

Firing Patterns in the Parallel Chip-Firing Game

Ziv Scully, Tian-Yi Jiang, Yan Zhang

► **To cite this version:**

Ziv Scully, Tian-Yi Jiang, Yan Zhang. Firing Patterns in the Parallel Chip-Firing Game. Louis J. Billera and Isabella Novik. 26th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2014), 2014, Chicago, United States. Discrete Mathematics and Theoretical Computer Science, DMTCS Proceedings vol. AT, 26th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2014), pp.537-548, 2014, DMTCS Proceedings. <hal-01207550>

HAL Id: hal-01207550

<https://hal.inria.fr/hal-01207550>

Submitted on 1 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Firing Patterns in the Parallel Chip-Firing Game

Ziv Scully^{1*}Tian-Yi Jiang^{1†}Yan X Zhang^{2‡}¹Massachusetts Institute of Technology, Cambridge, MA, USA²Department of Mathematics, UC Berkeley, Berkeley, CA, USA

Abstract.

The *parallel chip-firing game* is an automaton on graphs in which vertices “fire” chips to their neighbors. This simple model, analogous to sandpiles forming and collapsing, contains much emergent complexity and has connections to different areas of mathematics including self-organized criticality and the study of the sandpile group. In this work, we study *firing sequences*, which describe each vertex’s interaction with its neighbors in this game. Our main contribution is a complete characterization of the periodic firing sequences that can occur in a game, which have a surprisingly simple combinatorial description. We also obtain other results about local behavior of the game after introducing the concept of *motors*.

Résumé. Le *parallel chip-firing game*, c’est une automate sur les graphiques, dans lequel les sommets “tirent” des jetons à leurs voisins. Cette modèle simple, semblable aux tas de sable qui forment et s’effaissent, contient beaucoup de complexité émergeant et a des connections à différents domaines de mathématiques, incluant le *self-organized criticality* et l’étude du *sandpile group*. Dans cette projet, on étudie les *firing sequences*, qui décrivent les interactions de chaque sommet avec ses voisins dans le jeu. Notre contribution principale est une caractérisation complète des séquences de tir qui peuvent arriver dans une jeu, qui ont une description combinatorial assez simple. Nous obtenons aussi des autres résultats sur le conduite locale du jeu après l’introduction du concept des *motors*.

Keywords: chip-firing, combinatorics, finite automata, graph theory, sandpile model

1 Introduction

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 Biggs (1999). In computer science, it is able to simulate any two-register machine and is thus universal Goles and Margenstern (1997). Strongly related to the *abelian sandpile model*, which is itself a generalization of a sandpile

*Email: ziv@mit.edu

†Email: jiangty@mit.edu

‡Email: yanzhang@math.berkeley.edu. Was supported by an NSF Graduate Fellowship.

model introduced by Bak, Tang, and Wiesenfeld Bak et al. (1987, 1988) in the study of self-organized criticality, the parallel chip-firing game has even more links with other fields.

The game is played on a graph as follows (a detailed description will be given in Section 2): 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* one chip along each of its edges to the corresponding neighbor and loses the chip; otherwise, it *waits* and does nothing. All vertices that fire in a particular turn do so in parallel, before receiving any chips that their neighbors may fire. An example game is illustrated in Figure 1.

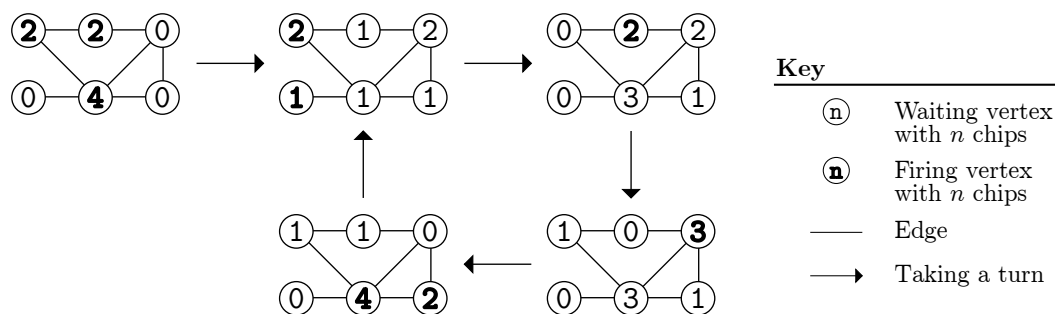


Fig. 1: A parallel chip-firing game. From an initial position in the upper left, the game eventually enters a period of length 4.

It is easy to see that 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.

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 Bitar and Goles (1992), simple cycles Dall'Asta (2006), the complete graph Levine (2008), and the complete bipartite graph Jiang (2010). For these graphs, the maximum period lengths are bounded by the number of vertices, but Kiwi et al. Kiwi et al. (1994) 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 Kominers and Kominers (2008) showed that games with a sufficiently large density of chips must have period 1. Dall'Asta Dall'Asta (2006) and Levine Levine (2008), 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 Dall'Asta (2006) characterized the periodic positions of games on cycles.

In this paper, which is an extended abstract for a longer article (Scully et al. (2013)), we hope to advance the understanding of both of these questions through the study of firing sequences and periodic firing patterns. Our main result is a complete characterization of the possible periodic firing patterns in parallel

chip-firing games in Section 3, which has some further implications, including settling a conjecture of Fey and Levine. We also develop a new tool for studying the chip-firing game: *motors*, which allow us to study the behavior of subgraphs in the game in Section 4. We conclude with some observations and comments on future work.

2 Definitions

Here we will only consider games on finite, undirected, connected graphs, though the definition of the game can be easily generalized for arbitrary multidigraphs. We use $N(v) = \{w \in V(G) \mid \{v, w\} \in E(G)\}$ to denote the *neighbors* of a vertex v , and $d(v) = \#N(v)$ to denote its *degree*.

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

$$F_t(v) = \begin{cases} 0 & \text{if } \sigma_t(v) \leq d(v) - 1 \\ 1 & \text{if } \sigma_t(v) \geq d(v) \end{cases}$$

After each turn, $\sigma_t(v)$ decreases by $d(v)$ if $F_t(v) = 1$ and then increases by 1 for each $w \in N(v)$ such that $F_t(w) = 1$. Thus, σ_t induces σ_{t+1} completely, so an initial position suffices to define a game on a given graph. The *firing sequence* of a vertex v in game σ is the sequence $(F_t(v))_{t \in \mathbb{N}}$. When $F_t(v) = 0$, we say v *waits* at t , and when $F_t(v) = 1$, we say v *fires* at t . We call a firing sequence $(f_t)_{t \in \mathbb{N}}$ *possible* if there exists a game with some vertex v such that $F_t(v) = f_t$ for all $t \in \mathbb{N}$.

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 $\{t, t + 1, \dots, t + T - 1\}$, 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 $t \geq t_0$. If the initial position of a game is periodic, we may also call the game itself periodic. Similarly to the firing sequence, the *periodic firing pattern* (PFP) of a vertex $v \in V(G)$ is the binary string $F_{t_0}(v) \dots F_{t_0+T-1}(v)$, where t_0 is the smallest natural number such that σ_{t_0} is periodic⁽ⁱ⁾. We write the PFP of v as $F(v)$. We also call a PFP *possible* if there exists a game with some vertex v that exhibits the given PFP.

3 Characterization of Periodic Firing Patterns

It turns out that we can characterize completely the possible periodic firing patterns. First, we define some notions for periodic binary strings (i.e. $p_0 p_1 \dots p_{n-1}$, where $p_i \in \{0, 1\}$ and all indices are considered modulo n). Our main result, Theorem 3.1, will be stated in terms of these notions. Throughout this section, $b \in \{0, 1\}$ and $\bar{b} = 1 - b$.

