrounding the chip-firing game, and define the primary algebraic object associated with the model, the **sandpile group**, and the special role **recurrent** elements play in the group; examples of articles on the sandpile group in the mathematical literature include [7], [12], and [16]. In Section 3, we begin stating our own results, investigating the graphs for which the simplest recurrent element, the **maximal stable configuration**, is the identity of the sandpile group, regardless of the choice of sink. The identity of the sandpile group is of particular interest, and has been previously investigated in [6] and [11] for $\mathbb{Z}^2$ lattice graphs. In Section 4, we find necessary conditions for the property developed in Section 3, the **complete maximal identity property**. In Section 5, we generalize the complete maximal identity property into the **complete identity property**, investigating graphs for which the recurrent identity remains the same regardless of the choice of sink. We consider the possible relationship between the complete identity property and bipartite graphs. We conclude in Section 6 with some conjectures surrounding the recurrent identity of sandpiles formed from graph products.

2 Preliminaries

We refer readers to [8] and [10] for detailed background on chip-firing. Unless otherwise stated, we restrict the chip-firing game to be on graphs which are simple, connected, and nontrivial (have more than one vertex). Recall that a simple graph is an unweighted, undirected graph without loops or multiple edges. A **sandpile** is a graph G that has a special vertex, called a **sink**. A **chip configuration** over the sandpile is a vector of integers indexed over all non-sink vertices of G. In standard convention, these numbers must be nonnegative integers, a discrete number of **chips**. A chip configuration c, if explicitly stated to be over all vertices, not just all non-sink vertices, may be used for convenience when the sink changes. The entry corresponding to the sink is simply excluded from the vector.

In a sandpile, a vertex can **fire** if it has at least as many chips as its (out)degree, at which point it sends chips along each edge to its neighboring vertices, with each edge transferring w, where w is the weight of the edge (in an undirected graph, each edge has weight 1); the vertex that fires loses the chips it fired. A vertex is said to be **active** if it can fire. The sink is not allowed to fire, nor are its chips considered; hence a chip configuration for a sandpile does not have an entry corresponding to the number of chips at the sink. A **stable** configuration is a configuration that has no active vertices. If by a sequence of firings a chip configuration c can result in a stable configuration, that stable configuration is called the **stabilization** and is denoted $\mathrm{Stab}(c)$. It has been proven in [10] that if a stabilization exists, then it is unique for each chip configuration, regardless of the order in which vertices are fired; moreover, regardless of the order in which vertices are fired, the chip configuration will eventually result in the stable configuration, always taking the same number of firings in total. It has also been shown in [10] that in a sandpile all chip configurations stabilize.

Definition 2.1. The **maximal stable configuration** m_G is the chip configuration in which every vertex v has $d_v - 1$ chips, where d_v is the degree of vertex v.

For a sandpile, a chip configuration c is called **accessible** if for all (stable) configurations d there exists a configuration e such that $\mathrm{Stab}(d+e) = c$, where $d+e$ is the componentwise addition of the vectors d and e.

Definition 2.2. A chip configuration is called **recurrent** if it is accessible and stable.

Note that m_G is always recurrent, even when there is only one recurrent configuration for a given sandpile, and is by far the simplest recurrent chip configuration.

Example 2.1. The cycle graph on four vertices C_4 may have its vertices indexed 0, 1, 2, and 3, where v_0 is the sink, as shown in Fig. 1, where the sink is colored green and the non-sink vertices colored blue.

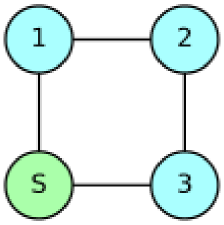
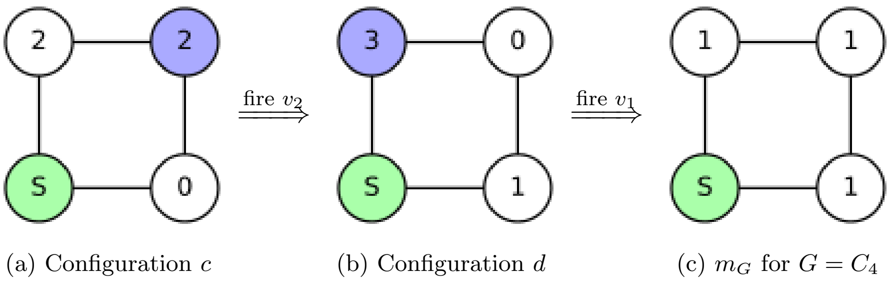


Figure 1: C_4 with sink v_0. The non-sink vertices are numbered 1, 2, and 3, as shown.

This sandpile may have a chip configuration $c = (2, 2, 0)$ as shown in Fig. 2a. In c, vertices v_1 and v_2 are active. If v_2 (colored purple in Fig. 2a) fires, the result is chip configuration $d = (3, 0, 1)$ as shown in Fig. 2b. In d, v_1 is the only active vertex (colored purple in Fig. 2b), and firing it results in the maximal stable configuration $m_G = (1, 1, 1)$ where in this example $G = C_4$ with sink at v_0, as shown in Fig. 2c.



(a) Configuration c (b) Configuration d (c) m_G for $G = C_4$

Figure 2: Chip-firing on C_4. The non-sink vertices are labeled with their number of chips.

If one labels the vertices of a sandpile G as $v_1, \dots, v_n$, where the graph G may be weighted with positive integer weights and directed, then the **Laplacian** of G is the $n \times n$ matrix $\Delta = D^{\mathrm{T}} - A^{\mathrm{T}}$, where D is the diagonal matrix where $D_{ii} = d_{v_i}$, and A is the adjacency matrix of G. That is, if a_{ij} is the weight of the edge from vertex v_i to v_j, and d_i is the out-degree of v_i,

$$\Delta_{ij} = \begin{cases} -a_{ij} & \text{for } i \neq j, \\ d_i & \text{for } i = j. \end{cases}$$

The **reduced Laplacian** Δ' of G is obtained by removing from Δ the row and column corresponding to the sink. To explicitly refer to the reduced Laplacian with the sink at v_i, the notation $\Delta^{(i)}$ is used. We can represent the firing of a non-sink vertex v as the subtraction of the column of Δ' corresponding to v from the chip configuration.

Example 2.2. The cycle graph on four vertices C_4 has Laplacian

$$\Delta = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

and the reduced Laplacian, for sink v_1, is

$$\Delta' = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

In order to view configurations before and after firing as equivalent, we define the **sandpile group** of G to be the group quotient

$$\mathcal{S}(G) = \mathbb{Z}^{n-1}/\Delta'\mathbb{Z}^{n-1}.$$

From the definition of the sandpile group, we see that $|\mathcal{S}(G)| = |\Delta'|$, where $|\mathcal{S}(G)|$ denotes the order of the sandpile group $\mathcal{S}(G)$ and $|\Delta'|$ denotes the determinant of the reduced Laplacian. This holds regardless of the choice of sink. By the Matrix-Tree theorem (see for example [13]), $|\Delta'|$ is the number of spanning trees of G. It is also shown in [10] that each equivalence class of $\mathcal{S}(G)$ contains exactly one recurrent configuration, and that the recurrent configurations form an abelian group with the operation being defined as the stabilization of the sum of two recurrent configurations.

