Abstract. The *parallel chip-firing game* is an automaton on graphs in which vertices "fire" chips to their neighbors. This simple model contains much emergent complexity and has many connections to different areas of mathematics. In this work, we study *firing sequences*, which describe each vertex's interaction with its neighbors in this game. First, we introduce the concepts of *motors* and *motorized games*. Motors both generalize the game and allow us to isolate local behavior of the (ordinary) game. We study the effects of motors connected to a tree, and show that motorized games can be transformed into ordinary games if the motor's firing sequence occurs in some ordinary game. Then, we completely characterize the periodic firing sequences that can occur in an ordinary game, which have a surprisingly simple combinatorial description.

MOTORS AND IMPOSSIBLE FIRING PATTERNS IN THE PARALLEL CHIP-FIRING GAME

TIAN-YI JIANG, ZIV SCULLY, YAN X ZHANG

1. Introduction.

Background. The *parallel chip-firing game*, also known as the *discrete fixed-energy sandpile model*, is an automaton on graphs in which vertices that have at least as many chips as incident edges "fire" chips to their neighbors. In graph theory, it has been studied in relation with the critical group of graphs [3]. In computer science, it is able to simulate any two-register machine and is thus universal [8]. As a specific case of the *abelian sandpile model*, which is itself a generalization of a sandpile model introduced by Bak, Tang, and Weisenfeld [1, 2] in the study of self-organized criticality, it has even more links with other fields.

The Game. The parallel chip-firing game is played on a graph as follows:

- At first, a nonnegative integer number of chips is placed on each vertex of the graph.
- The game then proceeds in discrete turns. Each turn, a vertex checks to see if it has at least as many chips as incident edges.
  - If so, that vertex *fires*.
  - Otherwise, that vertex *waits*.
- To fire, a vertex passes one chip along each of its edges. All vertices that fire in a particular turn do so in parallel.
- Immediately after firing or waiting, every vertex receives any chips that were fired to it.

Here we will only consider games on finite, undirected, connected graphs, though the definition of the game can be easily generalized for arbitrary multidigraphs. An example game is illustrated in Figure 1.1. Given a parallel chip-firing game as σ, we refer to the chip configuration, also called the position, at a particular time $t \in \mathbb{N}$ as σ_t.

The total number of chips on all vertices of the graph is constant throughout a game, so there are finitely many possible positions in every game. Therefore, every game eventually reaches a position σ_t that is identical to a later position σ_{t+p} for some $t, p \in \mathbb{N}$ with $p > 0$. (We write $\sigma_t = \sigma_{t+p}$.) The game is deterministic, so $\sigma_{t+n} = \sigma_{t+n+p}$ for all $n \in \mathbb{N}$. Thus, every parallel chip-firing game is eventually periodic.

In this paper, we concern ourselves with both *firing sequences* and *periodic firing patterns* of vertices. Each is a binary string representing whether or not a particular vertex fires or waits each turn. The sequence covers all times from 0 to infinity, while the periodic pattern covers just one period.

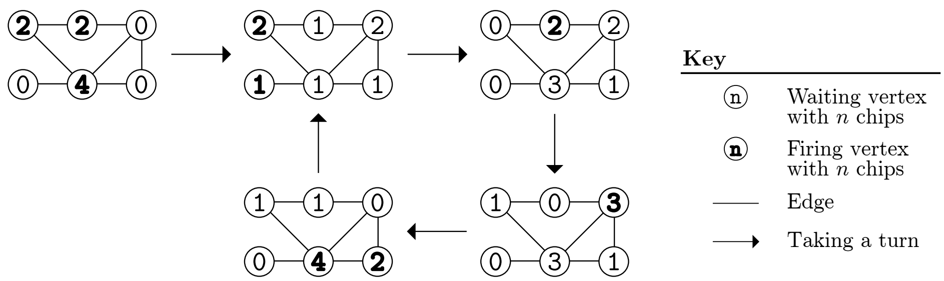

Fig. 1.1: A parallel chip-firing game. From an initial position in the upper left, the game eventually enters a period of length 4.

Previous Work. The periodicity of the parallel chip-firing game gives rise to two questions. First, what characteristics of a game and its underlying graph determine the length of a period? It is known exactly what periods are possible on certain classes of graphs, such as trees [4], simple cycles [6], the complete graph [13], and the complete bipartite graph [10]. For these graphs, the maximum period lengths are bounded by the number of vertices, but Kiwi et al. [11] constructed graphs on which the period of games can grow exponentially with polynomial increase in the number of vertices. There are also results regarding the total number of chips in a game. Kominers and Kominers [12] showed that games with a sufficiently large density of chips must have period 1. Dall'Asta [6] and Levine [13], in their respective characterizations of periods on cycles and complete graphs, related the total number of chips to a game's *activity*, the fraction of turns during which a vertex fires. The denominator of the activity must divide the period.

Second, we notice that some but not all positions σ_t are *periodic*, satisfying $\sigma_t = \sigma_{t+p}$ for some positive $p \in \mathbb{N}$. What characterizes periodic positions? This problem has not been as extensively studied. Dall'Asta [6] characterized the periodic positions of games on cycles.

Our Results. We hope to advance the understanding of both of these questions through the study of firing sequences and periodic firing patterns.

After precisely defining the parallel chip-firing game in Section 2, the first half of the paper develops a new tool for studying the chip-firing game: *motors*, vertices that fire with a regular pattern independent of normal chip-firing rules. Games with motors are called *motorized games*. Motors allow us to study the behavior of subgraphs in ordinary parallel chip-firing games. In Section 3 we show that vertices always "follow" a motor in periodic motorized games on trees. In Section 4, we prove that periodic motorized games can be transformed into ordinary games as long as the firing sequence of each motor occurs in an ordinary game.

The second half of the paper characterizes the possible periodic firing patterns in parallel chip-firing games. Section 5 briefly steps away from the game to study certain signed sums of periodic binary sequences. The result is an inequality applicable to edges of the graph of a parallel chip-firing game. In Section 6, we sum this inequality over all relevant edges to show that periodic firing patterns with both consecutive 0s and consecutive 1s cannot occur in a parallel chip-firing game. This, along with an already known construction, fully characterizes the periodic firing patterns possible in parallel chip-firing games. Finally, in Section 7, we examine some implications of this theorem.

2. Preliminaries.

Definitions. A *parallel chip-firing game* σ on a graph $G = (V(G), E(G))$ is a sequence $(\sigma_t)_{t\in\mathbb{N}}$ of ordered tuples with natural number elements indexed by $V(G)$. Each tuple represents the chip configuration at a particular turn, where each element of the tuple is the number of chips on the corresponding vertex. We define the following for all $v \in V(G)$:

$$\begin{aligned}
N(v) &= \{w \in V(G) \mid \{v,w\} \in E(G)\}\\
d(v) &= \#N(v)\\
\sigma_t(v) &= \text{number of chips on } v \text{ in position } \sigma_t\\
F_t(v) &= \begin{cases} 0 & \text{if } \sigma_t(v) \le d(v)-1\\ 1 & \text{if } \sigma_t(v) \ge d(v)\end{cases}\\
\Phi_t(v) &= \sum_{w\in N(v)} F_t(w).
\end{aligned}$$

In a parallel chip-firing game, σ_t induces σ_{t+1}. For all $v \in V(G)$,

$$\sigma_{t+1}(v) = \sigma_t(v) + \Phi_t(v) - F_t(v)d(v), \tag{2.1}$$

so an initial position suffices to define a game on a given graph. When $F_t(v) = 0$, we say v *waits* at t, and when $F_t(v) = 1$, we say v *fires* at t.

A position σ_t is *periodic* if and only if there exists $p \in \mathbb{N}$ such that $\sigma_t = \sigma_{t+p}$. The minimum such p for which this occurs is the *period* of σ and is denoted T. Abusing notation slightly, "a period" of a game σ may also refer to a set of times $\{t, t+1, \ldots, t+T-1\}$, where σ_t is periodic. The parallel chip-firing game is deterministic and there are finitely many possible positions on a given graph with a given number of chips, so for any game σ, there exists $t_0 \in \mathbb{N}$ such that σ_t is periodic for all $t \ge t_0$. If the initial position of a game is periodic, we may also call the game itself periodic.

Notation. Definitions for invented notation are given in the section indicated in the last column.

Parallel Chip-Firing		*Defined in*
$\sigma_t(v)$	Number of chips on vertex v in position σ_t.	Section 2
$F^\sigma(v)$	Periodic firing pattern of v.	Section 6
$F^\sigma_t(v)$	Indicates whether or not vertex v fires in σ_t.	Section 2
$\Phi^\sigma_t(v)$	Number of chips vertex v will receive in σ_t.	Section 2
$R^\sigma_f(v)$	Set of times for which a vertex v waits (if $f=0$) or fires (if $f=1$) in σ.	Section 3
T^σ	Period of σ.	Section 2
M^σ	Set of vertices that are motors in σ.	Section 3
P^σ	Set of periodic firing patterns in σ.	Section 6
Q^σ	Set of "clumpy" periodic firing patterns in σ.	Section 6

Graphs	
$V(G)$	Vertex set of graph G
$E(G)$	Edge set of graph G
$N_G(v)$	Neighbors of vertex v.
$d_G(v)$	The degree of vertex v in graph G.
Other	
$[a, b]$	The integer interval $\{a, a+1, \ldots, b\}$.

We leave out the subscript G or superscript σ if there is no ambiguity.

3. Motors. Let G be a graph. Suppose we wish to study the periodic behavior of games on G, focusing on a particular subgraph $H \subseteq G$. Consider

$$X = \{v \in V(G) \setminus V(H) \mid N(v) \cap V(H) \neq \emptyset\},$$

the set of vertices "just outside" of H. Knowing the initial chip configuration on $V(H) \cup X$ is in general not enough to determine all subsequent configurations because vertices in X may have interactions with vertices outside of $V(H) \cup X$. However, we do know that every vertex assumes a pattern of firing and waiting that repeats periodically as soon as a game reaches a periodic position. Therefore, we can simulate the presence of the rest of G by having each vertex in X fire with a regular pattern regardless of the number of chips it receives.

The *firing sequence* of a vertex v in game σ is the sequence $(F_t(v))_{t\in\mathbb{N}}$. A *motorized parallel chip-firing game*, or simply "motorized game", on G is a game σ obeying (2.1) with a non-empty set of *motors* $M \subseteq V(G)$. Each motor follows a predetermined firing sequence, firing without regard for the normal rules of the parallel chip-firing game, which means, for example, that a motor may have a negative number of chips. Put another way, for each $m \in M$, $F_t(m)$ does not depend on $\sigma_t(m)$. The term "ordinary game" refers to a game with no motors when there is ambiguity. A motorized game is shown in Figure 3.1.

If a motorized game σ is eventually periodic (which is the case if every motor's firing sequence is eventually periodic), then just as in an ordinary game, every vertex fires the same number of times each period. The proof is identical to the proof of this fact for ordinary games [10]: all neighbors of the vertex that fires the most times each period must also fire that maximal number of times, and by induction, so do all vertices. (Recall that we consider in this paper only connected graphs.)

We define

$$R_f(v) = \{t \in \mathbb{N} \mid F_t(v) = f\}.$$

Call an interval $[a, b]$ with $a < b$ a *max-clump* of $v \in V(G)$ if and only if $[a, b] \subseteq R_f(v)$ and $F_{a-1}(v) = F_{b+1}(v) = 1 - f$, where $f \in \{0, 1\}$. Given $v \in V(G)$, we can express $\mathbb{N}$ as the union of max-clumps of v and times during which v alternates between firing and waiting.

The proof of Theorem 3.2 follows the same structure as the proof that ordinary games on trees have period 1 or 2 [4]. In fact, we rely on a lemma originally introduced for that proof.

LEMMA 3.1 ([4, Lemma 1]). *Let σ be a game on G. For all $v \in V(G)$ and $f \in \{0, 1\}$, if $[a, b] \subseteq R_f(v)$, then there exists $w \in N(v)$ such that $[a-1, b-1] \subseteq R_f(w)$.*

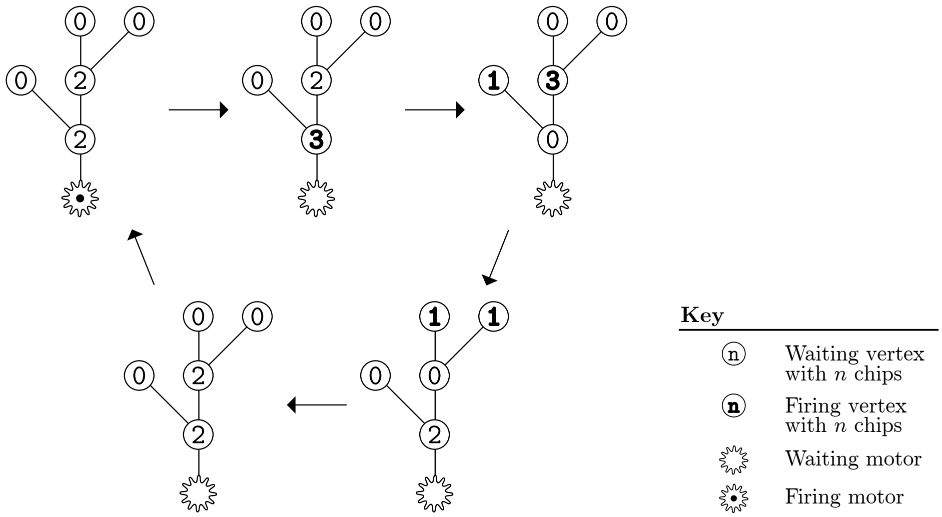

Fig. 3.1: A motorized parallel chip-firing game. The motor has firing sequence $(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, \dots)$.

Less technically, every burst of firing or waiting by a vertex must be supported by at least one of its neighbors. The lemma follows from the pigeonhole principle and Lemma 6.1, which we state and prove later.

THEOREM 3.2. *Let σ be a periodic motorized game on tree T. For all $v \in V(T)$ and $f \in \{0, 1\}$, if $[a, b] \subseteq R_f(v)$, then $[a - D, b - D] \subseteq R_f(m)$ for some $m \in M$, where D is the distance from m to v.*

Proof. The fact that σ is periodic means we need not worry about negative turn indices.

Let $v_0 = v$ and $[a_0, b_0] \supseteq [a, b]$ be a max-clump of v_0. By Lemma 3.1, given a vertex $v_{i-1} \notin M$ with $[a_{i-1}, b_{i-1}] \in R_f(v_{i-1})$, we can pick a vertex $v_i \in N(v_{i-1})$ and integers a_i and b_i such that $[a_i, b_i]$ is a max-clump of v_i and

$$[a_{i-1} - 1, b_{i-1} - 1] \subseteq [a_i, b_i] \subseteq R_f(v_i).$$

If there is a maximum i for which v_i exists, that vertex must be a motor, which would mean

$$[a - D, b - D] \subseteq [a_D, b_D] \subseteq R_f(m),$$

where D is the maximum i and $m = v_D \in M$. Thus, it suffices to show that the sequence $\{v_0, v_1, \dots\}$ eventually terminates. There are finitely many vertices in the graph, so it suffices to show that the v_i are all distinct.

T has no cycles, so if $v_i \neq v_{i-2}$ for all i, then all v_i are distinct. Suppose for contradiction that $v_i = v_{i-2}$ for some i. Then $[a_i, b_i] \cup [a_{i-2}, b_{i-2}] \subseteq R_f(v_i)$. However, $[a_{i-2} - 2, b_{i-2} - 2] \subseteq [a_i, b_i]$, so $[a_{i-2} - 2, b_{i-2}] \subseteq R_f(v_i)$. Therefore, $[a_{i-2}, b_{i-2}]$ is not a max-clump, a contradiction, so $v_i \neq v_{i-2}$ for all i. ∎

Call a firing sequence *clumpy* if it contains two consecutive 0s and two consecutive 1s; otherwise, call it *nonclumpy*.

COROLLARY 3.3. *Let σ be a periodic motorized game on tree M with a single* *motor m. If m has a nonclumpy firing sequence but has at least one max-clump, then $F_{t+D}(v) = F_t(m)$ for all $v \in V(T)$ and $t \in \mathbb{N}$, where D is the distance from v to m.*

Proof. Let $v \in V(T)$. By Theorem 3.2, v has a nonclumpy firing sequence because m does. All vertices fire the same number of times every period [10, Proposition 2.5], so v must have at least one max-clump, again because m does. For every max-clump $[a, b] \subseteq R_f(v)$, $[a-D, b-D] \subseteq R_f(m)$, where $f \in \{0, 1\}$. The non-max-clump intervals of v's firing sequence are alternations between 0 and 1, starting and ending with $1-f$. The same must be true of m for it to fire the same number of times each period as v. □

The reason we require the games in Theorem 3.2 and Corollary 3.3 to be periodic is to consider arbitrarily many past turns. We can likely weaken this condition if we require the statements to be true only after sufficiently many turns, though exactly how many turns that is could depend on the activity (firing frequency; see [13]) of the motor, the size of the tree, and the total number of chips in the initial position.

4. Simulating Motors. In this section, to refer to multiple chip-firing games unambiguously, we include the subscripts and superscripts in, for example, $d_G(v)$ and $F_t^\sigma(v)$.

We call a firing sequence $(f_t)_{t\in\mathbb{N}}$ *possible* if there exists an ordinary game σ on some graph G such that $F_t^\sigma(v) = f_t$ for all $t \in \mathbb{N}$. Our next theorem states that we can simulate motorized games with ordinary games as long as every motor's firing sequence is possible. Figure 4.1 demonstrates the concept.

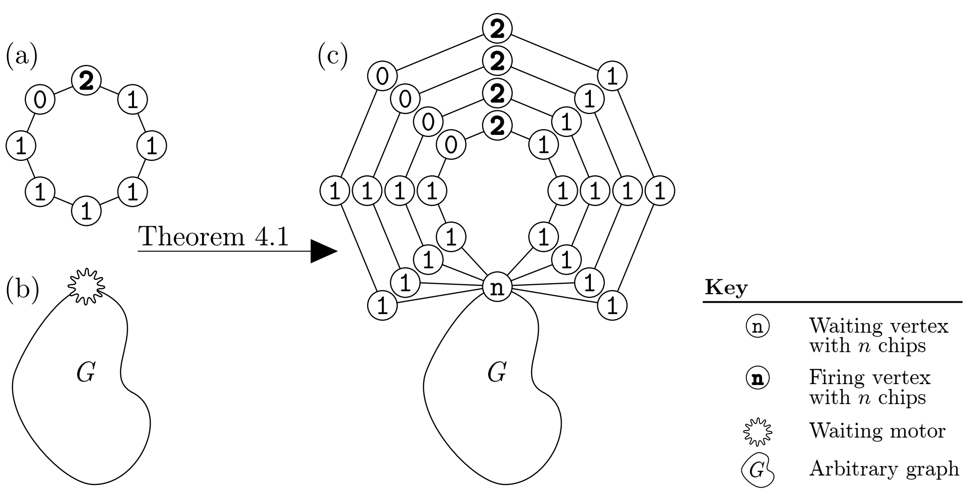

Fig. 4.1: Suppose the motor in motorized game (b) has firing sequence $(0, 0, 0, 0, 1, 0, 0, 0, \dots)$. This occurs in ordinary game (a). By using sufficiently many copies of (a) and carefully choosing n, we construct (c). The behavior of G in (c) is identical to the behavior of G in (b).

THEOREM 4.1. *Let σ be a periodic motorized game on G. If every motor's firing sequence is possible, then there exists an ordinary game σ' on a graph $H \supseteq G$ such that*

- $F_t^{\sigma'}(u) = F_t^\sigma(u)$ *for all $t \in \mathbb{N}$ and $u \in V(G)$,*
- $d_H(v) = d_G(v)$ *for all $v \in V(G) \setminus M^\sigma$, and*

- *the subgraph of H induced by $V(G)$ is G. (That is, H contains no edges between vertices of G that are not also in G.)*

Proof. Our approach will be, for each $m \in M^\sigma$, to attach many copies of a graph with a vertex with m's firing sequence to m. If sufficiently many copies are attached, the number of chips m has due to its neighbors in G becomes irrelevant as to whether or not it fires.

For each $m \in M^\sigma$, let A_m be a graph such that there exists a game σ^m and some vertex $u_m \in V(A_m)$ such that $F_t^{\sigma^m}(u_m) = F_t^\sigma(m)$ for all $t \in \mathbb{N}$. Let a_m and b_m be the minimum and maximum respectively of $\{\sigma_t(m) \mid t \in \mathbb{N}\}$. Let $k_m = b_m - a_m + 1$. Let H be the union of G and k_m copies of each A_m, with G and the copies of A_m disjoint except for $m = u_m$ for each $m \in M^\sigma$. (Thinking of the graphs as pointed topological spaces, each with basepoint m or u_m, H is a wedge sum.)

It is clear by construction that H contains no new edges between vertices of G and that

- $d_H(m) = k_m d_{A_m}(m) + d_G(m)$ for all $m \in M^\sigma$,
- $d_H(u) = d_{A_m}(u)$ for all $u \in V(A_m) \setminus \{m\}$ for each $m \in M^\sigma$, and
- $d_H(v) = d_G(v)$ for all $v \in V(G) \setminus M^\sigma$.

Suppose that for some $t \in \mathbb{N}$, σ'_t satisfies the following.

1. $\sigma'_t(m) = k_m \sigma_t^m(m) + d_G(m) + \sigma_t(m) - a_m$ for all $m \in M^\sigma$.
2. $\sigma'_t(u) = \sigma_t^m(u)$ for all $u \in V(A_m) \setminus \{m\}$ for each $m \in M^\sigma$.
3. $\sigma'_t(v) = \sigma_t(v)$ for all $v \in V(G) \setminus M^\sigma$.

We will show that σ'_{t+1} satisfies the above as well. We have $d_H(v) = d_G(v)$ for all $v \in V(G) \setminus M^\sigma$, so $F_t^{\sigma'}(v) = F_t^\sigma(v)$ for all $v \in V(G) \setminus M^\sigma$. Similarly, $F_t^{\sigma'}(u) = F_t^{\sigma^m}(u)$ for all $u \in V(A_m) \setminus \{m\}$ for each $m \in M^\sigma$. Finally, for all $m \in M^\sigma$, if $F_t^{\sigma^m}(m) = 0$, then

$$\begin{aligned}
\sigma'_t(m) &\leq k_m(d_{A_m}(m) - 1) + d_G(m) + \sigma_t(m) - a_m \\
&= k_m d_{A_m}(m) + d_G(m) + (\sigma_t(m) - b_m) - 1 \\
&\leq d_H(m) - 1,
\end{aligned}$$

and if $F_t^{\sigma^m}(m) = 1$, then

$$\begin{aligned}
\sigma'_t(m) &\geq k_m d_{A_m}(m) + d_G(m) + (\sigma_t(m) - a_m) \\
&\geq d_H(m),
\end{aligned}$$

so $F_t^{\sigma'}(m) = F_t^{\sigma^m}(m) = F_t^\sigma(m)$.

We know $F_t^{\sigma'}(v) = F_t^\sigma(v)$ for all $v \in V(H)$, so clearly $\sigma'_{t+1}(v) = \sigma_{t+1}(v)$ for all $v \in V(G) \setminus M^\sigma$ and $\sigma'_{t+1}(u) = \sigma_{t+1}^m(u)$ for all $u \in V(A_m) \setminus \{m\}$ for each $m \in M^\sigma$. Finally, we have

$$\begin{aligned}
\sigma'_{t+1}(m) &= k_m \sigma_t^m(m) + d_G(m) + \sigma_t(m) - a_m \ + \Phi_t^{\sigma'}(v) - F_t^{\sigma'}(v) d_H(v) \\
&= k_m \sigma_t^m(m) + d_G(m) + \sigma_t(m) - a_m + \Phi_t^\sigma(v) - F_t^\sigma(v) d_G(v) + \\
&\quad k_m \Phi_t^{\sigma^m}(v) - k_m F_t^{\sigma^m}(v) d_{A_m}(v) \\
&= k_m(\sigma_t^m(m) + \Phi_t^{\sigma^m}(v) - F_t^{\sigma'}(v) d_{A_m}(v)) + d_G(m) + \\
&\quad (\sigma_t(m) + \Phi_t^\sigma(v) - F_t^\sigma(v) d_G(v)) - a_m \\
&= k_m \sigma_{t+1}^m(m) + d_G(m) + \sigma_{t+1}(m) - a_m.
\end{aligned}$$

for all $m \in M^\sigma$.

$$\underbrace{01000100}\overbrace{11011}\underbrace{00}\overbrace{101011}$$

Fig. 5.1: A string's 0-sectors (marked below) and 1-sectors (marked above). Roughly speaking, b-sectors have no two $\overline{b}$s in a row and extend as far left as possible.

We can distribute chips in σ'_0 such that it satisfies (1), (2), and (3), in which case, by induction, σ'_t satisfies (1), (2), and (3) for all $t \in \mathbb{N}$, implying $F_t^{\sigma'}(v) = F_t^{\sigma}(v)$ for all $v \in V(G)$. ∎

Again, we may relax the condition that the game be periodic. In this case, the periodicity ensures that the number of chips on each motor is bounded. This is easily shown to be equivalent to each motor having the same activity, so that is a sufficient condition for the theorem. Of course, because all ordinary games are eventually periodic, any motorized game in which each motor has a possible firing sequence will eventually be periodic.

In Theorem 3.2, motors were primarily a convenient intuition and terminology; we could have proved a similar theorem within the context of the ordinary parallel chip-firing game, though its statement would have been messier. Theorem 4.1 demonstrates another way in which the motor concept is useful. Its constructive power makes certain conjectures easy to prove or disprove by example. For instance, motors make it easy to construct games in which the period isn't bounded by the number of vertices.

5. Signed Sums of Binary Sequences. Throughout this section, if $b \in \{0, 1\}$, we may write $\overline{b} = 1 - b$.

We take a break from the parallel chip-firing game to consider binary strings. We denote the i^{th} element of a binary string p as p_i. For simplicity, any integer equivalent to $i \bmod n$ may replace i. Let a *b-sector* of p, where $b \in \{0, 1\}$, be an integer interval $[x, y]$ such that

$$\begin{aligned} p_x = p_{y-1} = p_y &= b \\ p_{x-1} = p_{x-2} &= \overline{b} \\ \forall i \in [x+1, y-3] : p_i + p_{i+1} &\neq 2\overline{b}. \end{aligned}$$

That is, the image of a 0-sector is preceded by two 1s, starts with 0, ends with two 0s and contains no two consecutive 1s. The same statements with 0s and 1s swapped are true for 1-sectors.

It is easy to see that almost any string can be partitioned into 0- and 1-sectors in exactly one way, with exceptions only for always-alternating strings (e.g. 010101) that can be thought of as one 0-sector or one 1-sector. Figure 5.1 has an example.

Let

$$s_i(p) = \begin{cases} -1 & \text{if } i \text{ is in a 0-sector of } p \\ 1 & \text{if } i \text{ is in a 1-sector of } p \end{cases}$$

$$\delta_i(p) = \begin{cases} 0 & \text{if } i \text{ is in a } b\text{-sector of } p \text{ and } i+1 \text{ is in a } b\text{-sector of } p \\ 1 & \text{if } i \text{ is in a } b\text{-sector of } p \text{ and } i+1 \text{ is in a } \overline{b}\text{-sector of } p. \end{cases}$$

Our main theorem in this section concerns the sum

$$M_S(p,q) = \sum_{i\in S}(s_i(p)(p_i - q_{i-1}) + s_i(q)(q_i - p_{i-1}) - \delta_i(p) - \delta_i(q)), \tag{5.1}$$

where p, q are length-n binary sequences and $S \subseteq [0, n-1]$. This sum, superficially speaking, measures each sequence's "disagreement" with the other shifted back one step minus the number of sector switches. The rules of the parallel chip-firing game put a global upper bound on the total disagreement between vertices, yet the following theorem says that sector switches require disagreement. We show in Section 6 that this implies that firing sequences with sector switches are impossible once a game has become periodic.

Theorem 5.1. $M_{[0,n-1]}(p,q) \geq 0$.

Proof. We can caculate $M_{[0,n-1]}(p,q)$ as $\sum_{i=0}^{n-1} M_{\{i\}}(p,q)$, and $M_{\{i\}}(p,q)$ is determined by p_{i-1}, p_i, $s_i(p)$, $s_{i+1}(p)$, and the same data for q. The motivation for using $s_{i+1}(p)$ as opposed to $s_{i-1}(p)$ in $\delta_i(p)$ is that a switch away from a b-sector can occur between i and $i+1$ only if $p_{i-1} = p_i = b$. Let

$$\mu_i(p,q) = (p_{i-1}, p_i, s_i(p), s_{i+1}(p), q_{i-1}, q_i, s_i(q), s_{i+1}(q))$$

and let $\mathscr{G}$ be a weighted digraph with

$$V(\mathscr{G}) = \{\mu_i(p,q) \mid p, q \text{ strings}, i \in \mathbb{N}\}$$
$$E(\mathscr{G}) = \{(u,v,w) \mid \exists p, q, i\colon u = \mu_i(p,q), v = \mu_{i+1}(p,q), w = M_{\{i\}}(p,q)\}.$$

(The third item of each edge is its weight.) Note that not every possible tuple is a vertex. Define the weight of a path to be the sum of the weights of its member edges, and call a path negative if it has negative weight. We can calculate the $M_{[0,n-1]}(p,q)$ as the weight of a path induced by the sequence of vertices $(\mu_0(p,q), \ldots, \mu_{n-1}(p,q), \mu_0(p,q))$. Therefore, it suffices to show that $\mathscr{G}$ has no negative cycles. Running the Bellman-Ford algorithm [5] on $\mathscr{G}$ shows this to be the case. We describe $\mathscr{G}$ and the algorithm in a Python program in Appendix A. ∎

6. Nonclumpiness of Periodic Firing Patterns. We consider parallel chip-firing game σ on undirected graph G. The *periodic firing pattern* (PFP) of a vertex $v \in V(G)$ is the binary string

$$F_{t_0}(v) \ldots F_{t_0+T-1}(v),$$

where t_0 is the smallest natural number such that σ_{t_0} is periodic[1]. We write the PFP of v as $F(v)$. For simplicity, we assume here that $t_0 = 0$ and index PFPs modulo T.

Let P be the set of all PFPs occurring in σ. Call a PFP with both consecutive 0s and consecutive 1s *clumpy*, and let Q be the set of all clumpy PFPs occurring in σ. (Recall that the T^{th} and 0^{th} entries of a PFP are the same, so, for example, 011010 is clumpy.) It is known that, given almost any[2] nonclumpy PFP, one can construct a

[1] The reason we introduce PFPs instead of continuing to reason with firing sequences is because a PFP is aware of the period of the game it occurs in. For instance, the PFPs 01 and 0101 result in the same periodic firing sequence, but while latter might occur in the same game as the PFP 0011, the former cannot.

[2] The given construction requires that the PFP not be decomposable to a repeated substring. Using Theorem 4.1, one can expand the construction to any nonclumpy PFP other than those that are 01 or 10 repeated more than once. These PFPs turn out to be impossible, though the corresponding firing sequences are possible in games of period 2.

parallel chip-firing game on a simple cycle in which every vertex has that PFP shifted by some number of steps [6]. We prove here that clumpy PFPs cannot occur in any parallel chip-firing game.

Lemma 6.1. *For all $v \in V(G)$ and $a, b \in \mathbb{N}$,*

$$-d(v) + 1 \leq \sum_{t=a}^{b} (\Phi_{t-1}(v) - d(v)F_t(v)) \leq d(v) - 1. \tag{6.1}$$

Proof. We express $\sigma_b(v)$ in terms of $\sigma_{a-1}(v)$.

$$\sigma_b(v) = \sigma_{a-1}(v) + \sum_{t=a-1}^{b-1} (\Phi_t(v) - d(v)F_t(v))$$

$$\sigma_b(v) - d(v)F_b(v) = \sigma_{a-1}(v) - d(v)F_{a-1}(v) + \sum_{t=a}^{b} (\Phi_{t-1}(v) - d(v)F_t(v))$$

Recall that $0 \leq \sigma_t(v) - d(v)F_t(v) \leq d(v) - 1$ for all $t \in \mathbb{N}$ such that σ_t is periodic, which gives the desired inequality. ∎

We define

$$\begin{aligned} \tau(p) &= \{v \in V(G) \mid F(v) = p\} \\ \pi(p, q) &= \{\{v, w\} \in E(G) \mid F(v) = p, F(w) = q\} \\ m_S(p, q) &= \sum_{i \in S} (p_i - q_{i-1}) \end{aligned}$$

for binary strings p and q. We now prove our main result: clumpy PFPs do not occur in the parallel chip-firing game.

Theorem 6.2. *$\#\tau(p) = 0$ for all $p \in Q$.*

Proof. Roughly, summing an inequality given by Lemma 6.1 over all vertices with clumpy PFPs bounds a quantity below, and summing an inequality given by the Theorem 5.1 over all edges incident with a vertex with a clumpy PFP gives an upper bound on the same quantity. The lower bound is the total number of vertices with clumpy PFPs, and the upper bound is 0.

Let $a, b \in \mathbb{N}$ and $v \in V(G)$. Grouping the sum in (6.1) by v's neighbors instead of time steps yields

$$-d(v) + 1 \leq -\sum_{w \in N(v)} m_{[a,b]}(F(v), F(w)) \leq d(v) - 1.$$

Regrouping gives us

$$1 \leq \sum_{w \in N(v)} (1 + r m_{[a,b]}(F(v), F(w))),$$

where $r = \pm 1$. Let $p \in P$. The above summed over $v \in \tau(p)$ is

$$\#\tau(p) \leq \sum_{\substack{v \in \tau(p) \\ w \in N(v)}} (1 + r_v m_{[a,b]}(p, F(w))),$$

where each $r_v = \pm 1$ can depend on v. (Notation: ranges for outer sums are above ranges for inner sums.)

For all $p \in P$, let $\mathcal{X}(p)$ be the set of sectors of p. Abusing notation slightly, we may write $s_X(p)$ instead of $s_i(p)$ if $i \in X \in \mathcal{X}(p)$. Because each $X \in \mathcal{X}(p)$ is of the form $[a, b]$ for some $a, b \in \mathbb{N}$, we can sum the above inequality over $X \in \mathcal{X}(p)$ and $p \in Q$ to get

$$\sum_{p \in Q} \#\tau(p) \leq \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \\ X \in \mathcal{X}(p)}} (1 + r_{v,X} m_X(p, F(w))), \tag{6.2}$$

where each $r_{v,X} = \pm 1$ can depend on v and X.

Let $p \in Q$ and $q \in P$. If q is clumpy, then

$$M_{[0,T-1]}(p, q) = \sum_{X \in \mathcal{X}(p)} (s_X(p) m_X(p, q) - 1) + \sum_{X \in \mathcal{X}(q)} (s_X(q) m_X(q, p) - 1).$$

The -1 in each sum accounts for the $-\delta_i(p) - \delta_i(q)$ term in (5.1), the definition of M. If instead q is not clumpy, then $\mathcal{X}(q) = \{[0, T-1]\}$, so

$$M_{[0,T-1]}(p, q) = \sum_{X \in \mathcal{X}(p)} (s_X(p) m_X(p, q) - 1) + s_{[0,T-1]}(q) m_{[0,T-1]}(q, p).$$

However, $m_{[0,T-1]}(q, p) = 0$ because p and q have the same length and number of 1s.

Let $W = \{v \in V(G) \mid F(v) \in Q\}$ be the set of vertices with clumpy PFPs. Choosing $r_{v,X} = -s_X(F(v))$ and splitting the sum in (6.2) between neighbors with and without clumpy PFPs yields

$$\begin{aligned}
\sum_{\substack{p \in Q \\ X \in \mathcal{X}(p)}} \#\tau(p) &\leq \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \cap W \\ X \in \mathcal{X}(p)}} (1 - s_X(p) m_X(p, F(w))) + \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \setminus W \\ X \in \mathcal{X}(p)}} (1 - s_X(p) m_X(p, F(w))) \\
&= \sum_{\substack{p, q \in Q \\ e \in \pi(p,q)}} \left(\sum_{X \in \mathcal{X}(p)} (1 - s_X(p) m_X(p, q)) + \sum_{X \in \mathcal{X}(q)} (1 - s_X(p) m_X(q, p)) \right) \\
&\quad + \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \setminus W \\ X \in \mathcal{X}(p)}} (1 - s_X(p) m_X(p, F(w))) \\
&= -\sum_{\substack{p, q \in Q \\ e \in \pi(p,q)}} M_{[0,T-1]}(p, q) - \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \setminus W}} M_{[0,T-1]}(p, F(w)) \\
&\leq 0.
\end{aligned}$$

The last line follows from Theorem 5.1, and when we sum over $p, q \in Q$, we consider p and q unordered. (A sum over $\{p, q\} \subseteq Q$ is one alternative notation.) Sets have nonnegative sizes, so $\#\tau(p) = 0$ for all $p \in Q$. $\square$

7. Implications of Nonclumpiness. It is a basic property of the parallel chip-firing game that every vertex fires the same number of times each period [10]. This means, roughly speaking, that every periodic game is either "mostly waiting" with bursts of firing or "mostly firing" with bursts of waiting. (In fact, there is a bijection between them. Each periodic game has a complement that inverts firing and waiting [10].) This is because if a vertex waits twice in a row, then because it therefore never fires twice in a row, it fires less than half the time over the course of a period. Similarly, a vertex that fires twice in a row fires more than half the time. We cannot have a vertex that waits twice in a row and a vertex that fires twice in a row in the same periodic game because each vertex fires the same number of times each period.

Corollary 7.1. *Once a parallel chip-firing game reaches a periodic position, either no vertex fires twice in a row or no vertex waits twice in a row.*

That is, in periodic games, a firing sequence is possible if and only if it is nonclumpy.

Let the *interior* of a set of vertices W be $I(W) = \{v \in W \mid N(v) \subseteq W\}$. Because a waiting (or firing) vertex with only waiting (or firing) neighbors will wait (or fire) the following turn as well, the above observation proves the following conjecture of Fey and Levine [7].

Corollary 7.2. *Let σ be a periodic game on G,*

$$A = \{v \in V(G) \mid F_a(v) = 0\}$$
$$B = \{v \in V(G) \mid F_b(v) = 1\},$$

where $a, b \in \mathbb{N}$. Then either $I(A)$ or $I(B)$ is empty.

Interestingly, Corollary 7.2 also implies Theorem 6.2. If clumpy PFPs were possible, then a leaf attached to a motor with a clumpy PFP would be in $I(A)$ and $I(B)$ for appropriately chosen $a, b \in \mathbb{N}$.

In one of the first papers on the parallel chip-firing game, Bitar and Goles characterized parallel chip-firing games on trees in [4]. Corollary 3.3 and Theorem 6.2 allow us to characterize the behavior on tree-like subgraphs—subgraphs such that, if an edge to a root vertex is cut, become a tree separated from the rest of the graph—by making the root vertex a motor.

Corollary 7.3. *Let σ be a periodic game on G with $T \geq 3$ in which no vertex fires twice in a row, H be a tree-like subgraph of G and $m \in V(H)$ be the root of H. Then for all $v \in V(H)$,*

$$\sigma_t(v) = \begin{cases} d(v) & \text{if } F_{t-D}(m) = 1 \\ 0 & \text{if } F_{t-D-1}(m) = 1 \\ d(v) - 1 & \text{otherwise,} \end{cases}$$

where D is the distance from m to v. An analogous result holds if no vertex waits twice in a row.

In some sense, tree-like subgraphs are passive in that their vertices fire only in response to their root-side neighbor firing. In a periodic game, we can completely remove tree-like subgraphs without affecting the PFPs of the other vertices.

Corollary 7.4. *Let σ be a periodic game on G and $l \in V(G)$ be a leaf with $N(l) = \{m\}$. Suppose that no vertex fires twice in a row. Let G' be the subgraph of G induced by $V(G)\backslash\{l\}$. Define σ' on G' such that $\sigma'_0(v) = \sigma_0(v)$ for all $v \in V(G)\backslash\{m\}$.*

Then

$$\sigma'_0(m) = \begin{cases} \sigma_0(m) & \text{if } F_0^\sigma(l) = 1 \\ \sigma_0(m) - 1 & \text{otherwise.} \end{cases}$$

An analogous result holds if no vertex waits twice in a row.

Compared to σ', m has to have an extra chip to fire in σ. However, unless m fired the previous turn—which, because l is a leaf, is equivalent to saying l is firing this turn—m will have received the extra chip back from l, so removing both l and the chip has no effect on m as long as m does not fire while l has a chip. This corollary concerns a leaf, though the result generalizes to all tree-like subgraphs by repeated application, providing an alternate proof of Corollary 3.3.

8. Discussion and Directions for Future Work. We have introduced motors, studied motorized games on trees, and shown that motor-like behavior can be constructed in ordinary games, provided that each motor has a possible firing sequence. We then showed that periodic firing patterns are possible if and only if they are nonclumpy, which, among other things, allows classification of periodic games as "mostly waiting" or "mostly firing" and the removal of tree-like subgraphs without loss of generality.

We might expect that the space of motorized games be larger than that of ordinary games. Theorem 4.1 shows us that, as long as the firing sequences involved are possible, the parallel chip-firing game is in some sense just as "expressive" as its motorized variant. This allows, for example, the simulation of some aspects of the *dollar game*, a variant of the general chip-firing game discussed by Biggs [3]. In the dollar game, exactly one vertex, the "government", may have a negative number of chips and fires if and only if no other vertices can fire. We can construct a motorized parallel chip-firing game in which we replace the government with a motor that waits a sufficiently large number of steps between each firing such that it never fires in the same step as another vertex. Biggs showed that every dollar game tends towards a critical position regardless of the order of vertex firings, so this motorized parallel chip-firing game tends towards the same critical position. Theorem 4.1 may help reveal the extent to which the parallel chip-firing game can simulate additional aspects of the dollar game and other general chip-firing games.

Despite the expressiveness we get due to motors, the nonclumpiness of firing patterns tells us that the parallel chip-firing game is "easier" than its rules explicitly tell us it must be. In addition to results mentioned in Section 7, Theorem 6.2 is a step towards reducing the parallel chip-firing game to one of interacting "gliders". For example, consider the situation in Corollary 3.3. Intuitively, we can think of this corollary as stating that each firing of the motor creates a wave of gliders that travels away from the motor. In fact, the corollary, together with Theorem 6.2, implies that every periodic position on tree-like subgraphs must be the result of such gliders, providing a new test that can diagnose some positions that are never repeated. Every game on a simple cycle with period at least 3 can be described by gliders [6]. (See Figure 8.1.) We believe that this approach could be used to analyze periodic behavior of games on further classes of graphs, such as those in which each vertex is in at most once cycle.

Nonclumpiness is essentially an unwritten rule of periods in the parallel chip-firing game, which is unusual because no local property of the firing mechanic disallows clumpiness. By contrast, in other graph automata that are more restrictive than the

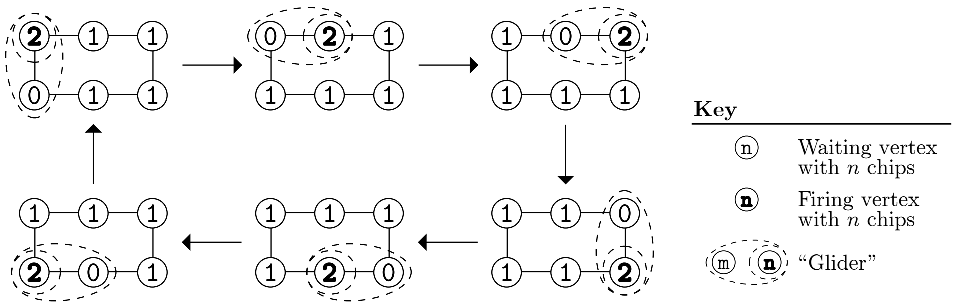

Fig. 8.1: A game on a 6-cycle in which a glider orbits once each period.

parallel chip-firing game, such as *source reversal* [9] (essentially a parallel chip-firing game with exactly one chip bound to each edge), nonclumpiness is obvious, even locally. In the other direction, motors make it simple to show that certain stronger restrictions do not apply to the parallel chip-firing game. For example, a path where the leaves are motors can yield a game in which some chips cannot be bound to a single edge, which is is a property of source reversal. We might ask which restrictions apply to which chip-firing-style games. Is the parallel chip-firing game on undirected graphs the most general game to which an analogue of Theorem 6.2 applies?

We hope that the intuition and constructive powers of motors and the reduction in the space of possible periodic games provided by nonclumpiness prove useful in further research.

Appendix A. The Bellman-Ford Algorithm. The Python program below calculates $\mathscr{G}$ from Section 5 and shows it has no negative cycles. The authors have also confirmed this by hand. For a more detailed description of the Bellman-Ford algorithm and a proof of its validity, see [5].

```
infty = float("inf")

class Vertex:
    def __init__(self):
        self.distance = infty
    __hash__ = object.__hash__

class Edge:
    def __init__(self, tail, head, weight):
        self.tail = tail
        self.head = head
        self.weight = weight

def verts(edges):
    return set(vert for e in edges for vert in (e.tail, e.head))

# Returns True if graph has a negative cycle (or has unreachable vertices).
def bellmanFord(edges):
    # For us, the start vertex is arbitrary.
```

```
    edges[0].tail.distance = 0
    # Find minimum length from the start vertex to each other vertex.
    for i in range(len(verts(edges)) - 1):
        for e in edges:
            newDistance = e.tail.distance + e.weight
            if newDistance < e.head.distance:
                e.head.distance = newDistance
    # Find triangle inequality failures, which are caused by negative cycles.
    # Also confirms we've searched the whole graph (infty < infty == False).
    for e in edges:
        if e.tail.distance + e.weight < e.head.distance:
            return True
    return False

# Throughout, we use 0 instead of -1 for s_i(p), correcting for it when needed.

# Checks if mu_i(p,q) is compatible with the definition of a sector.
def validVert(mu1):
    (p0, p1, s1, s2, q0, q1, t1, t2) = mu1
    return (p0==s1 if p0==p1 else s1==s2) and (q0==t1 if q0==q1 else t1==t2)

# Checks if the first state can be followed by the second.
def validEdge(mu1, mu2):
    (p0a, p1a, s1a, s2a, q0a, q1a, t1a, t2a) = mu1
    (p1b, p2b, s2b, s3b, q1b, q2b, t2b, t3b) = mu2
    return p1a==p1b and q1a==q1b and s2a==s2b and t2a==t2b

# Calculates M_{i}(p,q)
def weight(mu1):
    (p0, p1, s1, s2, q0, q1, t1, t2) = mu1
    return (2*s1-1)*(p1-q0) + (2*t1-1)*(q1-p0) - abs(s1-s2) - abs(t1-t2)

bits = {x : tuple((x // 2**i) % 2 for i in range(8)) for x in range(2**8)}
vs = {bits[i]: Vertex() for i in range(2**8) if validVert(bits[i])}
G = [Edge(vs[u], vs[v], weight(u)) for u in vs for v in vs if validEdge(u, v)]

# And Theorem 5.1 is...
print(not bellmanFord(G))
```

Appendix B. We thank the MIT PRIMES program for supporting the research that brought us together. We thank Tiankai Liu for improving upon a previous proof of Theorem 5.1. We would also like to thank Anne Fey, Lionel Levine and Tanya Khovanova for helpful discussions. Yan X Zhang was supported by an NSF Graduate Fellowship.

REFERENCES

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

[2] Per Bak, Chao Tang, and Kurt Wiesenfeld, *Self-organized criticality*, Phys. Rev. A **38** (1988), 364–374.

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

[4] Javier Bitar and Eric Goles, *Parallel chip firing games on graphs*, Theoret. Comput. Sci. **92** (1992), no. 2, 291–300.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms, Third Edition*, MIT Press, 2009.

[6] Luca Dall'Asta, *Exact solution of the one-dimensional deterministic fixed-energy sandpile*, Phys. Rev. Lett. **96** (2006), 058003.

[7] Anne Fey and Lionel Levine, private correspondence, 2011.

[8] Eric Goles and Maurice Margenstern, *Universality of the chip-firing game*, Theoret. Comput. Sci. **172** (1997), no. 1-2, 121–134.

[9] Eric Goles and Erich Prisner, *Source reversal and chip firing on graphs*, Theoret. Comput. Sci. **233** (2000), no. 12, 287 – 295.

[10] Tian-Yi Jiang, *On the period lengths of the parallel chip-firing game*, ArXiv e-prints (2010).

[11] Marcos A. Kiwi, René Ndoundam, Maurice Tchuente, and Eric Goles, *No polynomial bound for the period of the parallel chip firing game on graphs*, Theoret. Comput. Sci. **136** (1994), no. 2, 527–532.

[12] Paul M. Kominers and Scott D. Kominers, *Candy-passing games on general graphs, II*, ArXiv e-prints (2008).

[13] Lionel Levine, *Parallel chip-firing on the complete graph: Devil's staircase and Poincaré rotation number*, Ergodic Theory Dynam. Systems **31** (2010), no. 3, 891.