Given a binary string p , we say an index i is in a *b-region* of p if either $p_{i-1} = p_i = b$ or $p_{i-1} \neq p_i$ and p_{i+1} is in a *b-sector*. A *b-sector* of p is a maximal integer interval such that all of its member indices are

⁽ⁱ⁾ 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 the latter, which has period 4, might occur in the same game as the PFP 0011, the former, which has period 2, cannot.

$\overbrace{01000100} \quad \overbrace{11011} \quad \overbrace{00101011}$

Fig. 2: A string's 0-sectors (marked below) and 1-sectors (marked above).

in a b -region of p . Roughly speaking, b -sectors have no two \bar{b} s in a row and extend as far back as possible. See Figure 2 for an example. 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.

Call such a binary string p *clumpy* if the above decomposition into sectors gives at least 2 sectors. In other words, p has both a pair of consecutive 0s and a pair of consecutive 1s as substrings. Recall that the T^{th} and 0^{th} entries of a PFP are the same, so, for example, 011010 is clumpy. Our main result is as follows:

Theorem 3.1 *A clumpy binary string cannot be a PFP.*

It is known that, given almost any⁽ⁱⁱ⁾ nonclumpy PFP, one can construct a parallel chip-firing game on a simple cycle in which every vertex has that PFP shifted by some number of steps Dall'Asta (2006). Thus, our result gives a complete classification of possible PFPs in the parallel chip-firing game.

Let

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

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

Our main lemma for proving Theorem 3.1 concerns the sum

$$M_S(p, q) = \sum_{i \in S} (s_i(p)(p_i - q_{i-1}) + s_i(q)(q_i - p_{i-1}) - \delta_i(p) - \delta_i(q)), \quad (3.1)$$

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 lemma says that sector switches require disagreement:

Lemma 3.2 *We must have $M_{[0, n-1]}(p, q) \geq 0$.*

Proof idea: ⁽ⁱⁱⁱ⁾ We can calculate $M_{[0, n-1]}(p, q)$ 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

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

⁽ⁱⁱ⁾ The given construction requires that the PFP not be decomposable to a repeated substring. Using Theorem 4.4, 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.

⁽ⁱⁱⁱ⁾ We thank Tiankai Liu for this much-improved proof.

and let \mathcal{G} be a weighted digraph with

$$\begin{aligned} V(\mathcal{G}) &= \{\mu_i(p, q) \mid p, q \text{ strings}, i \in \mathbb{N}\} \\ E(\mathcal{G}) &= \{(u, v, w) \mid \exists p, q, i: u = \mu_i(p, q), v = \mu_{i+1}(p, q), w = M_{\{i\}}(p, q)\}. \end{aligned}$$

(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 cycle negative if it has negative weight. We can calculate $M_{[0, n-1]}(p, q)$ as the weight of a path induced by the sequence of vertices $(\mu_0(p, q), \dots, \mu_{n-1}(p, q), \mu_0(p, q))$. Therefore, it suffices to show that \mathcal{G} has no negative cycles. Running the Bellman-Ford algorithm Cormen et al. (2009) on \mathcal{G} shows this to be the case. We describe \mathcal{G} and reproduce the algorithm fully as a Python program in our longer work Scully et al. (2013). The key is that if i and $i + 1$ are in b - and \bar{b} -sectors of p , respectively, then $p_{i-1} = p_i = b$ and $p_{i+1} = \bar{b}$. \square

We need another lemma to bound the discrepancy between a vertex's activity and its neighbors' activity. The proof is a short computation and will be omitted:

Lemma 3.3 *For all $v \in V(G)$ and $a, b \in \mathbb{N}$, let $\Phi(v, t) = \sum_{w \in N(v)} F_{t-1}(w)$. Then*

$$-d(v) + 1 \leq \sum_{t=a}^b (\Phi(v, t) - d(v)F_t(v)) \leq d(v) - 1. \quad (3.2)$$

Our main result, Theorem 3.1 now follows.

Proof idea: Roughly, summing an inequality given by Lemma 3.3 over all vertices with clumpy PFPs bounds a quantity below by the number of vertices with clumpy PFPs, and summing an inequality given by Lemma 3.2 over all edges incident with a vertex with a clumpy PFP gives an upper bound of 0 on the same quantity.

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 . For a parallel chip-firing game σ on undirected graph G with period T , let P be the set of all PFPs occurring in σ and let Q be the set of all clumpy PFPs occurring in σ . 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 group the sum in Lemma 3.3 to eventually obtain:

$$\sum_{p \in Q} \#\tau(p) \leq \sum_{\substack{p \in Q \\ v \in \tau(p) \\ w \in N(v) \\ X \in \mathcal{X}(p)}} (1 + r_{v, X} m_X(p, F(w))), \quad (3.3)$$

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 (3.1), the definition of M . If instead q is not clumpy, then $\mathcal{X}(q) = \{[0, T - 1]\}$, and one can observe that the second term vanishes.

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 (3.3) between neighbors with and without clumpy PFPs yields

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

The last expression is nonpositive because of Lemma 3.2. Sets have nonnegative sizes, so $\#\tau(p) = 0$ for all $p \in Q$. □

It is a basic property of the parallel chip-firing game that every vertex fires the same number of times each period Jiang (2010). 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 these two types of games. Each periodic game has a complement that inverts firing and waiting Jiang (2010).) 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 3.4 *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 Fey and Levine (2011).

Corollary 3.5 *Let σ be a periodic game on G ,*

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

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

Interestingly, Corollary 3.5 also implies Theorem 3.1. If clumpy PFPs were possible, then a leaf attached to a motor (see Section 4) with a clumpy PFP would be in $I(A)$ and $I(B)$ for appropriately chosen $a, b \in \mathbb{N}$.

4 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.

A *motorized parallel chip-firing game*, or simply “motorized game”, on G is a game σ obeying the same rules as an ordinary game with the exception of 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. In this section, we use motors to isolate the study of local behavior. Our two main results have intuitive physical flavors: Theorem 4.2 shows that the firing patterns of a tree with one motor is “driven” by the motor; Theorem 4.4 shows that a motorized game can be simulated by an ordinary game (after attaching enough copies of the motor to have enough “activation energy”) so we are not needlessly introducing complexity.

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 (again, recall that we consider in this paper only connected graphs; see, for example, Jiang (2010)).

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 $F_{a-1}(v) = F_{b+1}(v) = \bar{f} = 1 - f$, 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 our first result, Theorem 4.2, follows the same structure as the proof that ordinary games on trees have period 1 or 2 Bitar and Goles (1992). In fact, we mostly rely on a lemma originally introduced for that proof:

Lemma 4.1 ((Bitar and Goles, 1992, 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 $[a - 1, b - 1] \subseteq R_f(w)$.*

In other words, every burst of firing or waiting by a vertex must be supported by at least one of its neighbors. This lemma follows from the pigeonhole principle and Lemma 3.3. We then obtain:

Theorem 4.2 *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 .*

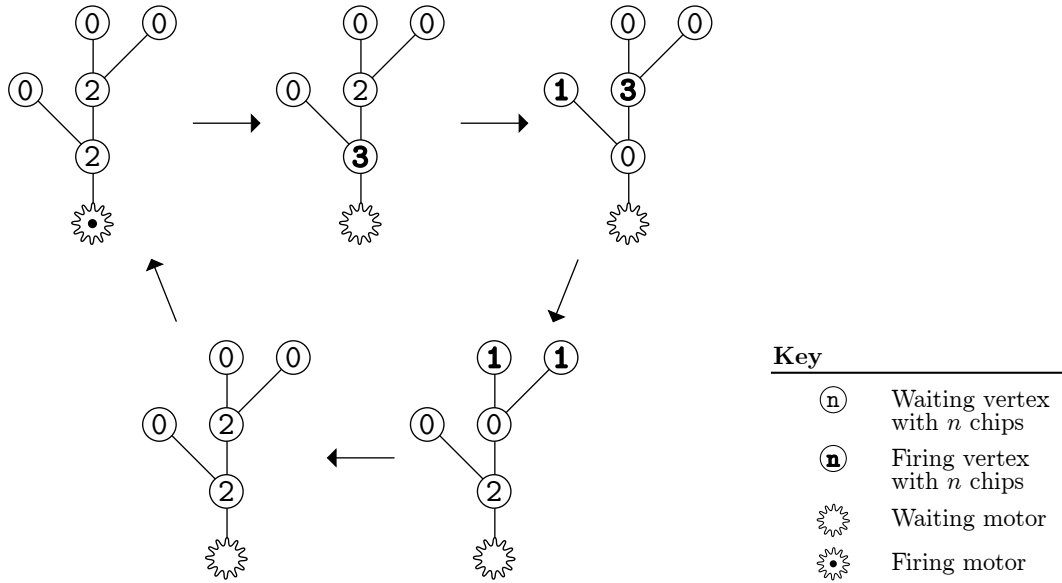


Fig. 3: A motorized parallel chip-firing game. The motor has firing sequence $(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, \dots)$.

Qualitatively, Theorem 4.2 is stating that in a tree, the firing of a non-motor vertex is “driven” by some motor. In the case where we have just one motor, this implies that all vertices copy the firing pattern of the motor (with a time delay):

Corollary 4.3 *Let σ be a periodic motorized game on a tree T 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 .*

For our second result, to refer to multiple chip-firing games unambiguously, we include the subscripts and superscripts in, for example, $d_G(v)$ and $F_t^\sigma(v)$. 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 demonstrates the concept.

Theorem 4.4 *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 .*

Proof idea: The proof is somewhat technical, but the main idea is very simple: for each $m \in M^\sigma$, we 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. The analogy of “motors” is quite appropriate: once we have enough “motors,” their combined strength is enough to drive the vertex to behave exactly like them. \square

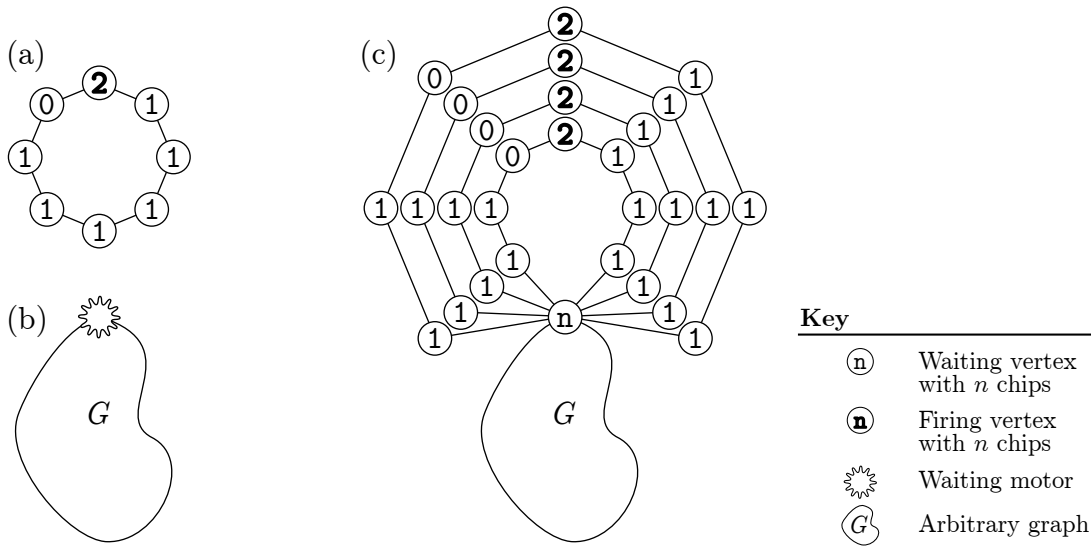


Fig. 4: Suppose the motor in motorized game (b) has firing sequence $(0, 0, 0, 0, 1, 0, 0, 0, \dots)$. This occurs in ordinary game (a). By using sufficiently many copies of (a) and carefully choosing n , we construct (c). The behavior of G in (c) is identical to the behavior of G in (b).

In Theorem 4.2, 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 4.4 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 Combining Nonclumpiness and Motors

In one of the first papers on the parallel chip-firing game, Bitar and Goles characterized parallel chip-firing games on trees in Bitar and Goles (1992). In this section, we explore a similar result on “tree-like” subgraphs.

We say that a subgraph H of G is a *tree-like* if (1) H is an induced subgraph of G , (2) H is a tree, and (3) exactly one vertex of H is adjacent to exactly one vertex of G outside H . Corollary 4.3 and Theorem 3.1 allow us to characterize the behavior on tree-like subgraphs by making the root vertex a motor.

Corollary 5.1 *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 5.2 *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 $V(G) \setminus \{l\}$. Define σ' on G' such that $\sigma'_0(v) = \sigma_0(v)$ for all $v \in V(G) \setminus \{m\}$. Then $\sigma'_0(m) = \sigma_0(m)$ if $F_0(l) = 1$ and $\sigma_0(m) - 1$ otherwise. An analogous result holds if no vertex waits twice in a row.*

Compared to σ' , vertex 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 4.3.

6 Discussion and Directions for Future Work

We 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.” We then introduced motors, studied motorized games on trees, and showed that motor-like behavior can be constructed in ordinary games, provided that each motor has a possible firing sequence. The two concepts combine to allow us to remove tree-like subgraphs without loss of generality.

We might expect the space of motorized games to be larger than that of ordinary games. Theorem 4.4 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 Biggs (1999). 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. 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 4.4 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. Theorem 3.1 is also a step towards reducing the parallel chip-firing game to one of interacting “gliders”. For example, consider the situation in Corollary 4.3. 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 3.1, 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 Dall’Asta (2006). (See Figure 5.) 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 parallel chip-firing game, such as *source reversal* Goles and Prisner (2000) (essentially a parallel chip-firing game with exactly one chip bound to each edge),

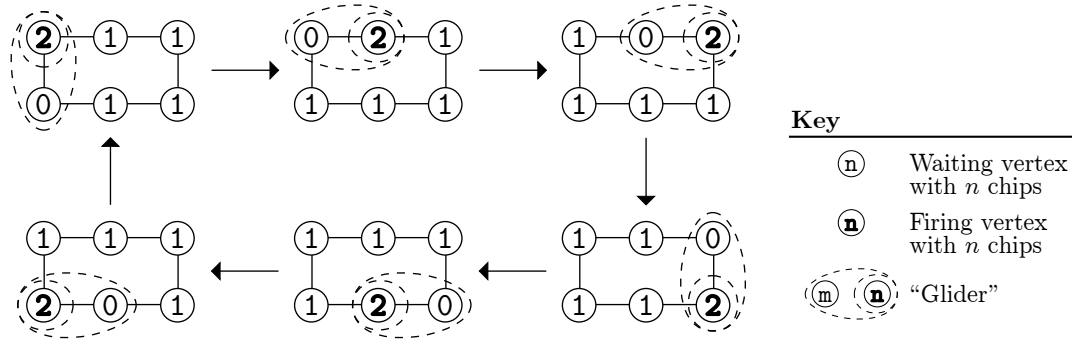


Fig. 5: A game on a 6-cycle in which a glider orbits once each period.

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 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 3.1 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.

Acknowledgements

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 Lemma 3.2, and Anne Fey, Lionel Levine, and Tanya Khovanova for helpful discussions. Finally, we would like to thank two anonymous referees who greatly helped improve the content and readability of this paper. Yan X Zhang was supported by an NSF Graduate Fellowship.

References

P. Bak. *How Nature Works: The Science of Self-organized Criticality*. Copernicus (Springer), 1st edition, Aug. 1996. ISBN 0387947914.

P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of the $1/f$ noise. *Physical Review Letters*, 59:381–384, Jul 1987.

P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. *Physical Review A*, 38:364–374, Jul 1988.

N. L. Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9:25–45, January 1999. ISSN 0925-9899.

J. Bitar and E. