Séminaire Lotharingien de Combinatoire **78B** (2017)
Article #30, 12 pp.

Proceedings of the 29th Conference on Formal Power Series and Algebraic Combinatorics (London)

Sorting via chip-firing

Sam Hopkins[1], Thomas McConville[1] and James Propp[2]

[1]*Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA*
[2]*Department of Mathematical Sciences, University of Massachusetts, Lowell, MA*

Abstract. We investigate a variant of the chip-firing process on the infinite path graph $\mathbb{Z}$: rather than treating the chips as indistinguishable, we label them with positive integers. To fire an unstable vertex, i.e. a vertex with more than one chip, we choose any two chips at that vertex and move the lesser-labeled chip to the left and the greater-labeled chip to the right. This labeled version of the chip-firing process exhibits a remarkable confluence property, similar to but subtler than the confluence that prevails for unlabeled chip-firing: when all chips start at the origin and the number of chips is even, the chips always end up in sorted order. Our proof of sorting relies upon an independently interesting lemma concerning unlabeled chip-firing which says that stabilization preserves a natural partial order on configurations. We also discuss extensions to other variants of the infinite path, an intriguing empirical observation on random firing of labeled chips, and a possible generalization to other types.

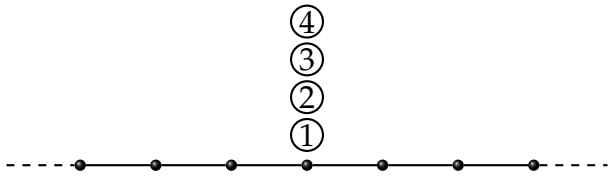
Keywords: chip-firing, abelian sandpile model, confluence

1 Introduction

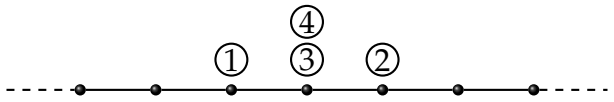
We introduce a labeled version of the chip-firing process. The chip-firing process is a discrete dynamical system that takes place on a graph. The name "chip-firing" was coined by Björner, Lovász, and Shor [3], but in fact this process is essentially the same as the abelian sandpile model (ASM) introduced by Bak, Tang, and Wiesenfeld [2] and further developed by Dhar [7, 8]. (See also the work of Engel [9].) Here "abelian" means that the order in which certain local moves are carried out has no effect on the final state. This property is also often called "confluence" in the context of abstract rewriting systems. Dhar views the ASM as a prototype for networks of communicating processors that achieve predictable results despite a lack of global synchronization; this point of view has been developed by Levine and his coauthors [4, 5, 6, 10], who introduced a broad family of such networks, called "abelian networks," to which the ASM belongs.

Björner, Lovász, and Shor were directly inspired by papers of Spencer [12] and Anderson et al. [1] that investigated chip-firing in the special case of the infinite path graph $\mathbb{Z}$ (which has vertex set $\mathbb{Z}$ with i and j joined by an edge whenever $|i-j|=1$). In fact, it is in exactly this special case of the infinite path graph $\mathbb{Z}$ that we introduce labeled chip-firing. Here is how the labeled chip-firing process on $\mathbb{Z}$ works: we start with n labeled

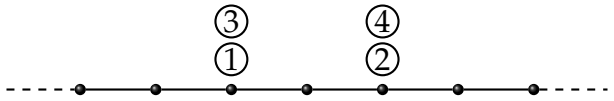
chips $(1), (2), \ldots, (n)$ at the origin; at each step we choose any two chips (a) and (b) with $a < b$ that occupy the same vertex i and *fire* these chips together, moving (a) to vertex $i-1$ and (b) to vertex $i+1$; we keep carrying out firings until no chips can fire. For example, suppose $n = 4$, so that we start from the following configuration (where we draw $\mathbb{Z}$ in the plane as a number line in the usual way, with $i-1$ to the left of i):



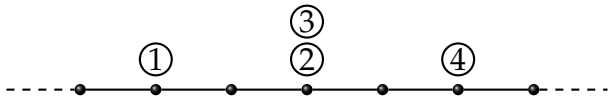
By firing chips (1) and (2) we reach the following configuration:



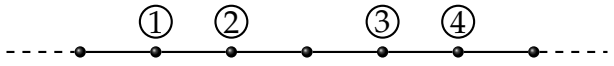
Then by firing (3) and (4) together we reach this configuration:



After firing (1) and (3), and then firing (2) and (4), in two more steps we reach the following configuration:



Finally, by firing (2) and (3) we reach the following *stable* configuration, which has no more possible firings:



In this process we made several arbitrary choices of which pairs of chips to fire. As it turns out, no matter what choices we made we would have always reached this same stable configuration where the chips appear in sorted order from left to right. This is a confluence result which says that the divergent paths our process can take must at some later point come back together. It is well known that confluence holds for the unlabeled chip-firing process (on any graph and for any configuration with sufficiently few chips to guarantee that stabilization is possible at all): this is one of the basic results of [3]. But confluence for the labeled chip-firing process is much subtler. Indeed, not all initial configurations are confluent in the labeled chip-firing process. This means in particular that the theory of abelian networks does not automatically apply to our situation. To see

that not all initial configurations are confluent, consider the configuration that has three chips at the origin rather than four. We can fire (1) and (2), or fire (1) and (3), or fire (2) and (3); in all three cases, the result is a stable configuration, but the labeled chips end up at different vertices, so confluence does not hold in this case. More generally, if we start with an *odd* number of labeled chips at 0, we can make sure that any preselected chip never moves away from 0, so confluence does not hold. Our main result, proved in the next section, says that the labeled chip-firing process on $\mathbb{Z}$ is confluent, and in particular sorts the chips, as long as the number of chips is *even*.

2 Main result

One of the main ways we will understand this labeled chip-firing process is by relating it to the usual unlabeled chip-firing process. Therefore, we first review unlabeled chip-firing on the graph $\mathbb{Z}$. A *configuration of unlabeled chips* on $\mathbb{Z}$ is some assignment of a finite number of indistinguishable chips to the vertices of $\mathbb{Z}$. All configurations, both labeled and unlabeled, will be on $\mathbb{Z}$ in what follows in this section. We use lowercase letters for unlabeled configurations. Formally, we treat an unlabeled configuration c as a function $c\colon \mathbb{Z} \to \mathbb{N}$ with $\sum_i c(i) < \infty$ and we think of c as having $c(i)$ chips at i. We use $\mathrm{supp}(c)$ to denote the *support* of c, i.e., $\mathrm{supp}(c) := \{i \in \mathbb{Z}\colon c(i) \geq 1\}$. We write $\max(c) := \max(\mathrm{supp}(c))$ and $\min(c) := \min(\mathrm{supp}(c))$. As is customary, we use the convention $\max(\varnothing) := -\infty$ and $\min(\varnothing) := \infty$. If c is a configuration and $i \in \mathbb{Z}$ is some vertex such that c has at least two chips at i, we may perform a *chip-firing move* at i, which moves one chip at i leftward one vertex and one chip at i rightward one vertex. If the resulting configuration is c' then in this case we also say that c' is obtained from c by *firing at vertex* i. We write $c \to d$ to mean that d is obtained from c by some sequence of (zero or more) chip-firing moves. We say c is *stable* if we cannot perform any chip-firing moves on c, that is, if $c \to d$ implies that $c = d$. Of course, the map $c \mapsto \mathrm{supp}(c)$ is a bijection between stable configurations of n chips and subsets of $\mathbb{Z}$ of size n.

The following confluence property of the chip-firing process is well known; for instance, it can be deduced from the main results of [3]. It was also proved in a special case (Lemma 3 below) by Anderson et al. [1], but the proof in general is more or less the same as in that special case. At any rate, it is essentially a consequence of Newman's lemma [11] (a.k.a. the diamond lemma) which says that "local confluence" is enough to imply confluence for terminating systems.

Lemma 1. *For any configuration c, there is a unique stable d with $c \to d$.*

We use $\widetilde{c}$ to denote the *stabilization* of c, i.e. the unique stable d with $c \to d$ guaranteed by Lemma 1. A fact (which follows immediately from Lemma 1) that we will use over and over again is that if $c \to d$ then $\widetilde{d} = \widetilde{c}$.

Let us introduce some notation for specific configurations of unlabeled chips. For two configurations c and d, let us use $c + d$ to denote their *sum*, i.e., the configuration with $(c + d)(i) = c(i) + d(i)$ for all $i \in \mathbb{Z}$. It is clear that if $c \to c'$ then $c + d \to c' + d$. For $n \in \mathbb{N}$ and a configuration c, we use the shorthand $nc := \overbrace{c + c + \cdots c}^{n \text{ terms}}$. For $i \in \mathbb{Z}$ we let δ_i denote the configuration that has a single chip at i and no other chips; in other words, δ_i is the unique stable configuration with $\mathrm{supp}(\delta_i) = \{i\}$. For $i, j \in \mathbb{Z}$, we let $\delta_{[i,j]}$ denote the configuration that has one chip at vertex k for all $i \leq k \leq j$ and no other chips; in other words $\delta_{[i,j]}$ is the unique stable configuration with $\mathrm{supp}(\delta_{[i,j]}) = [i, j]$. (We use the convention that if $i > j$ then $[i, j] = \varnothing$.) Note that $\delta_i = \delta_{[i,i]}$.

We now describe some formulas for specific stabilizations that will be needed later.

Proposition 2. *Suppose that $c = \delta_{[a+1,b-1]} + \delta_i$, where $a, b, i \in \mathbb{Z}$ satisfy $a < b$, $a \leq i$, and $i \leq b$. Then we have $\widetilde{c} = \delta_{[a,a+b-i-1]} + \delta_{[a+b-i+1,b]}$.*

Proof. We prove this by induction on $b - a$. If $b - a = 1$ the proposition is clear because then $c = \widetilde{c} = \delta_i$. So assume $b - a > 1$ and the result is known for smaller values of $b - a$. If $i = a$ or $i = b$ the proposition is also clear because then $c = \widetilde{c}$. So assume that $a < i < b$. Set $c' := \delta_{[a+1,i-1]} + \delta_{i-1}$ and $c'' := \delta_{i+1} + \delta_{[i+1,b-1]}$. By firing at vertex i we see that $c \to c' + c''$. Applying the inductive hypothesis gives

$$\widetilde{c'} = \delta_a + \delta_{[a+2,i]},$$
$$\widetilde{c''} = \delta_{[i,b-2]} + \delta_b.$$

So $c \to \delta_a + \delta_{[a+2,i]} + \delta_{[i,b-2]} + \delta_b = \delta_a + c''' + \delta_b$ where $c''' := \delta_{[a+2,b-2]} + \delta_i$. Applying the inductive hypothesis again gives

$$\widetilde{c'''} = \delta_{[a+1,a+b-i-1]} + \delta_{[a+b-i+1,b-1]},$$

so that $c \to \delta_{[a,a+b-i-1]} + \delta_{[a+b-i+1,b]}$. But $\delta_{[a,a+b-i-1]} + \delta_{[a+b-i+1,b]}$ is stable, which means we must have $\widetilde{c} = \delta_{[a,a+b-i-1]} + \delta_{[a+b-i+1,b]}$. □

The unlabeled configuration we are most interested in is $n\delta_0$. The following description of the stabilization of $n\delta_0$ is also well known, appearing for instance in the original paper of Anderson et al. [1]. For completeness we provide a short proof of this lemma using the previous proposition.

Lemma 3. *For all $n \geq 1$ we have,*

$$\widetilde{n\delta_0} = \begin{cases} \delta_{[-m,-1]} + \delta_{[1,m]} & \text{if } n = 2m \text{ is even;} \\ \delta_{[-m,m]} & \text{if } n = 2m + 1 \text{ is odd.} \end{cases}$$

Proof. The proof is by induction on n. The case $n = 1$ is clear; so suppose $n > 1$ and the result is known for $n-1$. Set $c := \widetilde{(n-1)\delta_0}$. We have $\widetilde{n\delta_0} = \widetilde{c + \delta_0}$. If $n = 2m$ is even then by induction $\widetilde{c} = \delta_{[-(m-1),m-1]}$, so $\widetilde{c+\delta_0} = \delta_{[-m,-1]} + \delta_{[1,m]}$ by Proposition 2. If $n = 2m+1$ is odd, then by induction we have $\widetilde{c} = \delta_{[-m,-1]} + \delta_{[1,m]}$ and so clearly we have $\widetilde{c+\delta_0} = c = [-m, m]$. □

Now let us describe labeled chip-firing. A *labeled configuration of chips* on $\mathbb{Z}$ is some assignment of a finite number of distinguishable chips, labeled by positive integers, to the vertices of $\mathbb{Z}$. We use uppercase calligraphic script for labeled configurations and use (i) to denote the chip labeled i. Formally, we treat a labeled configuration $\mathcal{C}$ as a function $\mathcal{C}\colon X \to \mathbb{Z}$ for some $X \subseteq \mathbb{Z}_{>0}$, and we think of chip (i) as being at the vertex $\mathcal{C}(i)$ in $\mathcal{C}$ for all $i \in X$. Normally we will take $X = [n]$ and thus study labeled configurations of the n chips $(1), (2), \ldots, (n)$. If $a < b$ and chips (a) and (b) are at the same vertex in $\mathcal{C}$, we may *fire* (a) and (b) together in $\mathcal{C}$ by moving (a) leftward one vertex and (b) rightward one vertex. (The important point is that **chips with lesser labels move leftward**.) We write $\mathcal{C} \to \mathcal{D}$ to mean that $\mathcal{D}$ is obtained from $\mathcal{C}$ by a sequence of labeled chip-firing moves of this form. If $\mathcal{C}$ is a labeled configuration we use $[\mathcal{C}]$ to denote the underlying unlabeled configuration: thus $[\mathcal{C}](i) := \#\mathcal{C}^{-1}(i)$ for all $i \in \mathbb{Z}$. We say that $\mathcal{D}$ is *stable* if $[\mathcal{D}]$ is stable. As mentioned, our strategy in understanding labeled chip-firing will be to relate it to unlabeled chip-firing. To that end, here are some very basic facts relating labeled and unlabeled chip-firing, which we will use without even citing specifically from now on.

Proposition 4.

- *If $\mathcal{C} \to \mathcal{D}$ then $[\mathcal{C}] \to [\mathcal{D}]$.*
- *If $c \to d$ and $c = [\mathcal{C}]$, then there exists $\mathcal{D}$ with $\mathcal{C} \to \mathcal{D}$ such that $d = [\mathcal{D}]$.*

Consequently, for any $\mathcal{C}$ there is some stable $\mathcal{D}$ with $\mathcal{C} \to \mathcal{D}$, and we have $[\mathcal{D}] = \widetilde{[\mathcal{C}]}$.

There need not be a unique stable $\mathcal{D}$ with $\mathcal{C} \to \mathcal{D}$: the previous proposition only determines $[\mathcal{D}]$ but not the way that the chips are labeled in $\mathcal{D}$. Nevertheless we are interested in cases where we do have a unique labeled stabilization. In particular, we will consider the labeled analog of $n\delta_0$, which has chips $(1), (2), \ldots, (n)$ at vertex 0 and no other chips; we denote this configuration by Δ^n. In other words, $\Delta^n(i) := 0$ for all $i \in [n]$. Of course, $[\Delta^n] = n\delta_0$. Note, as mentioned in the introduction, that Δ^3 already does not have a unique stabilization. On the other hand, our main result is that when n is even, Δ^n does have a unique stabilization.

First let us observe that there is a useful global symmetry in this labeled chip-firing process when we start from the configuration Δ^n. If $\mathcal{C}$ is a configuration of n labeled chips, define its *dual* $\mathcal{C}^*$ as follows: first reflect $\mathcal{C}$ horizontally about the vertex 0, then replace chip (i) by chip $(n+1-i)$ for all $1 \le i \le n$. Of course, $(\mathcal{C}^*)^* = \mathcal{C}$.

Lemma 5. *We have $\Delta^n \to \mathcal{C}$ if and only if $\Delta^n \to \mathcal{C}^*$.*

Proof. It is easy to see that the duality operation respects labeled chip-firing moves, meaning that if $\mathcal{D}$ is obtained from $\mathcal{C}$ by a labeled chip-firing move then $\mathcal{D}^*$ is obtained from $\mathcal{C}^*$ by a labeled chip-firing move. The lemma then follows since $(\Delta^n)^* = \Delta^n$. □

Very roughly speaking, to prove confluence of the labeled chip-firing process we study how far we can move chips via chip-firing. The following is obvious but important.

Proposition 6. *If $c \to d$ then* $\min(\widetilde{c}) \leq \min(d)$.

Proof. Each chip-firing move preserves or decreases the minimum occupied vertex, so we have $\min(d') \leq \min(d)$ for any $d \to d'$. Thus in particular we have $\min(\widetilde{d}) \leq \min(d)$. But if $c \to d$, then $\widetilde{d} = \widetilde{c}$. □

Applying Proposition 6 to our situation of interest tells us that if $\Delta^n \to \mathcal{C}$ then we have $\min([\mathcal{C}]) \geq -\lfloor n/2 \rfloor$ and, by Lemma 5, $\max([\mathcal{C}]) \leq \lfloor n/2 \rfloor$. This puts some constraint on the movement of chips during the labeled chip-firing process, but is not really so useful because it says nothing about the position of chips with particular labels. We want to strengthen this conclusion about how far chips can move to take into account chip labels.

First let us define some notion for restricting labeled configurations to a subset of chips. For a labeled configuration $\mathcal{C}$ with label set X and $Y \subseteq \mathbb{Z}_{>0}$, we use $\mathcal{C} \setminus Y$ to denote the restriction of $\mathcal{C}$ to the chips with labels in $X \setminus Y$. For any labeled configuration $\mathcal{C}$ and any $k \in \mathbb{N}$, we use the shorthand $\mathcal{C}|_{\geq k} := \mathcal{C} \setminus \{1, 2, \ldots, k-1\}$. We want some way to describe how the largest-labeled chips evolve in the labeled chip-firing process. So let us say that an unlabeled configuration d is *rightward-reachable* from an unlabeled configuration c, written $c \xrightarrow{R} d$, if d is obtained from c by a sequence of (zero or more) moves of the forms:

- perform a chip-firing move;
- move one chip rightward one vertex.

This notion precisely captures the way the largest-labeled chips evolve under labeled chip-firing. Namely, we have the following.

Proposition 7. *If $\mathcal{C} \to \mathcal{D}$ then $[\mathcal{C}|_{\geq k}] \xrightarrow{R} [\mathcal{D}|_{\geq k}]$.*

Proof. Suppose we fire two chips (a) and (b) in $\mathcal{C}$: if $a, b < k$, that firing does not affect $[\mathcal{C}|_{\geq k}]$; if $k \leq a, b$, that firing corresponds to a firing in $[\mathcal{C}|_{\geq k}]$; and if $a < k \leq b$, then that firing corresponds to moving a chip rightward in $[\mathcal{C}|_{\geq k}]$. □

We want a strengthening of Proposition 6 that applies to rightward-reachability. To that end, we define a partial order on unlabeled configurations of n chips that can informally be thought of as "$c \leq d$ means d is obtained from c by moving chips rightward"; it is defined formally as follows. If c and d are configurations of n unlabeled chips on $\mathbb{Z}$, we write $c \leq d$ if and only if $\sum_{i \geq j} c(i) \leq \sum_{i \geq j} d(i)$ for all $j \in \mathbb{Z}$. Observe that $c \leq d$ implies that $\max(c) \leq \max(d)$ and $\min(c) \leq \min(d)$. We write $c \lessdot d$ to mean that d covers c according to this partial order $\leq$. In other words, $c \lessdot d$ means that d is obtained from c by moving one chip rightward one vertex.

An important property of this partial order is that it is preserved under stabilization, as we establish right now. In fact, something even stronger is true: stabilization preserves the cover relations of this partial order. (Note that $c \mapsto \sum_i i \cdot c(i)$ is a rank function for $\leq$. It is easy to verify that chip-firing moves preserve $\sum_i i \cdot c(i)$. So in fact stabilization being order-preserving is easily seen to be equivalent to it preserving cover relations.)

Lemma 8. *If $c \lessdot d$ then $\widetilde{c} \lessdot \widetilde{d}$.*

Proof. That $c \lessdot d$ means there is some c' and $i \in \mathbb{Z}$ such that $c = c' + \delta_i$ and $d = c' + \delta_{i+1}$. Define $a := \max\{j \leq i : j \notin \operatorname{supp}(\widetilde{c'})\}$ and $b := \min\{j \geq i+1 : j \notin \operatorname{supp}(\widetilde{c'})\}$. Thus there exists a configuration c'' such that $\widetilde{c'} = c'' + \delta_{[a+1,b-1]}$ and $\operatorname{supp}(c'') \cap [a,b] = \varnothing$. Proposition 2 then implies

$$\widetilde{\widetilde{c'} + \delta_i} = c'' + \delta_{[a,a+b-i-1]} + \delta_{[a+b-i+1,b]},$$
$$\widetilde{\widetilde{c'} + \delta_{i+1}} = c'' + \delta_{[a,a+b-i-2]} + \delta_{[a+b-i,b]}.$$

In particular, $\widetilde{\widetilde{c'} + \delta_i} \lessdot \widetilde{\widetilde{c'} + \delta_{i+1}}$. But $c = \widetilde{\widetilde{c'} + \delta_i}$ and $d = \widetilde{\widetilde{c'} + \delta_{i+1}}$, so the claim is proved. □

Lemma 8 is the key lemma which allows us to establish confluence of labeled chip-firing. It is also interesting in its own right as a result purely concerning unlabeled chip-firing. We now apply Lemma 8 to give a strengthening of Proposition 6 which applies to rightward-reachability.

Corollary 9. *If $c \xrightarrow{R} d$ then $\widetilde{c} \leq \widetilde{d}$ and consequently* $\min(\widetilde{c}) \leq \min(d)$.

Proof. Suppose $c \xrightarrow{R} d$. Thus there is some sequence $c_0, c_0', c_1, c_1', \ldots, c_\ell, c_\ell'$ of configurations with $c = c_0$ and $c_\ell' = d$ such that:

- $c_i \to c_i'$ for all $0 \leq i \leq \ell$;
- $c_{i-1}' \lessdot c_i$ for all $1 \leq i \leq \ell$.

We claim that $\widetilde{c} \leq \widetilde{c_i} = \widetilde{c'_i}$ for all $0 \leq i \leq \ell$. That $\widetilde{c_i} = \widetilde{c'_i}$ follows from $c_i \to c'_i$. So the crucial part of the claim is to show $\widetilde{c} \leq \widetilde{c_i}$. Clearly this holds for $i = 0$ since by definition $c_0 = c$. So assume $1 \leq i \leq \ell$ and $\widetilde{c} \leq \widetilde{c_{i-1}}$. Because $c'_{i-1} \lessdot c_i$, from Lemma 8 we get that $\widetilde{c_{i-1}} = \widetilde{c'_{i-1}} \lessdot \widetilde{c_i}$. Together with $\widetilde{c} \leq \widetilde{c_{i-1}}$ this implies $\widetilde{c} \leq \widetilde{c_i}$. So the claim is proved by induction. Taking $i = \ell$ in the claim gives $\widetilde{c} \leq \widetilde{c'_\ell}$, which is to say $\widetilde{c} \leq \widetilde{d}$. This implies $\min(\widetilde{c}) \leq \min(\widetilde{d})$. But $\min(\widetilde{d}) \leq \min(d)$ by Proposition 6. □

Now we can apply Corollary 9 to restrict, based on their labels, how far chips can move in our situation of interest.

Lemma 10. *Suppose $\Delta^n \to \mathcal{C}$. Then $-\lfloor (n+1-k)/2 \rfloor \leq \mathcal{C}(k) \leq \lfloor k/2 \rfloor$ for all $1 \leq k \leq n$.*

Proof. First we show $-\lfloor (n+1-k)/2 \rfloor \leq \mathcal{C}(k)$. By Proposition 7, $[\Delta^n|_{\geq k}] \xrightarrow{R} [C|_{\geq k}]$. Thus by Corollary 9, $\min(\widetilde{[\Delta^n|_{\geq k}]}) \leq \min(\widetilde{[C|_{\geq k}]})$. But $[\Delta^n|_{\geq k}] = (n+1-k)\delta_0$, and so Lemma 3 tells us that $\min(\widetilde{[\Delta^n|_{\geq k}]}) = -\lfloor (n+1-k)/2 \rfloor$. Thus indeed chip (k) must be at or to the left of the vertex $-\lfloor (n+1-k)/2 \rfloor$. That $\mathcal{C}(k) \leq \lfloor k/2 \rfloor$ then follows via Lemma 5. □

We are now ready to prove the main theorem, which says that when the number n of chips is even, the labeled chip-firing process on $\mathbb{Z}$ necessarily sorts these chips. It can be shown that the number of firings in this process is $\Theta(n^3)$, so this procedure is not being offered as a practical way to sort.

Theorem 11. *Suppose $n := 2m$ is even and $\Delta^n \to \mathcal{D}$ where $\mathcal{D}$ is stable. Then for all $1 \leq k \leq m$ we have that $\mathcal{D}(k) = -(m+1)+k$ and $\mathcal{D}(m+k) = k$.*

Proof. Let $n = 2m$ be even and let $\Delta^n \to \mathcal{D}$ with $\mathcal{D}$ stable. For all $1 \leq k \leq m$, the assertion that $\mathcal{D}(m+k) = k$ follows from $\mathcal{D}(m+1-k) = -k$ by Lemma 5. Thus we prove only that $\mathcal{D}(k) = -(m+1)+k$ for all $1 \leq k \leq m$.

The proof is by induction on k. So let us first address the base case $k = 1$. Lemma 10 says that $\mathcal{D}(i) > -m$ for all $2 \leq i \leq n$. (Here we use crucially that $n = 2m$ is even.) But on the other hand, we know thanks to Lemma 3 that vertex $-m$ is occupied in $\mathcal{D}$. So in fact it must be occupied by chip (1).

Now assume $k \geq 2$ and the result holds for all smaller values of k. We will use some internal lemmas in the proof ("internal" because they assume the inductive hypothesis).

Lemma 12. *If $\mathcal{D}(k) > -(m+1)+k$ then for all $1 \leq j \leq k-1$, chip (k) never fired together with chip (j) in the labeled chip-firing process $\Delta^n \to \mathcal{D}$.*

Proof. Suppose that $\mathcal{D}(k) > -(m+1)+k$. And suppose to the contrary that chip (k) did fire together with chip (j) for some $1 \leq j \leq k-1$ at some point in the labeled chip-firing process $\Delta^n \to \mathcal{D}$. Let us concentrate on the last moment when this happened: let $\mathcal{C}'$ be the step before chip (k) fired with some chip (j) with $1 \leq j \leq k-1$

for the last time (and thus define j to be the label of this other chip). Let $\mathcal{C}$ be the result of firing (k) and (j) together in $\mathcal{C}'$. So $\Delta^n \to \mathcal{C}'$, $\mathcal{C}$ is obtained from $\mathcal{C}'$ by firing (k) and (j) together, and $\mathcal{D}$ is obtained from $\mathcal{C}$ by a sequence of firings that either do not involve (k), or fire (k) together with a chip with a greater label. This implies that $[\mathcal{C} \setminus \{k\}] \xrightarrow{R} [\mathcal{D} \setminus \{k\}]$ and Corollary 9 thus yields $\widetilde{[\mathcal{C} \setminus \{k\}]} \leq [\mathcal{D} \setminus \{k\}]$. As a consequence of the assumptions that $\mathcal{D}(k) > -(m+1)+k$ and that $k \leq m$, together with Lemma 3, we have $[-m, -(m+1)+k] \subseteq \operatorname{supp}([\mathcal{D} \setminus \{k\}])$. Thus, since $\min(\widetilde{[\mathcal{C} \setminus \{k\}]}) \geq \min(\widetilde{n\delta_0}) = -m$ and $\widetilde{[\mathcal{C} \setminus \{k\}]}$ has at most one chip at each vertex, we have $[-m, -(m+1)+k] \subseteq \operatorname{supp}(\widetilde{[\mathcal{C} \setminus \{k\}]})$. Next, note that $[\mathcal{C} \setminus \{k\}] \lessdot [\mathcal{C}' \setminus \{k\}]$. So by applying Lemma 8, we conclude that $\widetilde{[\mathcal{C} \setminus \{k\}]} \lessdot \widetilde{[\mathcal{C}' \setminus \{k\}]}$, i.e., that $\widetilde{[\mathcal{C}' \setminus \{k\}]}$ is obtained from $\widetilde{[\mathcal{C} \setminus \{k\}]}$ by moving one chip rightward one vertex. In particular this means that we must have $[-m, -(m+1)+k-1] \subseteq \operatorname{supp}(\widetilde{[\mathcal{C}' \setminus \{k\}]})$ (by the same reasoning as in the previous line about $\operatorname{supp}(\widetilde{[\mathcal{C} \setminus \{k\}]})$. Now, chips (k) and (j) occupy the same vertex in $\mathcal{C}'$, which means $[\mathcal{C}' \setminus \{k\}] = [\mathcal{C}' \setminus \{j\}]$. So by starting from $\mathcal{C}'$ and repeatedly firing all chips other than (j) until we stabilize these other chips, we can eventually reach some configuration $\mathcal{D}'$ with $\widetilde{[\mathcal{D}' \setminus \{j\}]} = \widetilde{[\mathcal{C}' \setminus \{j\}]} = \widetilde{[\mathcal{C}' \setminus \{k\}]}$.

The upshot of the previous paragraph is that if the lemma is false then we can find a configuration $\mathcal{D}'$ with $\Delta^n \to \mathcal{D}'$ and $[-m, -(m+1)+k-1] \subseteq \operatorname{supp}([\mathcal{D}' \setminus \{j\}])$ for some $1 \leq j \leq k-1$. Let us show that this is impossible. For an unlabeled configuration c and $\ell \in \mathbb{Z}$, define $\varphi_\ell(c) := \sum_{i \leq \ell} (i - \ell - 1) \cdot c(i)$. It is easy to verify that if c' is obtained from c by firing at vertex i then

$$\varphi_\ell(c') = \begin{cases} \varphi_\ell(c) - 1 & \text{if } i = \ell + 1; \\ \varphi_\ell(c) & \text{otherwise.} \end{cases}$$

Since φ_ℓ weakly decreases with each chip-firing move, we always have $\varphi_\ell(\widetilde{c}) \leq \varphi_\ell(c)$; moreover, if $\varphi_\ell(c) = \varphi_\ell(\widetilde{c})$ then vertex $\ell+1$ never fires during the stabilization process $c \to \widetilde{c}$. Now, we claim that (j) is strictly to the right of vertex $-(m+1)+k-1$ in $\mathcal{D}'$: indeed, otherwise $\varphi_{-(m+1)+k-1}([\mathcal{D}']) < \varphi_{-(m+1)+k-1}(\widetilde{[\mathcal{D}']}) = \varphi_{-(m+1)+k-1}(\widetilde{n\delta_0})$ since $[-m, -(m+1)+k-1] \subseteq \operatorname{supp}([\mathcal{D}' \setminus \{j\}])$. If chip (j) is strictly to the right of vertex $-(m+1)+k-1$ in $\mathcal{D}'$, as it must be, then $\varphi_{-(m+1)+k-1}([\mathcal{D}']) = \varphi_{-(m+1)+k-1}(\widetilde{n\delta_0})$. So if we continue to stabilize, that is, if we let $\mathcal{D}''$ be such that $\mathcal{D}' \to \mathcal{D}''$ and $\mathcal{D}''$ is stable, then the vertex $-(m+1)+k$ never fires during the labeled chip-firing process $\mathcal{D}' \to \mathcal{D}''$. Consequently, chip (j) always remains strictly to the right of $-(m+1)+k-1$ during the process $\mathcal{D}' \to \mathcal{D}''$. So chip (j) is strictly to the right of $-(m+1)+k-1$ in the stable configuration $\mathcal{D}''$. But this contradicts our inductive hypothesis since $1 \leq j \leq k-1$. □

Lemma 13. *Chip (k) must have fired together with chip $(k-1)$ at some point in the labeled chip-firing process $\Delta^n \to \mathcal{D}$.*

Proof. Note that in the labeled chip-firing process, chips (k) and $(k-1)$ interact in the same way with all chips (j) for $j \neq k, k-1$. So if chip (k) and chip $(k-1)$ never fire together in the labeled chip-firing process $\Delta^n \to \mathcal{D}$, we can swap the roles of (k) and $(k-1)$ to reach a stable configuration $\mathcal{D}'$ where (k) and $(k-1)$ have swapped places. This contradicts our inductive hypothesis which says that there is only one vertex $(k-1)$ could end up at in a stable configuration. □

Lemmas 12 and 13 together imply that $\mathcal{D}(k) \leq -(m+1)+k$. By our inductive hypothesis, we know that vertex $-(m+1)+j$ is occupied by (j) for all $1 \leq j \leq k-1$. Thus $\mathcal{D}(k) = -(m+1)+k$. Therefore, the theorem is proved by induction. □

3 Extensions

3.1 Other graphs

An obvious question is if the labeled chip-firing process can be extended to other graphs beyond $\mathbb{Z}$. While we are far from being able to propose an interesting extension of labeled chip-firing to arbitrary graphs, we have found that several minor variants of the infinite path continue to exhibit confluence of certain initial configurations.

The first kind of variant involves adding loops to some vertices in the infinite path $\mathbb{Z}$. For $S \subseteq \mathbb{Z}$, let us use $\mathbb{Z}\langle S \rangle$ to denote the graph obtained from $\mathbb{Z}$ by adding a loop at each $i \in S$. To fire at a vertex $i \in \mathbb{Z}\langle S \rangle$ that has a loop (that is, for which $i \in S$), we now need to choose three chips (a), (b), and (c) which occupy i. Suppose $a < b < c$; then the firing consists of moving (a) leftward one vertex and moving (c) rightward one vertex. We imagine that the middle-labeled chip (b) travels along the loop and comes back to i, and so the location of (b) does not change as a result of this firing. Adding loops can certainly change which configurations stabilize uniquely: for example, Δ^3 stabilizes uniquely as a configuration on $\mathbb{Z}\langle\{0\}\rangle$, but Δ^4 does not.

Conjecture 14. *Let $S \subseteq \mathbb{Z}$. For the purposes of this conjecture consider all configurations (labeled and unlabeled) to be on the graph $\mathbb{Z}\langle S \rangle$. Suppose $n \in \mathbb{N}$ is such that:*

- $\min(\widetilde{n\delta_0}) < \min(\widetilde{(n-1)\delta_0})$;
- $\sum_{i \in S'} i = 0$ *where* $S' := S \cap [\min(\widetilde{n\delta_0}) + 1, \max(\widetilde{n\delta_0}) - 1]$.

Then Δ^n has a unique labeled stabilization.

Conjecture 14 should follow from a minor modification of the arguments presented in the last section. Some special cases of Conjecture 14 are especially worth highlighting: the conjecture says that Δ^n has a unique labeled stabilization as a configuration

on $\mathbb{Z}\langle\{0\}\rangle$ whenever n is odd; and the conjecture says that Δ^n has a unique labeled stabilization as a configuration on $\mathbb{Z}\langle\mathbb{Z}\rangle$ whenever $n \equiv 3 \mod 4$.

The second kind of variant of the infinite path $\mathbb{Z}$ involves parallel edge, but for reasons of space we will not discuss in this abstract are conjectures in this direction.

3.2 Other configurations

Let us use the notation $\widetilde{\mathcal{C}} := \{\mathcal{D} \colon \mathcal{C} \to \mathcal{D} \text{ and } \mathcal{D} \text{ is stable}\}$. Another natural problem is to understand $\widetilde{\mathcal{C}}$ for more configurations $\mathcal{C}$. For example, for $n = 1, 3, 5, 7, 9, \ldots$ we have $\#\widetilde{\Delta^n} = 1, 3, 12, 54, 232, \ldots$. Set $n := 2m + 1$. Of course $[\mathcal{D}] = [-m, m]$ for $\mathcal{D} \in \widetilde{\Delta^n}$, so we may identify elements of $\widetilde{\Delta^n}$ with permutations. Completely describing the permutations in $\widetilde{\Delta^n}$ seems hard, but there are at least a few nontrivial things we can say. First of all, Lemma 10 puts some restrictions on $\widetilde{\Delta^n}$. We can also say the following: for any injective, order-preserving map $\iota \colon [n] \to [n+1]$, if we relabel a configuration $\mathcal{D} \in \widetilde{\Delta^n}$ according to ι, add a new chip (j) to the origin where $\{j\} := [n+1] \setminus \mathrm{im}(\iota)$, and then stabilize the resulting configuration, the chips have to appear in sorted order. Indeed, this is a consequence of our main theorem, Theorem 11, because one possible way to stabilize Δ^{n+1} is to ignore chip (j) for as long as possible and instead first stabilize the chips with labels in $\mathrm{im}(\iota)$. Finally, we offer the following attractive conjecture about $\widetilde{\Delta^n}$: the maximum number of inversions for a permutation in $\widetilde{\Delta^n}$ is exactly m.

A different way to understand configurations for which there is not unique labeled stabilization would be probabilistically. There are at least three reasonable ways to carry out labeled chip-firing randomly: (1) at each step choose a chip-firing move uniformly at random among all possible moves; (2) at each step choose an unstable vertex uniformly at random and then choose a pair of chips at that vertex uniformly at random; or (3) choose a stabilization sequence uniformly at random among all (labeled) stabilization sequences. Based on some limited computer simulations, it appears that when m is large, random labeled chip-firing applied to Δ^{2m+1} leads to all chips ending up sorted with probability around .33, under all three protocols. We have no intuition for why this probability should not converge to 0. It is clear that it cannot converge to a limit greater than $1/3$, since $2/3$ of the time the last move fails to put the chips in sorted order. We conjecture that the probability of sorting converges to $1/3$ exactly.

3.3 Other types

Pavel Galashin has pointed out a compelling way to rephrase our result that situates it in a broader context. Consider the vector $v \in \mathbb{Z}^n$ that records at each step the positions of the labeled chips; the origin $(0, \ldots, 0)$ corresponds to the starting configuration. Firing amounts to adding the vector $e_i - e_j$ with $i < j$ to the current vector v, and we are permitted to do this if and only if $e_i - e_j$ is perpendicular to v. These vectors are of course

the positive roots of the root system of type A_{n-1}. Galashin's computer experiments suggests that confluence occurs for "vector chip-firing" in some but not all other types. For instance, in type C_n it seems that confluence occurs when $n \equiv 1, 2 \mod 4$ and in type D_n when $n \equiv 3 \mod 4$. And it follows from our result via symmetry that B_n is confluent for *all* n.

Acknowledgements

We thank Pedro Felzenszwalb, Pavel Galashin, Caroline Klivans, Gregg Musiker, and Peter Winkler for useful comments and discussion. The first author was supported by NSF grant #1122374. The third author was supported by NSF grant #1001905.

References

[1] R. Anderson, L. Lovász, P. Shor, J. Spencer, É. Tardos, and S. Winograd. "Disks, balls, and walls: analysis of a combinatorial game". *Amer. Math. Monthly* **96** (1989), pp. 481–493. DOI.

[2] P. Bak, C. Tang, and K. Wiesenfeld. "Self-organized criticality: An explanation of the $1/f$ noise". *Phys. Rev. Lett.* **59** (1987), pp. 381–384. DOI.

[3] A. Björner, L. Lovász, and P. W. Shor. "Chip-firing games on graphs". *European J. Combin.* **12** (1991), pp. 283–291. DOI.

[4] B. Bond and L. Levine. "Abelian networks I. Foundations and examples". *SIAM J. Discrete Math.* **30** (2016), pp. 856–874. DOI.

[5] B. Bond and L. Levine. "Abelian networks II: Halting on all inputs". *Selecta Math. New Ser.* **22** (2016), pp. 319–340. DOI.

[6] B. Bond and L. Levine. "Abelian networks III: The critical group". *J. Algebraic Combin.* **43** (2016), pp. 635–663. DOI.

[7] D. Dhar. "Self-organized critical state of sandpile automaton models". *Phys. Rev. Lett.* **64** (1990), pp. 1613–1616. DOI.

[8] D. Dhar. "The Abelian sandpile and related models". *Physica A: Stat. Mech. Appl.* **263** (1999), pp. 4–25. DOI.

[9] A. Engel. "Why Does the Probabilistic Abacus Work?" *Educ. Stud. Math.* **7** (1976), pp. 59–69. DOI.

[10] A. E. Holroyd, L. Levine, and P. Winkler. "Abelian logic gates". 2015. arXiv:1511.00422.

[11] M. H. A. Newman. "On theories with a combinatorial definition of "equivalence."" *Ann. of Math. (2)* **43** (1942), pp. 223–243. DOI.

[12] J. Spencer. "Balancing vectors in the max norm". *Combinatorica* **6** (1986), pp. 55–65. DOI.