A configuration c over graph G with n vertices is **equivalent** to another configuration d when they have the same image in $\mathcal{S}(G)$, meaning they lie in the same equivalence class in $\mathcal{S}(G)$, or in other words,

$$c \equiv d \Leftrightarrow \text{there exists } \mathbf{v} \in \mathbb{Z}^{n-1} : c - d = \Delta' \mathbf{v}.$$

Notice that as Δ' is non-singular by the Matrix-Tree theorem, the vector $(\Delta')^{-1}(c-d)$ is the unique solution to the equation $c = d + \Delta'\mathbf{v}$, and thus

$$c \equiv d \Longleftrightarrow (\Delta')^{-1}(c-d) \in \mathbb{Z}^{n-1}.$$

However, two configurations being equivalent does not necessarily imply there exists a firing sequence that takes one to the other. Hence, we introduce the ideas of backfiring and unrestricted firing. To fire a vertex v is equivalent to subtracting a column of Δ' corresponding to v, and similarly to **backfire** v is equivalent to *adding* that column. **Unrestricted firing** is when vertices are allowed to fire and/or backfire regardless of the number of chips they have. Furthermore, the sink is allowed to fire, where firing the sink corresponds to backfiring all non-sink vertices.

Of interest in the sandpile group is the identity element, which leads us to the definition of the recurrent identity.

Definition 2.3. The **recurrent identity** is the recurrent configuration equivalent to the all-zero configuration.

Recall that each equivalence class contains exactly one recurrent configuration, and thus the recurrent identity is well-defined.

Example 2.3. Let $G = K_3$. We look at the sandpile of G with arbitrary sink s. We will show that the recurrent identity c is equivalent to the all-zero configuration d, where our two chip configurations are

$$c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and the reduced Laplacian is

$$\Delta' = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

Now we notice that

$$(\Delta')^{-1}(c-d) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \in \mathbb{Z}^2.$$

Hence, $c \equiv d$, and d can be obtained from c by unrestrictedly firing (see Section 3 for a formal definition) the two non-sink vertices once each. See Fig. 3 for a visual example of the firing sequence taking c to d, and thus the equivalence of c and d.

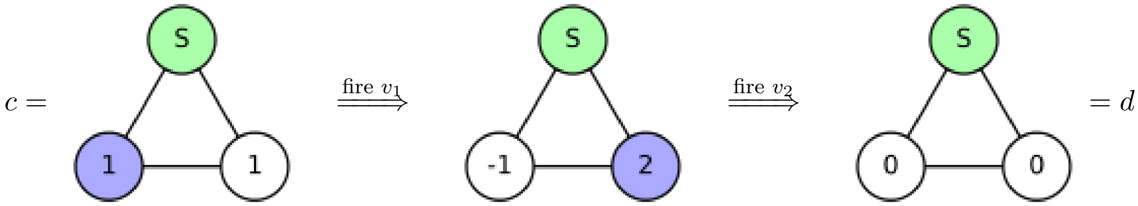
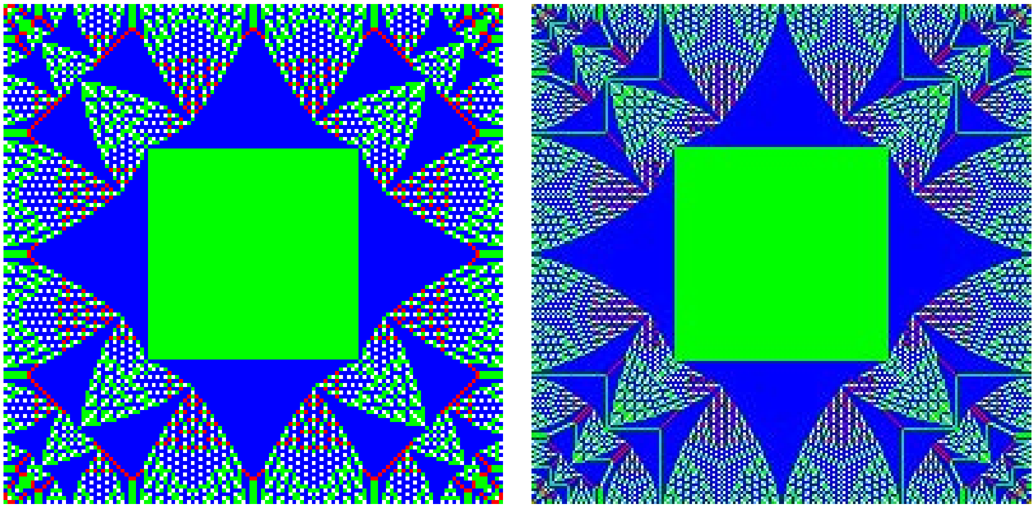


Figure 3: Equivalence of the recurrent identity and the all-zero configuration. The sink is colored green, and the vertex being fired is colored purple.

The recurrent identity has been studied previously, including some limiting behavior for $\mathbb{Z}^2$ lattice graphs as in [6] and [11]. For example, the recurrent identity for the 128×128 and 198×198 square grids with a boundary sink, or a sink connected to all the boundary vertices, is shown in Fig. 4. Many intriguing questions about the identity are still open.



(a) 128×128 square grid

(b) 198×198 square grid

Figure 4: The recurrent identity of the 128×128 and 198×198 square grids, as shown in [10]. 0, 1, 2, and 3 chips are displayed as orange, red, green, and blue, respectively.

3 The Complete Maximal Identity Property

The maximal stable configuration is guaranteed to be recurrent over all sandpiles, even when there is only one recurrent configuration, and is by far the simplest recurrent configuration. Furthermore, of interest among the recurrent elements is the identity of the abelian group, the recurrent identity. Hence, it is of interest to know for which graphs the maximal stable configuration is the recurrent identity, leading to our definition of the maximal identity property.

Definition 3.1. A graph G is said to have the **maximal identity property** at a vertex s if, having chosen s as the sink, the maximal stable configuration is the recurrent identity of the sandpile G.

The following lemma provides an equivalent condition for the maximal identity property.

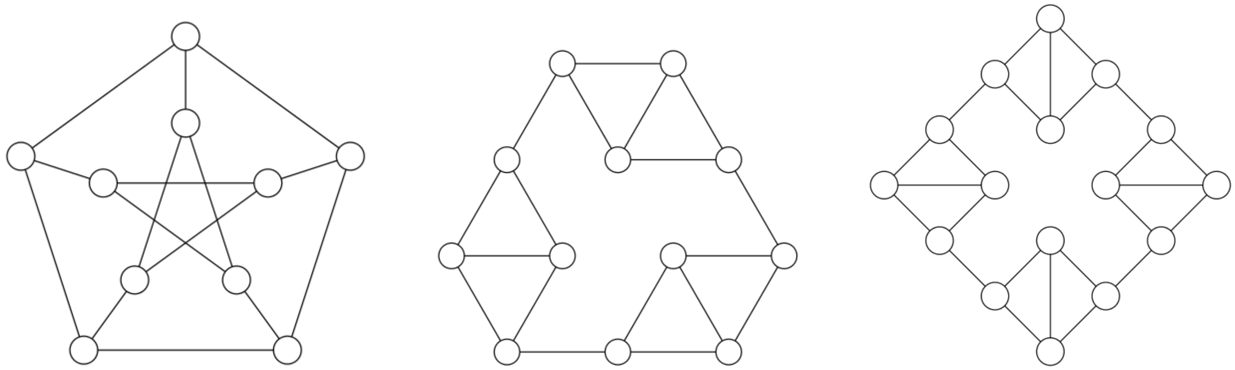
Lemma 3.1. *A necessary and sufficient condition for a sandpile G to have the maximal identity property is for there to exist an unrestricted firing sequence that takes m_G to $\mathbf{0}$.*

Proof. The maximal stable configuration m_G is always recurrent. An unrestricted firing sequence between m_G and $\mathbf{0}$ exists if and only if the two configurations are equivalent, which occurs if and only if m_G is the recurrent identity, resulting in the maximal identity property. ■

Another necessary and sufficient condition for a sandpile G to have the maximal identity property is for $\mathrm{Stab}(m_G + m_G) = m_G$.

Definition 3.2. A graph is said to have the **complete maximal identity property** if for all vertices v, it has the maximal identity property for sink v.

Example 3.1. Both the Petersen graph and the n-diamond ring for all positive integers n have the complete maximal identity property (see Fig. 5 for the Petersen graph and the n-diamond ring for $n = 3, 4$). The proofs are straightforward but tedious, and thus not included in this paper.



(a) The Petersen graph

(b) The 3-diamond ring

(c) The 4-diamond ring

Figure 5: The Petersen graph and the n-diamond ring for $n = 3, 4$

Proposition 3.2. *All complete graphs K_n, odd cycles C_{2n+1}, and trees have the complete maximal identity property.*

Proof. **Case 1:** K_n. By symmetry, we only need to prove that $m_{K_n} \equiv \mathbf{0}$ for one sink of K_n. We prove this via the unrestricted firing method of Lemma 3.1. The configuration m_{K_n} has $n-2$ chips at each non-sink vertex. Backfire the sink $n-2$ times to yield the all-zero configuration.

Case 2: C_{2n+1}. By symmetry, we only need to prove that $m_{C_{2n+1}} \equiv \mathbf{0}$ for one sink of C_{2n+1}. We prove this via the unrestricted firing method. Note that $m_{C_{2n+1}} \equiv \mathbf{1}$. Define the **rank** of a vertex to be the edge length of the shortest path from the sink to that vertex. Fire both vertices of rank n once. This results in a net movement on one chip to each of the vertices of rank $n-1$, ridding the vertices of rank n of chips. Now, as each vertex of rank $n-1$ has 2 chips, fire all vertices with rank at least $n-1$ twice to eliminate all chips from those vertices. Continue in this manner, firing all vertices with rank $n-k$ or higher $k+1$ times. This results in the all-zero configuration.

Case 3: Trees. There is only one spanning tree of a tree, the tree itself. Hence, we apply the Matrix Tree Theorem to find that there is only one equivalence class in the sandpile group, regardless of the choice of sink. Thus, there is only one recurrent element; as m_G is always recurrent, it is the only recurrent configuration. Therefore, m_G must be the recurrent identity. ■

The fact that a tree has a trivial sandpile group gives it a unique effect when added to other graphs. The concept of adding a tree of size n, where the size of a graph is its number of edges, to a vertex $v \in G$ is simply taking any tree T on $n+1$ vertices and n edges, then taking the disjoint union of T and G, and finally merging a vertex of T with vertex v. See Fig. 6 and Fig. 7 for an example on how trees are added to graphs. Notice that adding the tree of size 0, a single vertex, does not change the graph. The following theorem states how one can attach trees to any connected graph in the aforementioned manner to result in a graph that has the complete maximal identity property.

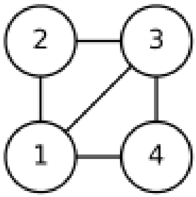


Figure 6: The Diamond Graph

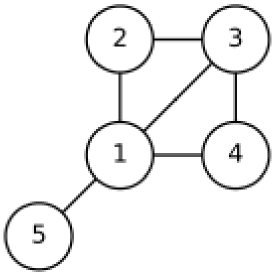


Figure 7: The Diamond Graph with a tree of size 1 added to vertex v_1

Theorem 3.3. *Given any connected graph G, there exists infinitely many graphs derived from adding trees to G that have the complete maximal identity property.*

To prove Theorem 3.3, we need the following lemma.

Lemma 3.4. *Let G be any graph. Let $\{T_v\}$ be a family of trees labeled by vertices of G and let G' be the graph obtained from G by adding each tree T_v to vertex v. Then G' has the complete maximal identity property if and only if the configuration c over G with $d_v + |T_v| - 2$ chips at all vertices v is equivalent to the identity for all selections of sinks, where $|T_v|$ is the number of vertices in the tree.*

Proof. We first prove the if direction. Pick a sink $s \in G \subseteq G'$. Let c be the chip configuration in the lemma statement. We observe that each tree T_v has size $c_v - (m_G)_v$. As trees have a trivial sandpile group, there exists an unrestricted firing sequence that moves all the chips of $m_{G'}$ in T_v to v. Each edge of T_v provides one additional chip to $m_{G'}$ from m_G, and thus once all the chips are moved towards their attachment point v in G, we find that the configuration that results is configuration $c \equiv m_{G'}$. As $c \equiv \mathbf{0_G}$, there exists an unrestricted firing sequence that takes c to $\mathbf{0}$ in G. Use this same sequence on the chip configuration c over G', except whenever vertex v in G is fired, fire vertex v and all the vertices of T_v in G'. This results in $\mathbf{0_{G'}}$, thus illustrating that G' has the maximal identity property for sink s. The choice of s was arbitrary, so it applies for all vertices in G.

However, G' also has the vertices in the trees that were added. We must prove that G' has the maximal identity property for these vertices as sinks. Pick a sink $s \in G'$ from tree T_v for arbitrary vertex $v \in G$. Use the unrestricted firing sequence that took $m_{G'}$ to $\mathbf{0}$ for sink v, to result in a configuration equivalent to m_G that has all of the chips at vertex v. As all trees have a trivial sandpile group, there exists a unrestricted firing sequence that takes all the chips at vertex v to s, if only T_v is considered. Now, whenever vertex v needs to be fired, fire all vertices in G, and all vertices of $T_{v'}$ for all $v' \neq v$. This allows all firings in $T_v \subset G'$ to work inside G'. This demonstrates that $m_{G'} \equiv \mathbf{0}$, and thus that G' has the complete maximal identity property.

To prove the only if direction, assume G' has the complete maximal identity property. Reversing the unrestricted firing sequences used to move all the chips of $m_{G'}$ in T_v to v, we find $c \equiv m_{G'}$ as the $|T_v| - 1$ chips T_v needs for the maximal stable configuration are taken away from v to result in v having $d_v - 1$ chips, the number of chips it needs for the maximal stable configuration. As G' has the complete maximal identity property, $c \equiv m_{G'} \equiv \mathbf{0_{G'}}$. ■

We now can prove Theorem 3.3.

Proof of Theorem 3.3. Say we have a graph G. By the Matrix-Tree Theorem, regardless of the choice of sink, $|\Delta'|$ is a constant integer. Say $|\Delta'| = k$. By Lemma 3.4, to prove the result it suffices to find infinitely many configurations c, where c is over all vertices, such that $c \geq m_G$ and for all selections of sinks, $c \equiv \mathbf{0}$. We will create such a configuration as follows. For each vertex v, let the number of chips on it be equal to a multiple of k that is greater than or equal to $d_v - 1$. This configuration can be represented by a vector $k\mathbf{x}$, where $\mathbf{x}$ is a vector composed of nonnegative integers. For a particular selection of sink, to prove that this configuration is equivalent to the identity, we must show that there exists a vector $\mathbf{y}$ with integer entries such that

$$\Delta'\mathbf{y} = k\mathbf{x}.$$

But we know that $|\Delta'| = k$, so using the fact that the adjoint matrix of an integer matrix (which Δ' is) has integer entries, we know that $k(\Delta')^{-1}$ has integer entries. Hence

$$\mathbf{y} = k(\Delta')^{-1}\mathbf{x},$$

and as $\mathbf{x}$ has integer entries, $\mathbf{y}$ has integer entries. As there are infinitely many multiples of a positive integer greater than a fixed value, there are infinitely many valid configurations c. Thus, there exists infinitely many graphs consisting of the given graph with trees attached to it that have the complete maximal identity property. ■

Because we may add trees to graphs to give them the complete maximal identity property, we wish to have a notion of irreducibility that eliminates such graphs which have trees added to them. This leads us to the classical notion of a biconnected graph.

Definition 3.3. A graph is **biconnected** if it remains connected even if one removes any single vertex and its incident edges.

In other words, for any two vertices in a biconnected graph, there exist at least two vertex-disjoint paths that connect them.

A search over all biconnected graphs with 11 or fewer vertices found that asides from odd cycles and complete graphs, there were only three and two biconnected graphs (up to isomorphism) with 8 and 10 vertices, respectively. These include the 2-diamond ring and the Petersen graph from Example 3.1. In addition, there are three other biconnected graphs with 12 vertices known to have the complete maximal identity property, including the 3-diamond ring; however, the search over all biconnected graphs with 12 vertices is too computationally intensive to complete.

The observed lack of any other biconnected graphs with an odd number of vertices that possess the complete maximal identity property prompts the following question.

Question 3.5. *Are the only biconnected graphs with an odd number of vertices that possess the complete maximal identity property cycle graphs and complete graphs?*

4 Necessary Conditions for the Complete Maximal Identity Property

In order to create necessary conditions for the complete maximal identity property, we first make the following definition.

Definition 4.1. A vector $c \in \mathbb{Z}^n$ is **compatible** if for any sink v_i we have

$$(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) \in \Delta^{(i)}\mathbb{Z}^{n-1}.$$

An equivalent definition is that for all sinks $s \in G$,

$$(\Delta^{(s)})^{-1}c' \in \mathbb{Z}^{|G|-1}.$$

Note that if a and b are compatible, then $a \pm b$ is compatible.

Lemma 4.1. *If c is compatible and $s = c_1 + \cdots + c_n$, then $s\mathbf{e}_i$ is compatible for all i.*

Proof. Notice that if $c \in \mathbb{Z}^n$ is in the image of Δ, then c is compatible. Since $(c_1, \ldots, c_n)$ is compatible, then

$$(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) = \sum_{j \in \{1,\ldots,n\} \setminus \{i\}} \alpha_j \Delta_j^{(i)},$$

where $\alpha_j \in \mathbb{Z}$ and $\Delta_j^{(i)}$ denotes the column of $\Delta^{(i)}$ corresponding to vertex v_j. Then

$$\left(c_1, \ldots, c_{i-1}, -\sum_{j \in \{1,\ldots,n\} \setminus \{i\}} c_j, c_{i+1}, \ldots, c_n \right) = \sum_{j \in \{1,\ldots,n\} \setminus \{i\}} \alpha_j \Delta_j$$

is in the image of Δ and thus is compatible. Subtracting the two compatible vectors, then $s\mathbf{e}_i$ is compatible for any i where $s = c_1 + \cdots + c_n$. ■

Clearly, $I = \{d \mid d\mathbf{e}_i \text{ is compatible for any } i\}$ forms an ideal in $\mathbb{Z}$. For a graph G with the complete maximal identity property, we know $k \in I$, where $k = |\Delta'|$, and

$$\sum_{v \in G} (\deg(v) - 1) = 2 \cdot \text{size}(G) - |G| \in I,$$

where size(G) denotes the number of edges of G, so their greatest common divisor is in I.

As $\mathbb{Z}$ is a principal ideal domain, for each graph G, $I = (x)$ for some positive integer x. Notice that x is the smallest positive element of I. This leads us to our definition of the minimal compatibility number of a graph.

Definition 4.2. The **minimal compatibility number** of a graph is the positive integer x such that $(x) = I := \{d \mid d\mathbf{e}_i \text{ is compatible for any } i\}$.

Lemma 4.2. *The minimal compatibility number of a graph G is* 1 *if and only if G is a tree.*

Proof. We first prove the only if direction. If the minimal compatibility number of a graph is 1, then every chip configuration is in the integer image of the reduced Laplacian and thus equivalent to each other. Hence, the sandpile group $\mathcal{S}(G)$ is the trivial group, so by the Matrix-Tree Theorem G must have only 1 spanning tree, or in other words, is a tree itself.

For the if direction, a tree has a trivial sandpile group and thus every chip configuration is equivalent to each other, meaning that each chip configuration is in the integer image of the reduced Laplacian, and thus the minimal compatibility number of the tree is 1. ■

Notice that $d\mathbf{e}_i$ is compatible if each least common denominator of the column of $(\Delta')^{-1}$ corresponding to vertex i for all sinks s divides d. Hence, $d \in I$ if the least common denominator of all entries of the inverses of all reduced Laplacians of a graph G divides d, and thus the minimal compatibility number of a graph is the least common denominator of all entries of the inverses of all reduced Laplacians of a graph G.

Proposition 4.3. *If a non-tree graph G has the complete maximal identity property, then*

$$\gcd\left(\left|\Delta'\right|, 2 \cdot \text{size}(G) - |G|\right) > 1.$$

Proof. We will prove the contrapositive.

If for a non-tree graph G,

$$\gcd\left(\left|\Delta'\right|, 2 \cdot \text{size}(G) - |G|\right) = 1,$$

the graph does not have the complete maximal identity property, as if it did then

$$\gcd\left(\left|\Delta'\right|, 2 \cdot \text{size}(G) - |G|\right) = 1 \in I,$$

so the minimal compatibility number of G is 1, which by Lemma 4.2 means G is a tree, yielding a contradiction. ■

Proposition 4.4. *If a graph G has the complete maximal identity property, then the minimal compatibility number x satisfies $x \leq |G|^2 - 2|G|$ for $|G| > 2$, and $x = 1$ if $|G| = 2$.*

Proof. If $|G| = 2$, then G must be the connected graph with two vertices, which is a tree and thus the result follows from Lemma 4.2.

We now assume $|G| > 2$. As

$$2 \cdot \text{size}(G) - |G| \in I,$$

we have

$$x \leq 2 \cdot \text{size}(G) - |G|,$$

as x is the smallest positive element of I and $2 \cdot \text{size}(G) - |G| > 0$. This is because G is a connected graph so $\text{size}(G) \geq |G| - 1$, and thus $2 \cdot \text{size}(G) - |G| \geq |G| - 2 > 0$.

The maximum value of $\text{size}(G)$ of a graph with fixed order is $\frac{|G|(|G|-1)}{2}$, so

$$x \leq 2 \cdot \text{size}(G) - |G| \leq \left(|G|^2 - |G|\right) - |G| = |G|^2 - 2|G|.$$

■

Proposition 4.5. *The minimal compatibility number for both K_n and C_n is n.*

Proof. We will show $n\mathbf{e}_i$ is compatible, with an irreducible firing vector (i.e. the elements have no common factor). The **firing vector** for a configuration c is the vector $(\Delta')^{-1}c$, where each element of the vector corresponds to how many times the corresponding non-sink vertex must fire from the all-zero configuration to reach c. Note that the firing vector may not necessarily have integer entries. As the reduced Laplacian is non-singular by the Matrix-Tree Theorem, the firing vector is well-defined, existing uniquely as the firing vector that takes the all-zero configuration to any configuration c. If $n\mathbf{e}_i$ is compatible, but the firing vector is irreducible, meaning all the entries are integers which do not share a nontrivial common factor, then $n'\mathbf{e}_i$ for all positive integers $n' < n$

will not have a firing vector that has all integer entries; this is because that firing vector can be obtained by multiplying the firing vector for $n\mathbf{e}_i$ by $\frac{n'}{n}$.

For the complete graph K_n, fire the sink and then backfire v_i to result in n chips at v_i. The firing vector consists of -1 at all non-sink vertices except v_i which is -2. Hence, this is an irreducible firing vector, and $x = n$.

For the cycle graph C_n, we will first show that $n\mathbf{e}_i$ is compatible for all i. Notice that on the cycle graph, one may select a proper connected subgraph (i.e. an "arc" of the circle) and fire all of them, resulting in the outer vertices losing one vertex and the vertices adjacent to them but not in the arc gaining a chip. This may be repeated for iteratively larger arcs, each time including one more vertex on each end. Doing so enables us to send those chips an arbitrary distance away from the ends of the original arc (as long as there are no self-intersection issues).

We will first look at odd n. From the sink, give each vertex a single chip by pairing vertices equidistant from the sink, and having the sink fire chips to those two vertices. Then, pair vertices equidistant from vertex v and fire their chips to vertex v. This results in vertex v having all n chips, with no other vertex having chips. See Fig. 8 for an example of the described firing process.

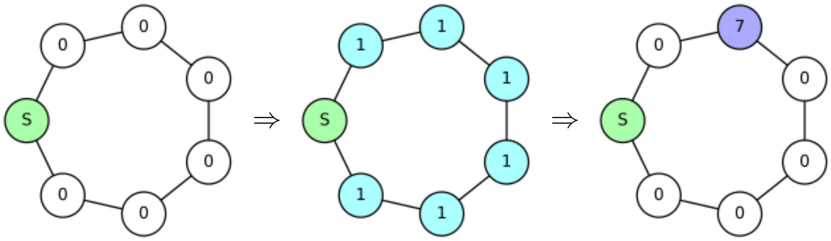


Figure 8: Example firing process for C_n when $n = 7$. The sink is colored green, vertices with 1 chip are colored blue, and vertices with $n = 7$ chips are colored purple.

Now we will look at even n. From the sink, give each vertex except the vertex diametrically opposite the sink a single chip as before. Then, pair vertices equidistant from vertex v except the vertex diametrically opposite v and fire their chips to vertex v like before. This results in vertex v having $n-1$ chips, the vertex diametrically opposite the sink having -1 chips, and the vertex diametrically opposite v having 1 chip. Then, fire chips from the sink and the vertex with 1 chip to v and the vertex diametrically opposite the sink, resulting in v having n chips and no other vertex having chips. See Fig. 9 for an example of the described firing process.

Finally, we will show that n is the smallest element in I. We will do this by showing the firing vector for the case where the sink and v are adjacent is irreducible. Let the sink be vertex 1, with the vertices labeled in order so that v is vertex n. First, fire vertex 1. Then, fire vertices 1 and 2. After that, fire vertices 1, 2, and 3, and so on, with the final step firing vertices 1 through $n-1$. Notice that each step pushes a chip in the positive direction; the first step sends a chip to vertex 2, the next step moves that chip to vertex 3, and so on. After all these steps, that chip will arrive at vertex n. At the same time, each step gives vertex n a chip from the sink, vertex 1. Hence, after

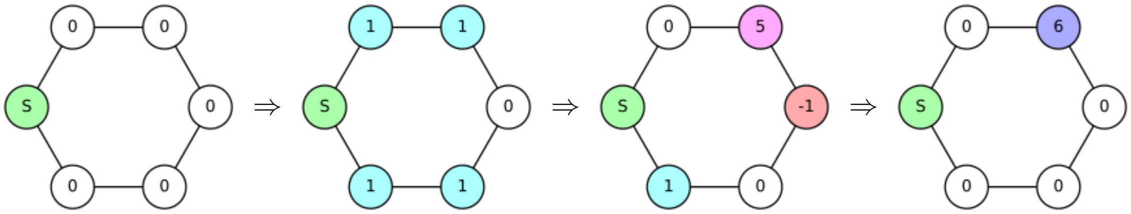


Figure 9: Example firing process for C_n when $n = 6$. The sink is colored green, vertices with 1 chip are colored blue, vertices with -1 chips are colored red, vertices with $n - 1 = 5$ chips are colored pink, and vertices with $n = 6$ chips are colored purple.

these $n - 1$ steps, vertex n will receive $1 + (n - 1) = n$ chips, with no other vertex having chips. During this process, vertex $n - 1$ was only fired once, and thus the firing vector is irreducible. ■

5 The Complete Identity Property

Using the definition of compatibility, a graph has the complete maximal identity property if and only if m_G is compatible. With this concept, the definition of the complete maximal identity property can be generalized to simply when there exists a compatible configuration that is recurrent for all sinks but need not be m_G, as seen in the following definition.

Definition 5.1. A graph G is said to have the **complete identity property** if there exists a chip configuration c on all the vertices such that for all choices of sink s, the configuration c with respect to sink s is the recurrent identity for the sandpile of G at s.

If G has the complete identity property, c_G is the chip configuration on all vertices that gives the recurrent identity for all choices of sink s.

Note that if a graph G has the complete maximal identity property, it has the complete identity property.

Odd cycle graphs have the complete maximal identity property, but even cycle graphs do not. The generalization of the complete maximal identity property to the complete identity property helps resolve this, as seen by Proposition 5.1.

Proposition 5.1. *For any positive integer n, attaching a single tree of size 1 to any vertex in the even cycle graph C_{2n} results in a graph with the complete identity property.*

Proof. Say G is the graph resulting from attaching a single tree of size 1 to any vertex in C_{2n}. Let vertex v_0 be the vertex added to the even cycle graph and vertex v_1 be the vertex connected to it. Let the vertices of the cycle graph be numbered v_1 through v_{2n} in clockwise order. Notice that v_{n+1} is diametrically opposite of v_1. See Fig. 10 for an example of G, when $n = 3$.

We claim the common recurrent identity configuration c_G is the configuration that has 0 chips at v_0 and v_{n+1}, 2 chips at v_1, and 1 chip everywhere else. Let this configuration be denoted as d_G

for the proof. We will show that this configuration is the recurrent identity for all of the vertices. See Fig. 11 for an example of d_G, when $n = 3$.

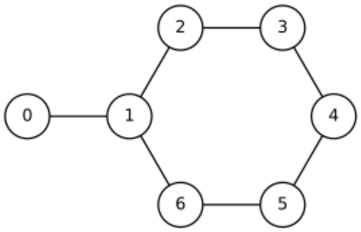


Figure 10: G when $n = 3$

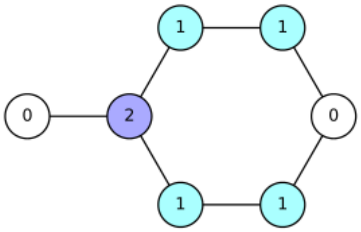


Figure 11: d_G when $n = 3$. Vertices with 1 and 2 chips are colored blue and purple, respectively.

Case 1: sink at v_0. To prove that the configuration d_G is recurrent, take any stable configuration and add chips to it to yield the configuration that has 2 more chips than d_G at v_{n+1}, which is the configuration resulting from taking the maximal stable configuration and adding 1 chip to v_{n+1}, or $m_G + \mathbf{e}_{n+1}$. From here, we stabilize to reach d_G. Vertex v_{n+1} fires, activating $v_{n+1\pm1}$. Vertex $v_{n+1\pm1}$ fires, activating v_{n+1} again as well as $v_{n+1\pm2}$. This process continues, sending the two chips originally at v_{n+1} to v_1, which fires, activating the vertices all the way back to v_{n+1}, and then repeats this process again, giving the two chips to v_0, thus reaching d_G. Hence, d_G is recurrent. Furthermore, using the same reasoning, all the chips in d_G may be moved to v_0 as there are no chips at v_{n+1}, and the chips at $v_{n+1\pm i}$ (chips that are the same distance from vertex 1) may be paired together and moved towards v_1, and from there all the chips moved to v_0 by firing the entire cycle graph as many times as necessary.

Case 2: sink at v_1. Follow the same procedure as v_0, ending when the two chips reach v_1.

Case 3: sink at v_{n+1}. Follow a similar procedure but for the proof of d_G being recurrent, add chips to any stable configuration to yield the configuration that is the maximal stable configuration with two extra chips at v_1, and stabilize to yield d_G.

Case 4: sink at v_i for $2 \leq i \leq n$. To prove that the configuration d_G is recurrent, take any stable configuration and add chips to it to yield the configuration that has 1 more chip than m_G at v_{n+i}. Firing v_0 immediately after each time v_1 fires, the proof of the recurrence of d_G proceeds exactly like the case for the sink at v_1, acting on the cycle graph.

To prove that d_G is equivalent to the all-zero configuration, pair off vertices equidistant from v_i in the cycle graph and move their chips to v_i, resulting in the configuration with v_1 having 1 chip, v_{n+1} having -1 chips, and v_{n+i} having 1 chip. Using this observation, we move the two chips at v_1 and v_{n+i} away from each other until the chip from v_{n+i} reaches v_{n+1}, and thus the chip at v_1 reaches v_i, the sink. This results in the all-zero configuration, and thus d_G is the recurrent identity at v.

Case 5: sink at v_i for $n + 2 \leq i \leq 2n$. By symmetry, follow the same procedure as described in case 4 for v_{2n+2-i} where $2 \leq 2n + 2 - i \leq n$.

This completes the proof. ■

In order to create a graph from an even cycle graph that has the complete identity property, we needed to add a single tree of size 1 to any vertex. This property is also seen in the case of complete bipartite graphs, as the following theorem shows.

Theorem 5.2. *For all positive integers m, n, attaching a single tree of size 1 to any vertex in the complete bipartite graph $K_{m,n}$ results in a graph that has the complete identity property.*

Proof. It suffices to show the result for when the additional tree is attached to one of the m vertices, as the same proof would hold with m and n interchanged. Furthermore, we may assume that $m, n \geq 2$, as if any of them are 1, the resulting graph is a tree and thus has the complete maximal identity property and hence the complete identity property by Proposition 3.2.

Say the additional vertex is a, the vertex it is attached to is b, the n vertices of one side of the complete bipartite graph compose set C, and the $m-1$ vertices in the set of m vertices of one side of the complete bipartite graph that is not b compose set D. We claim c_G is the configuration with a having 0 chips, b having n chips, each vertex in C having $m-1$ chips, and each vertex in D having 0 chips. Let this configuration be denoted as d_G for the proof. See Fig. 12 for an example of the graph and the naming convention for the vertices.

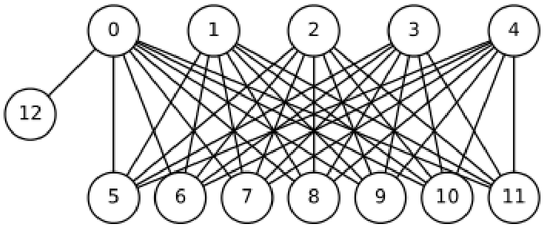


Figure 12: Example of $K_{m,n}$ with a tree of size 1 attached, for $m = 5, n = 7$. Vertex 12 is a, vertex 0 is b, vertices 5 through 11 compose set C, and vertices 1 through 4 compose set D.

We will separate into four cases depending on where the sink is.

Case 1: sink at a. From any stable configuration add chips to reach the configuration with b having n chips, each vertex in C having $m-1$ chips, and each vertex in D having n chips. We then stabilize. This results in all vertices in D firing, then all vertices in C firing, then all vertices in D firing again to result in b having $2n$ chips and C having $2m-3$ chips each. Then, firing b, all vertices in C, and then all vertices in D results in a net loss of 1 chip at b, and this occurs as long as b has at least $n+1$ chips to start the cycle, and the cycle starts with all vertices in C having at least $m-1$ chips, which it does as $m \geq 2$, so $2m-3 \geq m-1$. Hence, after n iterations of this cycle, we eventually reach b having n chips, all vertices in C having $2m-3$ chips each, and all vertices in D having 0 chips each. Fire all vertices in C and then all vertices in D to yield all vertices in C having lost 1 chip each, and b having gained n chips. Follow the cycle previously shown to return b back to having n chips. This process can thus result in a net loss of 1 chip at each vertex in C, as long as all vertices in C had at least m chips to start with, giving it at least $m-1$ chips when it enters the other cycle. Cycling this process until it is no longer possible results in b having n

chips, all vertices in C having $m-1$ chips each, and all vertices in D having no chips. This is d_G, and it is stable. Hence, d_G is recurrent. Now, to prove that d_G is the recurrent identity, fire all vertices in $C \cup D$ each $m-1$ times to clear all vertices in $C \cup D$ of chips, and then fire b, all vertices in C, and all vertices in D until b is also clear of chips. Table 1 shows a stabilization of this configuration to d_G, and then the unrestricted firing sequence that takes d_G to $\mathbf{0}$, tracking the chips at each non-sink vertex. As during the process the vertices in C are indistinguishable, the column for C tracks the number of chips at each of the vertices in C. The chips at each vertex in D are similarly tracked. Hence, d_G is the recurrent identity.

a	b	C	D	Firing step
Sink	n	$m-1$	n	D, C, D fire
Sink	$2n$	$2m-3$	0	b, C, D fire
Sink	$2n-1$	$2m-3$	0	Repeat firing b, C, D a total of n times
Sink	n	$2m-3$	0	C, D fire, b, C, D fire n times. Repeat until no longer possible
Sink	n	$m-1$	0	Reached d_G
Sink	n	$m-1$	0	C, D fire $m-1$ times
Sink	nm	0	0	b, C, D fire nm times
Sink	0	0	0	Reached $\mathbf{0}$

Table 1: Firing sequences for proof that d_G is the recurrent identity with sink at a

Case 2: sink at b. From any stable configuration add chips to reach the configuration with a having 0 chips, each vertex in C having $m-1$ chips, and each vertex in D having n chips. Stabilizing this configuration results in first firing all vertices in D, resulting in all vertices in C having $2m-2$ chips each. Then, firing all vertices in C and then all vertices in D results in a net loss of 1 chip at each vertex in C, and the cycle works as long as all vertices in C start with at least m chips so that they can fire. Repeating this process, we find that the stabilization of the specified accessible configuration is d_G. Hence, d_G is recurrent. Now, to prove that d_G is the recurrent identity, fire all vertices in $C \cup D$ each $m-1$ times to result in the all-zero configuration. Table 2 shows the firing sequences that prove d_G is the recurrent identity for sink at b.

a	b	C	D	Firing step
0	Sink	$m-1$	n	D fires
0	Sink	$2m-2$	0	C, D fire repeatedly until no longer possible
0	Sink	$m-1$	0	Reached d_G
0	Sink	$m-1$	0	C, D fire $m-1$ times
0	Sink	0	0	Reached $\mathbf{0}$

Table 2: Firing sequences for proof that d_G is the recurrent identity with sink at b

Case 3: sink in C. Say the sink is $s_C \in C$, and let $C' = C \setminus \{s_C\}$. From any stable configuration add chips to reach the configuration with a having 0 chips, b having n chips, each vertex in C' having m chips, and each vertex in D having n chips. We fire all vertices in C' to yield b and each vertex in D having $2n-1$ chips each, both being active. Notice that we may fire b and all vertices in D, and then all vertices in C', as long as b and all vertices in D have at least $n+1$ chips each, and without any requirement of the starting chips for all vertices in C', as they will each get the m chips they need to fire. This process results in a net loss of 1 chip at each of b and all vertices in D. Performing this operation until it is no longer possible, with n chips at b and each vertex in D each, we then fire all vertices in D to yield d_G. Hence, d_G is recurrent. To prove that d_G is the recurrent identity, notice that d_G is equivalent to the configuration that preceded it in the proof of its recurrence, which had n chips at each of b and all vertices in D. Continuing the procedure of firing b and all vertices in D and then all vertices in C, we can do this n times to clear b and all vertices in D of chips, resulting in the all-zero configuration. Table 3 shows the firing sequences that prove d_G is the recurrent identity for sink in C.

a	b	C'	D	Firing step
0	n	m	n	C' fires
0	$2n-1$	0	$2n-1$	b, D, C' fire repeatedly until no longer possible
0	n	0	n	D fires
0	n	$m-1$	0	Reached d_G
0	n	$m-1$	0	Backfire D
0	n	0	n	b, D, C' fire n times
0	0	0	0	Reached $\mathbf{0}$

Table 3: Firing sequence for proof of recurrence of d_G with sink in C

Case 4: sink in D. Say the sink is $s_D \in D$, and let $D' = D \setminus \{s_D\}$. From any stable configuration add chips to reach the configuration with a having 0 chips, b having $2n$ chips, each vertex in C having $m-1$ chips, and each vertex in D' having n chips. Notice that at this state b and all vertices in D' are active. Furthermore, firing these vertices and then a gives $m-1$ chips to each vertex in C, so as long as they each have at least 1 chip, they can fire. This results in b and D' regaining their n chips, and costing each vertex in C m chips, meaning that each vertex in C has lost one chip in total. We perform this procedure $m-1$ times to result in the configuration with $2n$ chips at b and n chips at each vertex in D'. Firing b, a, and all vertices in D' then gives d_G. Hence, d_G is recurrent. To prove that d_G is the recurrent identity, fire a and b to clear a and b of chips, resulting in C having m chips. Backfire the sink m times to result in the all-zero configuration. Table 4 shows the firing sequences that prove d_G is the recurrent identity for sink in D.

This completes the proof. ■

By Proposition 5.1 and Theorem 5.2, we see that even cycles and complete bipartite graphs both have the property that attaching a single tree of size 1, or a single edge and vertex, to any vertex in the graph results in a graph with the complete identity property. In addition, adding a

a	b	C	D'	Firing step
0	$2n$	$m-1$	n	b, D', a, C fire $m-1$ times.
0	$2n$	0	n	b, D', a fire
0	n	$m-1$	0	Reached d_G
0	n	$m-1$	0	a, b fire
0	0	m	0	s_D backfires m times
0	0	0	0	Reached $\mathbf{0}$

Table 4: Firing sequence for proof of recurrence of d_G with sink in D

single edge and vertex to a tree results in a tree, which has the complete maximal identity property by Proposition 3.2, which implies the complete identity property. All of these three types of graphs are bipartite. While not all bipartite graphs, or even regular bipartite graphs, have this property (for example, the hypercube in 3 dimensions, which is isomorphic to $C_4 \square K_2$, where the Cartesian product $\square$ is defined in Section 6, is one such counterexample), a computer search found that all connected graphs of 10 vertices or less which had this property were bipartite. This motivates the following conjecture.

Conjecture 5.3. *Let G be a connected graph. If for all vertices $v \in G$, attaching a single tree of size 1 to v results in a graph with the complete identity property, then G is bipartite.*

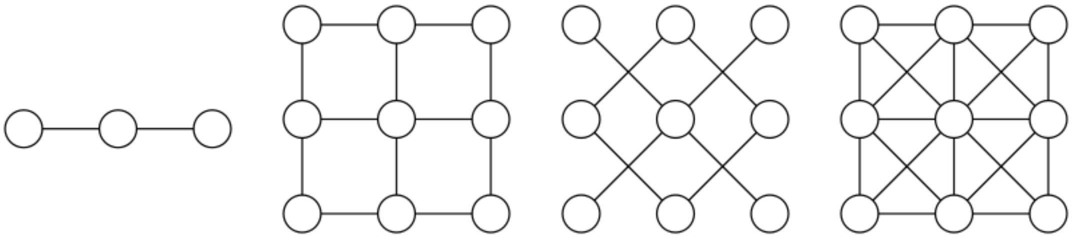
6 Conjectures on Graph Products

Recall that the Cartesian product, tensor product, and strong product are binary operations on graphs that form a graph whose vertices are ordered pairs of vertices of the two daughter graphs. For the Cartesian product, denoted $\square$, two vertices share an edge if in one of the daughter graphs the two vertices share an edge and in the other the vertices are the same. For the tensor product, denoted $\times$, two vertices share an edge if in both daughter graphs the two vertices share an edge. The strong product, denoted $\boxtimes$, is the union of the Cartesian product and the tensor product. See Fig. 13 for examples of the Cartesian, tensor, and strong products between two copies of the path graph with 3 vertices P_3.

Investigating the behavior of graph products with respect to the complete maximal identity property, we find that in general, the strong product, Cartesian product, and tensor product all do not preserve the complete maximal identity property. However, we do find some patterns.

Proposition 6.1. *Let P_n be the path graph with n vertices. The strong product between P_2 and P_k has the complete maximal identity property if and only if $k = 2$ or $k \equiv 1 \pmod 3$.*

Proof. We will refer to the canonical labelling of the vertices of the graph $P_2 \boxtimes P_k$ as via the ordered pairs (i, j) where $i \in \{0, 1\}$ and $j \in \{0, 1, \ldots, k-1\}$. By symmetry, it suffices to prove the graph has the maximal identity property for sinks with $i = 0$ and $j \leq \frac{k-1}{2}$.



(a) P_3 (b) $P_3 \square P_3$ (c) $P_3 \times P_3$ (d) $P_3 \boxtimes P_3$

Figure 13: The Cartesian, tensor, and strong products between two copies of P_3

We will prove the $j = 0$ case first. Starting with the maximal stable configuration, incrementally fire all the vertices with second coordinate $k-1$ until those have no chips, and then fire all the vertices with second coordinate at least $k-2$ until those have no chips, and so on. The resulting configuration has $4k-6$ chips at each of the two vertices with second coordinate 1. By symmetry, in order for the two vertices with second coordinate 1 to have the same number of chips (eventually 0), the number of times the vertices with second coordinate at least 1 can be fired must all be the same. Fire all of these vertices $k-2$ times. Then the three vertices connected to the sink each have $2k-2$ chips. Fire all non-sink vertices $2k-2$ times to clear the graph of all chips.

Now we assume $k \geq 3$. For $1 < j \leq \frac{k-1}{2}$, we follow a similar process to result in the vertices with second coordinate $j-1$ having all the chips originally with second coordinate less than j, and thus having $4j-2$ chips each. Similarly, the vertices with second coordinate $j+1$ have $4k-4j-6$ chips each. Fire the vertices with second coordinate at least $j+1$ a total of $k-2j-1$ times each to have the vertices with second coordinate $j \pm 1$ having $4j-2$ chips each, and $(1, j)$ having $4k-8j$ chips.

In order for the vertices with second coordinate not equal to j to all have 0 chips, they must all be fired the same number of times. Hence, we essentially only have two operations: fire $(1, j)$ or fire all vertices with second coordinate not equal to j. From this, we can see that we can fire all the non-sink vertices $4j-2$ times, resulting in all the non-sink vertices having no chips except $(1, j)$ having $4k-12j+2$ chips. Now notice that the smallest increment by which we can change the number of chips at that vertex without changing the other vertices is to fire it twice and then fire all the non-sink vertices once. Or, in other words, fire it once and backfire the sink once. This results in a net loss of 6 chips. So for $P_2 \boxtimes P_k$ to have the complete maximal identity property we must have that $4k - 12j + 2 \equiv 0 \mod 6$ for all values of j, or equivalently $k \equiv 1 \mod 3$. If this were not the case, we would be able to get the all-zero configuration with a fractional number of firings, and as the reduced Laplacian is nonsingular, this is the unique firing vector needed, and thus the maximal stable configuration is not equivalent to the all-zero configuration.

To prove that the other graphs do not have the complete maximal identity property, let $k \not\equiv 1$ mod 3, and use the same process as before to arrive at $(1, j)$ having 2 or 4 chips. Using the fact that the reduced Laplacian is non-singular, there is a unique firing vector that results in this configuration. However, it does not have integer entries, as the configuration with 6 chips at $(1, j)$ does not have all of its entries being multiples of 3, rather having all of the vertices with second coordinate not equal to j being backfired once and $(1, j)$ being backfired twice. Hence, the two configurations resulting from $k \not\equiv 1 \mod 3$ are not equivalent to the identity, and thus do not have the complete maximal identity property. ■

After looking at whether $P_i \boxtimes P_j$ has the complete maximal identity property for all values of i and j where $1 \leq i, j \leq 100$, we conjecture that the cases presented in Proposition 6.1 are the only such graphs with the complete maximal identity property:

Conjecture 6.2. *For $i \leq j \in \mathbb{Z}_{>1}$, the only graphs $P_i \boxtimes P_j$ which have the complete maximal identity property are $P_2 \boxtimes P_j$ where $j \equiv 1 \mod 3$ or $j = 2$, which yields K_4.*

A computer program in SageMath verified that the conjecture holds for $1 \leq i \leq j \leq 100$.

Similarly, the following proposition on the Cartesian product was proven.

Proposition 6.3. *The Cartesian product between K_4 and P_2 has the complete maximal identity property.*

After looking at whether or not $K_i \square P_j$ has the complete maximal identity property for all values of i and j where $1 \leq i, j \leq 50$, we conjecture that that the case presented in Proposition 6.3 is the only such graph with the complete maximal identity property.

Conjecture 6.4. *For $i, j \in \mathbb{Z}_{>1}$, the only graph $K_i \square P_j$ which has the complete maximal identity property is $K_4 \square P_2$.*

Another computer program in SageMath verified that the conjecture holds for $1 \leq i, j \leq 50$.

We provide a proof of the special cases of Conjecture 6.4 when $j = 2, 3$.

Proof of the $j = 2, 3$ cases. For $j = 2$, let the sink be (0,0). Instead of looking at abscissas of 0 through $i - 1$, by symmetry all the positive abscissas for a specified ordinate must fire the same number of times (even if this were not true, the solution that follows would yield a non-integer number of firings for each vertex, and as the reduced Laplacian is non-singular, no other firing vector will yield the all-zero configuration, and thus the proof holds). So, we will combine the vertices with the same ordinate and a positive abscissa together to yield a weighted cycle graph of 4 vertices. The reduced Laplacian is

$$\Delta' = \begin{bmatrix} i & 1-i & 0 \\ 1-i & 2i-2 & 1-i \\ 0 & 1-i & 2i-2 \end{bmatrix}.$$

Thus, the firing vector $\mathbf{v}$ to reach the maximal stable configuration $\mathbf{c}$ is

$$\mathbf{v} = (\Delta')^{-1}\mathbf{c} = (\Delta')^{-1} \begin{bmatrix} i-1 \\ (i-1)^2 \\ (i-1)^2 \end{bmatrix} = \frac{1}{i+2} \begin{bmatrix} 3i(i-1) \\ (i-1)(3i+2) \\ 2(i-1)(i+1) \end{bmatrix}.$$

To have the complete maximal identity property, it suffices to show that for this particular sink, $\mathbf{v}$ has integer entries. So

$$i+2 \mid \gcd\left(3i(i-1),(i-1)(3i+2),2(i-1)(i+1)\right).$$

We will first analyze $i+2 \mid 3i(i-1)$. Notice that $\gcd(i,i+2)=\gcd(2,i) \mid 2$. We also have $\gcd(i+2,i-1)=\gcd(i+2,3) \mid 3$ and $\gcd(i+2,3) \mid 3$. Hence, we find that $i+2 \mid 2\cdot 3^2$. This corresponds to integer values of i greater than 1 being 4, 7, and 16. Verifying that these hold for the other two divisibility criteria, we find only $i=4$ yields the complete maximal identity property.

For $j=3$, we follow the same process for combining vertices. The reduced Laplacian is

$$\Delta' = \begin{bmatrix} i & 1-i & -1 & 0 & 0 \\ 1-i & 2i-2 & 0 & 1-i & 0 \\ -1 & 0 & i+1 & 1-i & 0 \\ 0 & 1-i & 1-i & 3i-3 & 1-i \\ 0 & 0 & 0 & 1-i & 2i-2 \end{bmatrix}.$$

The firing vector $\mathbf{v}$ to reach the maximal stable configuration c is

$$\mathbf{v} = (\Delta')^{-1}\mathbf{c} = \frac{1}{(i+1)(i+3)} \begin{bmatrix} 2i(3i-2)(i+3) \\ (3i-2)(2i^2+6i+1) \\ 5i^3+8i^2-6i+1 \\ 5i^3+11i^2-5i-1 \\ (3i-2)(i^2+3i+1) \end{bmatrix}.$$

This requires that

$$(i+1)(i+3) \mid \gcd\left(5i^3+8i^2-6i+1, 5i^3+11i^2-5i-1\right)$$

which results in

$$(i+1)(i+3) \mid 3i^2+i-2.$$

With $i+1 \mid 3i^2+i-2$, the condition is equivalent to $i+3 \mid 3i-2$, or $i+3 \mid 11$, or $i=8$. But this does not yield a vector with integer entries, so hence there are no solutions for $j=3$. ■

Acknowledgements

I sincerely thank my mentor, Yibo Gao of the Department of Mathematics at the Massachusetts Institute of Technology (MIT), for guiding me throughout the entire math research process as well as constantly introducing me to new ideas in chip-firing. I would like to thank Professor David Perkinson of the Department of Mathematics at Reed College for suggesting the project and his guidance during the chip-firing research project. I would also like to thank Dr. Tanya Khovanova of the Department of Mathematics at MIT for her advice throughout the research process. I thank Dr. Slava Gerovitch and Professor Pavel Etingof of the Department of Mathematics at MIT for their operation of the PRIMES-USA program; I also thank the MIT PRIMES-USA Program as well as the Department of Mathematics at MIT.

References

[1] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of 1/f noise. *Phys. Rev. Lett.*, 59:381–384, 1987.

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

[3] Norman L. Biggs. Algebraic potential theory on graphs. *Bull. London Math. Soc.*, 29(6):641–682, 1997.

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

[5] Anders Björner, László Lovász, and Peter W. Shor. Chip-firing games on graphs. *European J. Combin.*, 12(4):283–291, 1991.

[6] Sergio Caracciolo, Guglielmo Paoletti, and Andrea Sportiello. Explicit characterization of the identity configuration in an abelian sandpile model. *J. Phys. A*, 41(49):495003, 17, 2008.

[7] Robert Cori and Dominique Rossin. On the sandpile group of dual graphs. *European J. Combin.*, 21(4):447–459, 2000.

[8] Scott Corry and David Perkinson. *Divisors and sandpiles.* American Mathematical Society, Providence, RI, 2018.

[9] Deepak Dhar. Self-organized critical state of sandpile automation models. *Phys. Rev. Lett.*, 64(23):2837, 1990.

[10] 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*, volume 60 of *Progr. Probab.*, pages 331–364. Birkhäuser, Basel, 2008.

[11] Yvan Le Borgne and Dominique Rossin. On the identity of the sandpile group. *Discrete Math.*, 256(3):775–790, 2002. LaCIM 2000 Conference on Combinatorics, Computer Science and Applications (Montreal, QC).

[12] Lionel Levine. The sandpile group of a tree. *European J. Combin.*, 30(4):1026–1035, 2009.

[13] Richard P. Stanley. *Algebraic combinatorics.* Undergraduate Texts in Mathematics. Springer, Cham, 2018.

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

[15] Jan van den Heuvel. Algorithmic aspects of a chip-firing game. *Combin. Probab. Comput.*, 10(6):505–529, 2001.

[16] David G Wagner. The critical group of a directed graph. *arXiv preprint math/0010241*, 2000.