INTRODUCTION TO THE CHIP-FIRING GAME

HAOLIN (TONY) ZHONG

Abstract. The chip-firing game is a dynamic process on a graph in which vertices weighed by their "chip values" (which together compose a chip configuration on the graph) fire chips to neighboring vertices under a set of rules determining when a vertex is "ready to fire" or, conversely, when a configuration is "stable" in the sense that no vertex is ready to fire. This paper introduces fundamental definitions and properties of this combinatorial subject. Using the graph Laplacian, we define two types of firings (firing at an individual vertex and cluster-firing of multiple vertices) and two types of games (with or without a *sink*, through which chips are consumed without returning to the system). From there, we will study basic properties of the game, including the fact that order does not matter in sequences of firings at individual vertices (confluence), criteria for when a configuration will eventually cease to fire, and the structures and duality of critical and superstable configurations. We are concerned with two basic motivating questions throughout this paper: 1) Given the rules of firing and whether a sink exists, since the behavior of a chip-firing process is entirely determined by its initial chip configuration, can we effectively categorize these configurations from a combinatorial perspective to study their properties? 2) If the chip-firing game starting from some initial chip configuration can either eventually stabilize or go on infinitely, to what extent do we need to modify the chip values on different types of configurations so that they will behave otherwise? What do we make of configurations that remain stable even after a substantial change to its chip values?

Contents

1. Introduction 2
2. Confluence 3
3. Stabilization of a Chip Configuration 6
4. Criticality and Superstability 8
4.1. Chip-firing Games Involving a Sink 8
4.2. Criticality 9
4.3. Superstability 11
5. Duality 12
5.1. Energy Minimizers 12
5.2. Proof of Duality 14
6. Conclusion 16
Acknowledgments 16
References 18

Date: AUGUST 29, 2022.

1. Introduction

As a subject of combinatorics and graph theory, the chip-firing game has attracted increasing attention from mathematicians since its formalization in the 1980s. One of the earliest appearances of this subject is in Spencer's 1986 paper on the "balancing game." To get a sense of how the game works, let us start with a connected graph G. For the purpose of this paper, assume G is finite, simple, and undirected; the study of infinite or directed chip-firing processes may be a subsequent pursuit of the reader. In the chip-firing process, each vertex is assigned a non-negative integer chip value. We define this notion formally as the chip configuration of the graph.

Definition 1.1. Let $G = (V, E)$ be a graph of vertex set V and edge set E with $|V| = n$. A *chip configuration* for G is a nonnegative integer-valued vector $\mathbf{c} = (c_1, c_2, ..., c_n) \in \mathbb{Z}^n_{\geq 0}$, where c_i is the chip value assigned to vertex i.

To actually make sense of G as a chip-firing system, we define the following notions.

Definition 1.2. A vertex v is *ready to fire* if $c_v \geq \deg(v)$, i.e., the chip value assigned for v is greater than or equal to its number of neighbors.

During the chip-firing process, a vertex *fires* by sending exactly one chip to each of its neighbors. A new chip configuration $\mathbf{c}'$ is reached after the firing, in which the chip value of the firing vertex decreases by its degree and that of each of its neighbors increases by 1. A firing is *legal* if all entries of $\mathbf{c}'$ remain nonnegative. Equivalently, we say a vertex is "ready to fire" if its firing is legal.

The chip-firing process can continue indefinitely or terminate at a *stable* configuration, in which no vertex is ready to fire. Figure 1 illustrates an example of a chip-firing process that terminates (i.e., reaches a stable configuration) within finite firings. Figure 2 is an example of an infinite chip-firing process that runs in cycles.

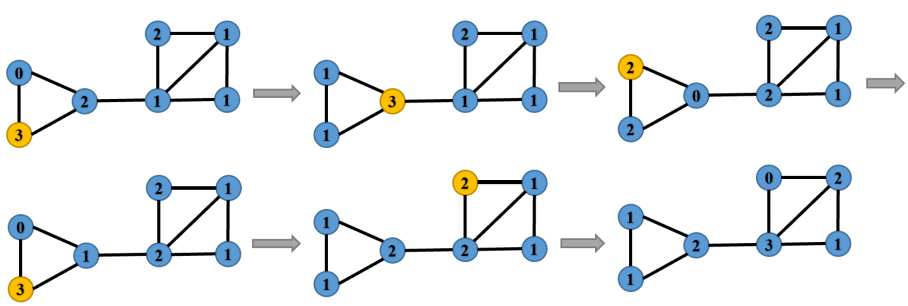

Figure 1. An example of a finite chip-firing process. The vertex that is about to fire at each step is distinguished.

The chip-firing game may be considered a model of various real-world phenomena—the circulation of currency in an economy, the firing of neurons in the brain, etc. More abstractly, the chip-firing game is a dynamic system. The subject has been historically introduced via two distinct motivations, the abelian sandpile model and the combinatorial dollar game. From the perspective of the abelian sandpile model, without further discussing the details of this concept, we are interested in how

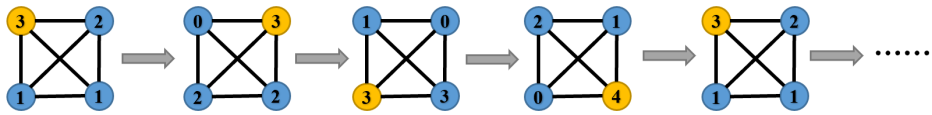

FIGURE 2. An example of an infinite chip-firing process.

the chip-firing process as a dynamic system demonstrates *self-organized criticality*. In particular, this paper explores the question of roughly to what extent a stable configuration (i.e., a chip-valued graph in which no vertex fires) will avalanche into additional firings or remain approximately "stable" upon the addition of some chips. The answer to this question is closely connected to the study of critical and superstable configurations. The combinatorial motivation, on the other hand, concerns the abstraction of the chip-firing process via methods such as matroid theory, a generalization of the linear algebraic notions of vector spaces and linear independence in a combinatorial environment. This paper will not cover matroid theory, but the reader should certainly see it as a next step in their studies.

This paper aims to introduce to the reader the basic notions and properties of the chip-firing game as a combinatorial subject and a dynamic system, eventually leading to a brief exploration of the important questions posed above. We roughly follow *The Mathematics of Chip-Firing* by Klivans but consider various orders and ways to introduce our concepts and theorems and incorporate different approaches to proofs. Chapter 2 discusses the theorem of commutativity of the chip-firing process. Chapter 3 introduces a simple criterion for finite vs. infinite graphs. Chapter 4 focuses on chip-firing processes involving a sink, from which arises the important notions of criticality and superstability. We will prove a duality property between these critical and superstable configurations in Chapter 5. Chapter 6 concludes the paper and points to directions for further studies in the subject.

2. CONFLUENCE

To begin with, we introduce the *graph Laplacian*, a way of representing information of a graph fundamental to the study of the chip-firing game. The definition follows naturally from the definition of vertex firing. Recall that when a vertex i fires, its chip value decreases by $\deg(i)$ and the chip value of each of its neighbors increases by 1. We record this information through the vector $\mathbf{w_i}$. For a vertex i and any vertex j,

$$(\mathbf{w_i})_j = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

In other words, this vector represents the inverse of the changes in chip value during a firing: each neighboring vertex to i records a decrease by 1. This leads to the definition of the graph Laplacian, which stacks all of the w_i's in an $n \times n$ matrix.

Definition 2.1. Let $G = (V, E)$, $|V| = n$. The *graph Laplacian* L is the $n \times n$ matrix

$$L_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \text{ and } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

For instance, the graph from Figure 1 may be represented with the following Laplacian. According to this drawing, the vertices are ordered from left to right, and then from top to bottom on the same column.

$$\begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

In the context of the matrix, a vertex firing at i corresponds to subtracting the ith row of the graph Laplacian from the current chip configuration. In other words, if we reach configuration $\mathbf{c}'$ from $\mathbf{c}$ by firing at i, then

$$\mathbf{c}' = \mathbf{c} - L\mathbf{e_i},$$

where $\mathbf{e_i}$ is the ith vector in the standard basis of $\mathbb{R}^n$. Whether the chip configuration is represented as a row or column does not matter since the graph Laplacian is symmetric. It will also be useful here to redefine the notion of stable configuration in terms of the Laplacian. For a graph G with $|V| = n$, a configuration $\mathbf{c}$ is *stable* if

$$\mathbf{c} - L\mathbf{e_i} < \mathbf{0} \ \ \forall \ \ 1 \leq i \leq n,$$

where the inequality is componentwise. This redefinition is just another expression of the condition that no vertex is ready to fire. Furthermore, we observe that L is the difference of two familiar matrices: let D be the diagonal matrix whose diagonal entries are the degrees of the corresponding vertex, and let A be the adjacency matrix with 1 at entry ij when $(i,j) \in E$ and 0 everywhere else. Then

$$L = D - A.$$

For now, we are considering chip-firing processes in which vertices fire individually in sequence. Another type of firing is called *cluster firing*, in which a collection of vertices in the graph all fire simultaneously. Like individual firing, a cluster firing is legal if it results in a nonnegative chip configuration on all vertices. There are critical mathematical differences between individual and cluster firings, and the motivation for studying cluster firing will be discussed in greater detail in section 3. Nevertheless, we define cluster firing now as it will soon become useful in proving our first major theorem.

Let $U = \{i_1, ..., i_k\} \subseteq V$ be a non-empty collection of vertices in G. Let λ_U be the characteristic vector of U, i.e.,

$$\lambda_U = \sum_{j=1}^{k} e_{i_j}$$

Definition 2.2. For a graph G with Laplacian L_G, let $U \subseteq V$ be non-empty. From an initial configuration $\mathbf{c}$, the *cluster firing* of U results in the final configuration $\mathbf{c}'$ via the following transformation:

$$\mathbf{c}' = \mathbf{c} - L_G \lambda_U$$

For example, on the graph in Figure 1, at the third step, we could have cluster-fired the two leftmost vertices of chip values of 2 to directly get the penultimate configuration. A more interesting fact is that some legal cluster-firings do not require that each vertex being fired in the cluster firing is ready to fire on its own. For example, on the grid graph from Figure 3 below, we see that the vertex colored green in the initial configuration is not ready to fire on its own. However, cluster firing it together with the vertex below results in a nonnegative final configuration—so this move is legal. A simple explanation we give for this perhaps counterintuitive phenomenon is that, if we treat the two vertices as firing individually, we see that they fire to each other, so these two particular firings cancel out. Hence, the node in green is really only effectively firing at two other vertices, which is not a problem when it has a chip value of 2.

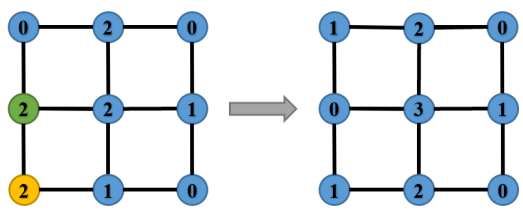

FIGURE 3. An example of a legal cluster firing. Note that the node colored in green would not have been ready to fire on its own.

An immediate question that may arise with the process of firing at individual vertices is, given that there are multiple vertices ready to fire at a moment, whether the order of firing affects long-term properties of the chip-firing process. In other words, starting from the same initial configuration $\mathbf{c}$, will different orders of firing eventually lead to different stable configurations? Will some orders run on forever while others stop at a stable configuration? We will show that the answer is no.

A chip configuration $\mathbf{d}$ is *reachable* from another configuration $\mathbf{c}$ if there exists a sequence of legal firings from $\mathbf{c}$ to $\mathbf{d}$. A fundamental property of the chip-firing process is *confluence*, which explains why order doesn't matter. We formally introduce it at the local and global levels.

Theorem 2.3 (Commutativity of the Chip-firing Process)**.** .

(1) *Local confluence (the diamond property). Let* $\mathbf{c}$ *be an initial chip configuration of a graph* G*. Let* $\mathbf{c_1}$*,* $\mathbf{c_2}$ *be two configurations reachable from* $\mathbf{c}$ *with one firing. Then there exists a configuration* $\mathbf{d}$ *reachable from both* $\mathbf{c_1}$ *and* $\mathbf{c_2}$ *after one firing.*

(2) *Global confluence (uniqueness of stable configuration). Let* $\mathbf{s}$ *be a stable configuration reachable from an initial configuration* $\mathbf{c}$ *in finite firings. Then* $\mathbf{s}$ *is the uniquely reachable stable configuration starting from* $\mathbf{c}$*.*

Proof. (1) Suppose $\mathbf{c_1}$ and $\mathbf{c}$ are two configurations reachable from $\mathbf{c}$ via one firing. Suppose $\mathbf{c_1}$ is obtained from $\mathbf{c}$ by a firing at vertex v_1, and $\mathbf{c_2}$ is obtained from $\mathbf{c}$ by a firing at v_2. W.L.O.G. let us fire from v_1. Note that both before and after this firing, v_2 is ready to fire since a firing can never decrease the chip value except on the firing vertex itself. Similarly, after v_1 fires, v_2 will always remain ready to fire. Hence, we get the common configuration $\mathbf{d}$ from either firing v_2 from $\mathbf{c_1}$ or firing v_1 from $\mathbf{c_2}$. In one step, this is the cluster-firing of $U = \{v_1, v_2\}$ from $\mathbf{c}$, i.e., $\mathbf{d} = \mathbf{c} - L\lambda_{\{v_1,v_2\}}$.

(2) For uniqueness of the stable configuration, start from a chip configuration $\mathbf{c}$ and assume that there exist two stabilizations $\mathbf{c_1}$ and $\mathbf{c_2}$. Since there must exist paths of finitely many firings from $\mathbf{c}$ to $\mathbf{c_1}$ and $\mathbf{c_2}$, simply apply the diamond property as many times as we need to get that any order of firing will eventually reach the same stable configuration. So $\mathbf{c_1}$ and $\mathbf{c_2}$ are equal.

□

The proof to this theorem leads to two immediate consequences.

(1) The sequence of firings from a chip configuration $\mathbf{c}$ to the unique stable configuration it eventually reaches (called the *stabilization* of $\mathbf{c}$, or stab($\mathbf{c}$)), always has the same length. Again, apply the diamond property repeatedly.

(2) In any sequence of such firings leading to a stable configuration, called the *stabilizing sequence*, every vertex fires the same number of times.

At any configuration, because the order of firing among vertices ready to fire does not matter, and that we can guarantee that every vertex that is ready to fire will always get its turn to fire, we may simplify our construction of the whole chip-firing process (with respect to individual firing, not cluster firing) as a step-wise dynamic process. In a sequence of *individual* firings (whether or not they lead to a stabilization), starting from the initial configuration, we define a *step* by firing all vertices ready to fire on a particular configuration. We may treat this as firing these vertices "simultaneously," but again, note that this is not cluster firing: the notion of steps groups a series of individual firings together and serves as a tool to analyze the behavior of the chip-firing process as it evolves laterally over time. On the contrary, to allow cluster firing would mechanically change the rules of the firing process altogether (we may consider this difference along some sort of a "vertical" direction, which will become our primary focus in the next few chapters). From the lateral direction, then, we are interested in studying the steps with which a configuration takes to reach (or never reach) a stabilization instead of the long sequence of firings at individual vertices. The concept of efficiency of a chip-firing process (i.e. the number of steps it takes to reach a stabilization) is fundamentally based on this notion, although we will not discuss this topic in detail. For more on efficiency, see Chapter 2.4 of Klivans.

3. Stabilization of a Chip Configuration

A natural question we may ask about any given graph and chip configuration is whether or not it stabilizes (or, equivalently, whether the game is finite/infinite). There are many approaches to this question, and the most basic theorem is a trichotomy based on the number of edges and vertices of the graph and its *total* chip value across all vertices. This criterion is particularly useful because we do not

need to have the specific configuration. This allows us to draw more generalizable conclusions about the graph.

Before that, we introduce two lemmas that will soon become handy. We start with a graph of n vertices, m edges, and total chip value N.

Lemma 3.1. *If a chip-firing game is infinite (i.e., the initial configuration never stabilizes), then every vertex fires infinitely many times during the process.*

Proof. In any infinite chip-firing game, there must exist at least one vertex v that fires infinitely many times. Suppose u is a neighbor of v. As v sends chips infinitely to u, since u cannot accumulate more than N chips, which is the total amount of chips in the game, u must be ready to fire after accumulating some finite amount of chips less than or equal to N. Otherwise, if u never fires, then all chips will eventually reach u, and the process terminates within finite steps. Now, since u fires after accumulating finitely many chips from v, and v sends infinitely many chips to u, u must also fire infinitely many times. Repeat this argument for all vertices in the connected graph. □

Lemma 3.2. *If a chip-firing game is finite, then there is at least one vertex in the graph that never fires in the process.*

Proof. We equivalently show that the game is infinite if every vertex has fired at least once. In this case, consider the vertex v that has not fired the most amount of time among all vertices. Then, every other vertex has fired since v last fired, and each of the neighbors of v has fired at least one chip to v. Then, v has at least $\deg(v)$ chips and is ready to fire. The same argument runs on indefinitely. □

We are now ready for the big theorem.

Theorem 3.3. *Again, let G be a graph with n vertices, m edges, and let any configuration of G have N chips in total.*

(1) If $N > 2m - n$, then the chip-firing process is infinite.
(2) If $m \leq N \leq 2m - n$, then there is guaranteed an initial configuration that will eventually terminate in a finite game as well as an initial configuration that will never terminate in an infinite game.
(3) If $N < m$, then the process is finite.

Proof. We should first understand that on the graph, $2m - n = \sum_{v \in V}(\deg(v) - 1)$. When $N > 2m - n$, after putting $\deg(v) - 1$ chips on each vertex v, there will be at least one vertex v_0 with chip value $\geq \deg(v_0)$. Then, there will always be at least one vertex ready to fire at all steps, and the game never terminates. Conversely, when $N \leq 2m - n$, there is always an initial configuration in which each vertex v has chip value $\leq \deg(v)$.

Now, in the case of $m \leq N \leq 2m - n$, we need to construct an initial configuration that never stabilizes. It suffices to show the case of $N = m$ since we can always add more chips to an unstable configuration and maintain its instability.

Consider an acyclic orientation of G, i.e., an assignment of orientation on the edges of G that does not contain a cycle. For each vertex v, let $\deg^+(v)$ denote the outdegree of v. Since there are m chips in total and $m = \sum_{v \in V} \deg^+(v)$, we can place $\deg^+(v)$ chips on each vertex v. We proceed to show that this configuration never stabilizes.

We know that any acyclic orientation must have a root. In this case, let the root r be where the firing first occurs in this orientation, with r having $\deg(r) = \deg^+(r)$ chips. After r fires, in the new configuration, we obtain a new acyclic orientation by reversing the direction of all edges connected to r. This decreases the outdegree of r by $\deg^+(r)$ (hence to 0) and increases the outdegree of each neighbor of r by 1. Now for all vertices, the number of chips at the vertex is again equal to its outdegree. Hence we may identify another root and repeat infinitely.

Finally, suppose $N < m$. As the chip-firing process begins, we record the pairing of each edge with the first chip that is fired across it. Since $N < m$, there must exist at least one edge that is not associated with any chip. Then, at least two vertices (incident to this edge) never fire. By Lemma 3.2, the game is finite. □

4. Criticality and Superstability

4.1. **Chip-firing Games Involving a Sink.** On top of our current construction of the chip-firing game, an important class of chip-firing processes involves distinguishing a particular vertex q as the *sink*. The sink is special in that it is not required to have a nonnegative chip value at any time. Intuitively, we may consider the sink vertex as a literal sink that continuously absorbs chips from the rest of the graph and flushes them out of the system, contributing to a net decrease in the total chip value in the process. Formally, for a graph G, we define a chip configuration involving a sink as an integer-valued vector $(c_1, c_2, ..., c_n, c_q) \in \mathbb{Z}^n$ such that $c_i \geq 0$ for all $i \neq q$. Similarly, the configuration is stable if c_i is strictly less than the degree of vertex i in G for all $i \neq q$.

In fact, there are two types of chip-firing processes involving a sink, and the difference lies in whether or not the sink is permitted to fire. We formally define the two types of processes as follows.

Definition 4.1. Let G be a graph of $n+1$ vertices with a distinguished vertex q as the sink. A chip-firing process *involving a sink that does not fire* behaves as follows. From an initial configuration $\mathbf{c}$, a non-sink vertex that is ready to fire fires at each step. The process terminates when a stable configuration is reached.

Definition 4.2. Let G be a graph of $n+1$ vertices with a distinguished vertex q as the sink. A chip-firing process *involving a sink that fires* behaves as follows. From an initial configuration $\mathbf{c}$, a non-sink vertex that is ready to fire fires at each step. When a stable configuration is reached, the sink fires regardless of its chip value.

We may immediately summarize several important differences between these two types of processes. With a monotonic decrease in chip values in a finite chip-firing game, the game in which the sink does not fire always eventually terminates and leads to a stabilization. We will soon prove that, in games in which the sink fires, any initial configuration will still eventually stabilize. However, we can easily see that this process will *never terminate*: firings from the sink under a stable configuration will eventually give its non-sink neighbors enough chips to be ready to fire again. We note here the important difference between the termination of a chip-firing process and the stabilization of an initial configuration, as the two are often assumed equivalent by a mathematician new to the subject. The termination of a chip-firing process simply means that all vertices have ceased to fire on the graph. Stabilization, on the other hand, refers to the condition in which no vertex on the graph is ready to fire. On graphs without a sink, the process indeed terminates

when it reaches a stabilization; by contrast, on graphs with a sink, we have defined the rules so that the sink is allowed to fire after a stable configuration is reached—this does not take away the fact that the configuration had been stable before the sink fired.

We now prove that in any chip-firing game involving a sink, a stabilization is guaranteed.

Proposition 4.3. *Let G be a graph with sink q. Any finite initial configuration of G will eventually stabilize whether or not q fires.*

Proof. In both types of processes, if the process does not stabilize, then any vertex other than q can only fire finitely many times as it has finitely many chips to begin with and its chip value decreases monotonically as it fires. Suppose that an initial configuration never stabilizes. Then, there is a vertex v that fires infinitely many times without the graph being stabilized. Then, all of its neighbors fire infinitely many times as well. In a connected graph, this implies that there must exist a directed path from v to q such that every vertex before q fires infinitely many times. Then, a neighbor of q fires infinitely many times, and we have a contradiction. □

Usually, the precise chip value at the sink does not matter. There is no restriction on its range of possible values, and the condition for whether or not the sink fires is unrelated to its chip value. Therefore, by convention, we normalize a chip-firing process involving a sink by making the graph's total chip value always equal to zero. This means that the value at the sink is always equal to the negative of the total chip value at all non-sink vertices, i.e., $c_q = -\sum_{i \neq q} c_i$. Intuitively, as mentioned earlier, the sink continuously flushes chips out of the rest of the system. However, this normalization demonstrates that the total amount of chips in the entire system is conserved when we consider the sink to be a part of the system.

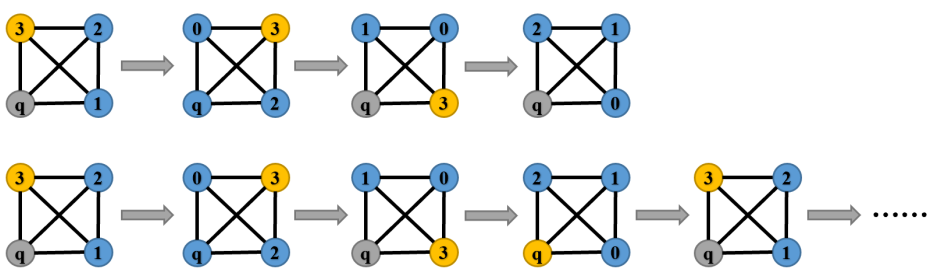

FIGURE 4. An example of a finite chip-firing process involving a sink. The sink is allowed to fire in the version below, but note that regardless of the status of the sink, the graph arrives at a stable configuration by the fourth step (by normalization, $c_q = -3 < \deg(q)$).

4.2. **Criticality.** This is an extremely important concept. For now, our definitions and theorems apply to chip-firing processes involving a sink.

Definition 4.4. A chip configuration $\mathbf{c}$ on a graph G with sink q is *critical* if it is stable and reachable from some finite component-wise larger configuration $\mathbf{b}$ in which every non-sink vertex is ready to fire. Equivalently, let L be the graph

Laplacian of G. Then $\mathbf{c}$ is critical if $\mathbf{c} = \text{stab}(\mathbf{b})$ for some configuration $\mathbf{b}$ such that $b_i \geq L_{ii}$ for any $i \in V$ and $i \neq q$.

In other words, the notion of criticality concerns stable configurations that actually appear when we run the chip-firing process from an arbitrarily large initial configuration. The key intuition here is that not all stable configurations are critical. Some stable configurations can never be generated from firings on the graph. For instance, the all-zero configuration $(0, 0, ..., 0)$ is stable but not critical on many graphs of a sufficiently large size. If it is reachable from a configuration with a total chip value greater than zero, then all chips in this configuration must eventually reach the sink. There are many graphs that cannot accomplish this. For example, we may easily check that no nonzero initial configuration on the K_4 graph with a sink (as in Figure 4) can reach the zero configuration. We only need to check a number of configurations of small sizes, since had any larger ones been able to reach the zero configuration, they would have needed to reduce to one of these smaller configurations before they can continue.

Since the definition of critical configurations rests on their generation from larger initial configurations, like the property of confluence, a natural follow-up question at this point is whether this critical configuration is uniquely reachable from another configuration. The answer is yes, but unique to each *firing class* of the graph. This is a good opportunity to introduce equivalence relations in the chip-firing game.

Definition 4.5. Let G be a graph with $|V| = n$. Let L be the graph Laplacian of G. Two chip configurations $\mathbf{c}$, $\mathbf{c}' \in \mathbb{Z}^n$ are *firing-equivalent*, i.e., $\mathbf{c} \sim \mathbf{c}'$, if $\mathbf{c} - \mathbf{c}' = L\mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^n$. The equivalence class of $\mathbf{c}$, i.e., $[\mathbf{c}]$, is commonly referred to as the *firing class* of $\mathbf{c}$.

In games that involve a sink, firing equivalence is defined in terms of the *reduced graph Laplacian* L_q, which is derived from the graph Laplacian by removing the row and column corresponding to q. Let $\mathbf{c}$, $\mathbf{c}' \in \mathbb{Z}^{n-1}$ be normalized configurations of G. Then $\mathbf{c} \sim \mathbf{c}'$ if $\mathbf{c} - \mathbf{c}' = L_q\mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^{n-1}$.

The three criteria for an equivalence relation follow immediately from this definition.

These above definitions offer us the tools to formally introduce the uniqueness of critical configurations per firing class.

Theorem 4.6. *For a graph G with a sink q, $|V| = n+1$ and reduced graph Laplacian L_q, there exists a unique critical configuration per firing class.*

We will need two lemmas before we can prove this theorem.

Lemma 4.7. *Let G be a graph with sink q, $|V| = n+1$, and reduced graph Laplacian L_q. Then there exists an integer-valued vector $\mathbf{z} \in \mathbb{Z}^n$ such that $L_q\mathbf{z} \geq 1$.*

Proof. Consider the vector $\mathbf{r} = (L_q)^{-1}\mathbf{1}$ with $\mathbf{1}$ being the all-ones vector. Since the inverse of a rational matrix is rational, each component of $\mathbf{r}$ is rational, so we write $r_i = a_i/b_i$ for all $1 \leq i \leq n$. Let $b = b_1b_2...b_n$. Then let $\mathbf{z} = b\mathbf{r}$. We have $L_q\mathbf{z} = L_qb(L_q)^{-1}\mathbf{1} \geq \mathbf{1}$. □

Lemma 4.8. *Let $\mathbf{c}$ and $\mathbf{c}'$ be two configurations of G with sink q and $|V| = n + 1$ such that $\mathbf{c} \sim \mathbf{c}'$, i.e., $\mathbf{c} - \mathbf{c}' = L_q\mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^n$. Then, there exists a configuration $\mathbf{b}$ that can legally fire to both $\mathbf{c}$ and $\mathbf{c}'$.*

Proof. The trick is to break $\mathbf{w}$ into a nonnegative part and a negative part. Define $I := \{1 \leq i \leq n \mid w_i \geq 0\}$ and $J := \{1 \leq j \leq n \mid w_j < 0\}$. Then

$$\mathbf{c} - \sum_J Lw_j e_j = \mathbf{c}' + \sum_I Lw_i e_i.$$

Define $\mathbf{b} := \mathbf{c} - \sum_J Lw_j e_j$. Then $\mathbf{b}$ legally reaches both $\mathbf{c}$ and $\mathbf{c}'$. □

We are ready to prove the big theorem.

Proof. (of Theorem 4.6) For existence, let $[\mathbf{c}]$ be a firing class on G. By Lemma 4.7, there exists $\mathbf{w} \in \mathbb{Z}^n$ such that $L_q\mathbf{w}$ is all positive. For any $t \in \mathbb{N}$, define $\mathbf{c_t} := \mathbf{c} + tL_q\mathbf{w}$. Since t is arbitrary, we can find a sufficiently large t such that $\mathbf{c_t}_i \geq \deg(i)$ for all $i \in V$. Then $\mathrm{stab}(\mathbf{c_t}) \in [c]$ (i.e., the stabilization of $\mathbf{c_t}$) is critical. Note that $\mathbf{c_t} \sim \mathbf{c}$, and $\mathrm{stab}(\mathbf{c_t}) \sim \mathbf{c_t}$, since we can always write a sequence of firings in terms of subtractions of the product of L_q and some integer-valued vector.

For uniqueness, suppose $\mathbf{c} \neq \mathbf{c}'$ are both critical and firing equivalent. By Lemma 4.8, there exists $\mathbf{b}$ that can legally reach both $\mathbf{c}$ and $\mathbf{c}'$. Hence $\mathbf{c}$ and $\mathbf{c}'$ are both stabilizations of $\mathbf{b}$. But by Theorem 2.3, $\mathrm{stab}(\mathbf{b})$ is unique, and we have a contradiction. □

Here, observe that we used Theorem 2.3 although it was not introduced under the context of the sink. Since Theorem 2.3 only concerns the behavior of the chip-firing process before or until it reaches a stabilization, whether or not the sink fires afterwards does not alter this result (remember that we defined that the sink is only allowed to fire once a stabilization is reached). Hence, we can use the theorem safely here, and the same idea applies to a number of other results we have so far proved.

4.3. **Superstability.** As the name itself implies, the condition of superstability describes configurations that are "more stable" among the collection of all stable configurations of a given graph. For any finite stable configuration, we can always destabilize it by adding chips to specific vertices (we call the subsequent chain of firings caused by this addition *avalanching*). Hence, for two stable (or, as we're often interested in, critical) configurations of the same graph, one is more stable than the other if it requires the addition of more chips to become destabilized.

For a graph G, the least stable among all of its stable configurations is the unique maximal stable configuration $\mathbf{c}_{\max}(G)$. Adding a single chip to any vertex will destabilize it. Hence, the chip value at each vertex is its degree minus 1. $\mathbf{c}_{\max}(G)$ uniquely exists for every graph G.

The most stable configurations of G are called *superstable* configurations.

Definition 4.9. A configuration $\mathbf{c}$ is *superstable* if there are no legal cluster firings of $\mathbf{c}$, i.e., for any nonempty collection of vertices $U \subseteq V$, $\mathbf{c} - L_G\lambda_U < \mathbf{0}$.

It is a fact that all superstable configurations are stable, i.e., if there are no legal cluster firings, then there are no legal individual firings. However, as Figure 3 has shown, not all legal cluster firings can correspond to a sequence of legal individual firings. The condition of a legal individual firing is stricter than the condition of a legal cluster firing.

With the definition of superstability, we now already have the tools to study some elementary properties of the structure of the set of superstable configurations of a graph G.

Proposition 4.10. *For a graph G, the set of superstable configurations is componentwise downward closed. That is, if $\mathbf{c}$ is a superstable configuration, and each component of a configuration $\mathbf{d}$ is less than or equal to the corresponding component of $\mathbf{c}$, then $\mathbf{d}$ is also superstable.*

Proof. Let L be the graph Laplacian for G. Suppose $\mathbf{c}$ is superstable, $\mathbf{d}$ is componentwise less than $\mathbf{c}$ but not superstable. Then there exists some nonempty collection of vertices U such that $\mathbf{d} - L\lambda_U \geq 0$. Then $\mathbf{c} - L\lambda_U \geq 0$. But $\mathbf{c}$ is superstable, so we have a contradiction. □

The inverse is true for critical configurations due to duality, which will be formally stated and proved in the next section.

Proposition 4.11. *The set of critical configurations is componentwise upward closed to $\mathbf{c}_{max}(G)$. That is, if $\mathbf{c}$ is a critical configuration, and configuration $\mathbf{b}$ satisfies $\mathbf{c}_i \leq \mathbf{b}_i \leq \mathbf{c}_{maxi}$ $\forall\, 1 \leq i \leq |V|$, then $\mathbf{b}$ is also a critical configuration.*

5. Duality

We now directly state the powerful duality theorem that will be used to complete the proof of Proposition 4.11.

Theorem 5.1. *For a graph G, a configuration $\mathbf{c}$ is superstable if and only if $\mathbf{c}_{max} - \mathbf{c}$ is critical.*

We will need to introduce several definitions and lemmas before we can prove this theorem.

5.1. Energy Minimizers.

The primary purpose of introducing energy minimizers here is to prove the theorem that there exists a unique superstable configuration per firing class, analogous to that of critical configurations. Nevertheless, the concept itself entails important ideas that are worth discussing. We observe that once we launch a chip-firing process, the initial configuration enters a momentum that drives toward a stabilization if it exists. This is analogous to, for example, an object with a high potential energy entering acceleration as it falls onto the earth, eventually finding an equilibrium under a state with a lower potential energy. From this perspective, we may similarly treat the chip-firing process as one in which a configuration in a state of "higher energy" moves towards a configuration of lower energy. Accordingly, each firing corresponds to a decrease in energy of the configuration.

As the name itself suggests, energy minimizers are solutions to the problem of *energy minimization*: given a configuration $\mathbf{c}$, find a configuration of the lowest energy that is firing equivalent to $\mathbf{c}$. In our finite, discrete space, the solution as the energy minimizer to this problem necessarily exists—think analogously of the well-ordering principle for $\mathbb{N}$. Formally, energy is defined as follows.

Definition 5.2. Let G be a graph with a sink q and Laplacian L_q. The *energy* of a configuration $\mathbf{c}$, $E(\mathbf{c})$, is defined as

$$E(\mathbf{c}) = \|{L_q}^{-1}\mathbf{c}\|^2,$$

where $\|\cdot\|$ is the Euclidean norm.

As mentioned above, we will prove that there is a unique superstable configuration per firing class. Using energy minimizers, the goal is to break this statement into two parts: 1) there is a unique energy minimizer per firing class, and 2) the conditions of a configuration being superstable and an energy minimizer are equivalent. We will need two lemmas before we can prove these two theorems.

For notation purposes, for any vector $\mathbf{w} \in \mathbb{Z}^n$, we define $\mathbf{w}^+ \in \mathbb{Z}^n_{\geq 0}$ by

$$w_i{}^+ = \begin{cases} z_i & \text{if } z_i \geq 0 \\ 0 & \text{if otherwise.} \end{cases}$$

Conversely, we define $\mathbf{w}^-$ by zeroing all positive components of $\mathbf{w}$.

Lemma 5.3. *Let G be a graph with a sink q and Laplacian L_q. Suppose that for two nonnegative configurations $\mathbf{c}$, $\mathbf{c}'$, there exists $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{c}' = \mathbf{c} - L_q\mathbf{w}$. Then the configuration $\mathbf{d}$ obtained via only firings implied by positive entries of $\mathbf{w}$ is also nonnegative. More compactly,*

$$\mathbf{d} = \mathbf{c} - L_q\mathbf{w}^+ \geq \mathbf{0}.$$

Proof. First, note that $L_q\mathbf{w} = L_q\mathbf{w}^+ + L_q\mathbf{w}^-$. When $\mathbf{w}_i{}^+ = 0$, $-(L_q\mathbf{w}^+)_i \geq 0$, so $\mathbf{d}_i \geq \mathbf{c}_i \geq 0$. When $\mathbf{w}_i{}^+ > 0$, it is automatically true that $\mathbf{w}_i{}^- = 0$, so $(L_q\mathbf{w}^-)_i \geq 0$. Equivalently, $(L_q\mathbf{w})_i \geq (L_q\mathbf{w}^+)_i$, and so

$$\mathbf{d}_i - (L_q\mathbf{w}^+)_i \geq \mathbf{d}_i - (L_q\mathbf{w})_i \geq \mathbf{c}_i - (L_q\mathbf{w})_i = \mathbf{c}'_i \geq 0.$$

So $\mathbf{d}_i \geq 0$. □

Lemma 5.4. *Let G be a graph with a sink q and Laplacian L_q. Suppose that for two configurations $\mathbf{c}$, $\mathbf{c}'$, there exists $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{c}' = \mathbf{c} - L_q\mathbf{w}$. Then*

$$\begin{aligned} E(\mathbf{c}') &= E(\mathbf{c}) + \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T L_q{}^{-1}\mathbf{c} \\ &= E(\mathbf{c}) - \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T L_q{}^{-1}\mathbf{c}' \end{aligned}$$

Proof. By direct computation,

$$\begin{aligned} E(\mathbf{c}') &= \|L_q{}^{-1}(\mathbf{c} - L_q\mathbf{w})\|^2 \\ &= \|L_q{}^{-1}\mathbf{c} - \mathbf{w}\|^2 \\ &= \|L_q{}^{-1}\mathbf{c}\|^2 - 2\mathbf{w}^T L_q{}^{-1}\mathbf{c} + \mathbf{w}^T\mathbf{w} \\ &= E(\mathbf{c}) + \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T L_q{}^{-1}\mathbf{c} \\ &= E(\mathbf{c}) - \mathbf{w}^T\mathbf{w} - 2(\mathbf{w}^T L_q{}^{-1}\mathbf{c} - \mathbf{w}^T\mathbf{w}) \\ &= E(\mathbf{c}) - \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T L_q{}^{-1}\mathbf{c}'. \end{aligned}$$

□

Theorem 5.5. *Let G be a graph with a sink q and Laplacian L_q. There exists a unique energy minimizer for every firing class $[\mathbf{c}]$ of G.*

Proof. The existence of such an energy minimizer is again by the well-ordering principle, as we can always find a bijection between the set of positive integers and the set of possible configurations in the class of $[\mathbf{c}]$. For uniqueness, we need to show that for any configuration $\mathbf{c}$, there cannot exist two distinct energy minimizers both equivalent to $\mathbf{c}$. Suppose such $\mathbf{c_1}$, $\mathbf{c_2} \geq \mathbf{0}$ exist. Since $\mathbf{c_1} \sim \mathbf{c_2}$, there exists

some $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{c_2} = \mathbf{c_1} - L_q\mathbf{w}$. By Lemma 5.3, the configuration $\mathbf{d} = \mathbf{c_1} - L_q\mathbf{w}^+ \geq 0$. By Lemma 5.4,

$$E(\mathbf{d}) = E(\mathbf{c_1}) - (\mathbf{w}^+)^T\mathbf{w}^+ - 2(\mathbf{w}^+)^T {L_q}^{-1}\mathbf{d}.$$

Here, we will use a fact without proving it: ${L_q}^{-1} > 0$ at every entry. This statement follows from a property of the graph Laplacian L_q itself that it is a *non-singular M-matrix*. This definition arises from the study of avalanche finite matrices, a topic beyond the scope of this paper. A detailed introduction on avalanche finite systems can be found at Chapter 6.1 of Klivans.

Returning to the proof, since ${L_q}^{-1} > 0$ entry-wise and $\mathbf{d} \geq 0$, it follows that $2(\mathbf{w}^+)^T L_q\mathbf{d} \geq 0$. Then, $E(\mathbf{d}) \leq E(\mathbf{c_1}) - (\mathbf{w}^+)^T\mathbf{w}^+$. But since $\mathbf{c_1}$ is an energy minimizer and $\mathbf{d} \sim \mathbf{c_1}$, it must be true that $E(\mathbf{d}) \geq E(\mathbf{c_1})$, and hence $\mathbf{w}^+ = 0$. Then $\mathbf{w} \leq 0$.

By the same method, we may also obtain that

$$E(\mathbf{c_1}) = E(\mathbf{c_2}) + (\mathbf{w}^+)^T\mathbf{w}^+ - 2(\mathbf{w}^+)^T {L_q}^{-1}\mathbf{c_1}.$$

Since $\mathbf{w} \leq 0$, we have $E(\mathbf{c_2}) \leq E(\mathbf{c_1})$, and that $E(\mathbf{c_2}) = E(\mathbf{c_1})$ if and only if $\mathbf{w} = 0$. Since $\mathbf{c_1}$ is an energy minimizer, it must be the case that $\mathbf{w} = 0$. So $\mathbf{c_1} = \mathbf{c_2}$. □

Theorem 5.6. *Let G be a graph with a sink q and Laplacian L_q. A configuration $\mathbf{c}$ is superstable if and only if it is an energy minimizer.*

Proof. Suppose $\mathbf{c}$ is superstable. We proceed to show that it is an energy minimizer. Let $\mathbf{c}' \geq 0$ be equivalent to $\mathbf{c}$. Then there exists $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{c}' = \mathbf{c} - L_q\mathbf{w}$. Then, by Lemma 5.3, the configuration $\mathbf{d} = \mathbf{c} - L_q\mathbf{w}^+ \geq 0$. By the definition of superstability of $\mathbf{c}$, it must be the case that $\mathbf{w}^+ = 0$. Hence, $\mathbf{w} \leq 0$. Also, by Lemma 5.4,

$$E(\mathbf{c}') = E(\mathbf{c}) + \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T {L_q}^{-1}\mathbf{c}.$$

And so $E(\mathbf{c}') \geq E(\mathbf{c})$.

Now suppose $\mathbf{c}$ is an energy minimizer. Assume $\mathbf{c}$ is not superstable. Then, there exists $\mathbf{w} > 0$ (not zero at every component) such that $\mathbf{c}' = \mathbf{c} - L_q\mathbf{w} \geq 0$. Again, we write down the equation

$$E(\mathbf{c}') = E(\mathbf{c}) - \mathbf{w}^T\mathbf{w} - 2\mathbf{w}^T {L_q}^{-1}\mathbf{c}'$$

and we get $E(\mathbf{c}') \leq E(\mathbf{c}) - \mathbf{w}^T\mathbf{w} < E(\mathbf{c})$. But $\mathbf{c}$ is an energy minimizer, and we have a contradiction. □

5.2. **Proof of Duality.** We first prove a lemma.

Lemma 5.7. *Let G be a graph with Laplacian L. Let $\mathbf{b}$ be an unstable configuration that reaches a stable configuration $\mathbf{c}$ via*

$$\mathbf{c} = \mathbf{b} - \sum_{j=1}^{k} Le_{i_j}$$

for some nonempty collection of vertices $U = \{i_1, i_2, ..., i_k\} \subseteq V$. Then, for every vertex v unstable in $\mathbf{b}$, there exists some $j \in [k]$ such that $i_j = v$.

Proof. Assume that $i_j \neq v$ for all j. Then, consider $\sum_{j=1}^{k}(Le_{i_j})_v = \sum_{j=1}^{k} L_{i_j,v}$. By our assumption, each component of this vector lies on a non-diagonal entry of L and can only take the value of 0 or -1 and is thus non-positive. So $\sum_{j=1}^{k}(Le_{i_j})_v \leq \mathbf{0}$. Since also v is unstable in $\mathbf{b}$, we have $\mathbf{c}_v \geq \mathbf{b}_v \geq L_{vv}$, and v is unstable in $\mathbf{c}$. But $\mathbf{c}$ is stable, so we arrive at a contradiction. □

We are now ready to prove the theorem of duality.

Proof. (of Theorem 5.1) Suppose $\mathbf{c}$ is superstable. By definition, $\mathbf{c}_{\max}$ is the unique maximal stable configuration of G. By Proposition 4.11, $\mathbf{c}_{\max} - \mathbf{c}$ is stable. By Lemma 4.7, there exists some vector $\mathbf{z} > 0$ sufficiently large such that, for any i, $(L\mathbf{z})_i \geq L_{ii}$, so $(\mathbf{c}_{\max} - \mathbf{c} + L\mathbf{z})_i \geq L_{ii}$. Set $\mathbf{b} = \mathbf{c}_{\max} - \mathbf{c} + L\mathbf{z}$. Then $b_i \geq L_{ii}\ \forall\ i$. To finish proving that $\mathbf{c}_{\max} - \mathbf{c}$ is critical, the goal now is to show that there exists some permutation of matrix operations as a sequence of legal firings from $\mathbf{b}$ to $\mathbf{c}_{\max} - \mathbf{c}$.

As $\mathbf{z} > 0$, we can rewrite $\mathbf{z} = \sum_{j=1}^{k} e_{i_j}$. Then

$$\mathbf{c}_{\max} - \mathbf{c} = \mathbf{b} - \sum_{j=1}^{k} Le_{i_j}.$$

We proceed to construct a permutation σ of the k steps such that, for any $1 \leq l \leq k$, if we define $\mathbf{b_l} = \mathbf{b} - \sum_{j=1}^{l} e_{i_{\sigma(j)}}$, then

$$(\mathbf{b_l})_{i_{\sigma(l+1)}} \geq L_{i_{\sigma(l+1)} i_{\sigma(l+1)}}. \tag{5.8}$$

We will construct σ inductively. For the base case $\sigma(1)$, we have $j = l$, so $(\mathbf{b_l})_j \geq L_{jj}$ on a single vertex j. Suppose we have $\sigma(1)$, ... $\sigma(m)$ for some $m < k$ such that Equation 5.8 holds for all $1 \leq l \leq m-1$. Then, when $l = m$, we have

$$\mathbf{b_m} = \mathbf{c}_{\max} - \mathbf{c} + L\mathbf{z}', \tag{5.9}$$

where

$$\mathbf{z}' = \sum_{j=1}^{k} Le_{i_j} - \sum_{j=1}^{m} Le_{i_{\sigma(j)}} > 0.$$

Equivalently,

$$\mathbf{c}_{\max} - \mathbf{c} = \mathbf{b_m} - \left(\sum_{j=1}^{k} Le_{i_j} - \sum_{j=1}^{m} Le_{i_{\sigma(j)}} \right).$$

Since $\mathbf{c}$ is superstable, rewriting Equation 5.9 as $\mathbf{c} - L\mathbf{z}' = \mathbf{c}_{\max} - \mathbf{b_m}$, by the definition of superstability, there exists a vertex v such that $(\mathbf{c}_{\max} - \mathbf{b_m})_v < 0$. Equivalently, $(\mathbf{b_m})_v \geq L_{vv}$, as any chip value that exceeds the maximal stable configuration necessarily makes the corresponding vertex ready to fire. Hence, $\mathbf{b_m}$ is unstable. Then, by Lemma 5.7, there exists a permutation $1 \leq \sigma(m+1) \leq k$ distinguished from all of $\sigma(1)$, ..., $\sigma(m)$ such that $i_{\sigma(m+1)} = v$. We have shown that a permutation σ exists for all of the k steps, hence the existence of a legal sequence of firings from $\mathbf{b}$ to $\mathbf{c}_{\max} - \mathbf{c}$.

Note that we only need to show one direction to complete this proof. By Theorem 4.6, Theorem 5.5, and Theorem 5.6, critical configurations and superstable configurations are unique per firing class. Since we now know that the dual of a superstable configuration is critical, the reverse direction follows immediately. □

6. Conclusion

This paper has briefly introduced the setup of the chip-firing game and several of its fundamental properties including confluence and the duality between critical and stable configurations. It also offers a taste at some of the most important problems in this subject. The concept of self-organized criticality concerns the extent to which an external impetus changes the behavior of a dynamic system. In our subject, distinguishing superstable configurations from the collection of all stable configurations entails this idea.

Due to the short scope of this project, there remain several fundamental notions yet to be discussed, such as efficiency, structures of critical and superstable configurations as sets, and Dhar's Burning Algorithm, an effective way of determining whether a configuration is superstable on a given graph. These concepts prepare the reader for further studies in the chip-firing game. A natural next-step is to study the bijection between the set of superstable and critical configurations and the set of spanning trees on a given graph. This is based on the famous theorem that, on a graph G, *the number of critical configurations of G equals the number of spanning trees of G*. The study of chip-firing games in terms of spanning trees is where the second combinatoric perspective mentioned in the introduction really begins to manifest itself. The proof of the theorem for the bijection between critical configurations and spanning trees is largely based on matroid theory, including a generalization of linear algebraic concepts to a combinatorial setting.

In the broader picture, the perspective of abelian sandpile models mentioned in the introduction is indeed algebraic, revolving around the study of *sandpile groups*. At this point, we will already have the tools to generate geometric wonders from particular types of graphs such as the square grid graph. For example, with a sufficiently large chip size, the identity element of a sandpile group on such a graph generates fractal-like images like Figure 5. Eventually, of course, we will also eliminate the parameters set at the beginning of this paper and enter the study of infinite graphs and graphs with loops and multiple edges. Until then, fire some chips on graphs of your own and have fun.

Acknowledgments

I would like to thank my mentor Phillip Lo for guiding and helping me through the entire research process. From content selection, discussion of materials and secondary sources to the weeks of feedback and revisions that often crossed enormous time zone differences and tricky schedule-related barriers, this paper would not have been a feasible task without Phillip's help. This is the first ever research project I've undertaken for which I am the sole author. Through the bumpy beginnings and some unnecessarily frequent moments of procrastination, I learned and grew, slowly, as a student of mathematics, researcher, writer, and thinker.

I would also like to thank Professor May for organizing the mathematics REU program, without which all of this would not have existed. Although I only had the opportunity to participate in the program for five weeks in person, I truly benefitted from the variety of lectures and talks in terms of subject area and level of difficulty that the program has provided. I sat through many lectures that had virtually no connection to my research topic (as far as I can comprehend) and which I barely understood, but I kept returning to Ryerson because of the possibility of my mind clicking with a concept or two, the undescribable but immanent joy from

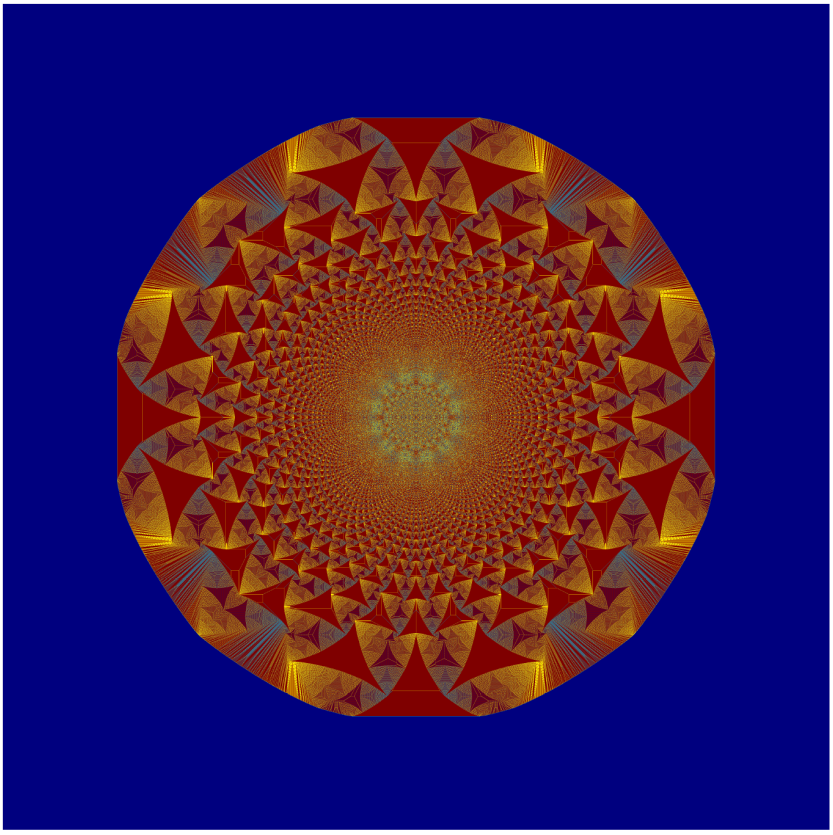

FIGURE 5. This fractal-like image is generated via the stabilization of about 20 million chips on a grid graph. We color every vertex by its chip value in the final configuration: 0's are in dark blue, 1's are in light blue, 2's are in yellow, 3's are in brown. Image adapted from Chapter 1 of Klivans.

which lighting up my whole day. Personally, listening to talks perhaps beyond my current ability has also allowed me to have a taste of the connection of the different fields of mathematics at a higher and more abstract level beyond their rigid classification that I encounter much more commonly at my level of studies. In this regard, I must also thank Professor Rudenko for the awesome five weeks of lectures at the apprentice program. The breadth of topics Professor Rudenko has covered provided me with a solid combinatorial background to write this paper and, more importantly, to take further math courses in subjects such as algebra, geometry, and algebraic geometry.

Finally, I would like to thank Tony Yang and Bill Lai for offering me a place to stay when my rent got cancelled at the last minute. Their charity might not necessarily be returned but will surely be heartfeltly remembered.

References

[1] Bak, Per and Tang, Chao and Wiesenfeld, Kurt. Self-organized Criticality: An Explanation of 1/f Noise. Physical Review Letters 59, 381-384. 1987. 10.1103/PhysRevLett.59.381.

[2] Anders Bjorner, Laszlo Lovasz, Peter W. Shor. Chip-firing Games on Graphs. European Journal of Combinatorics 12, 283-291. 1991. https://doi.org/10.1016/S0195-6698(13)80111-4.

[3] Caroline J. Klivans. The Mathematics of Chip-firing. https://www.dam.brown.edu/people/cklivans/Chip-Firing.pdf.

[4] J. Spencer. Balancing Vectors in the Max Norm. Combinatorica 6, 55-65. 1986. https://doi.org/10.1007/BF02579409.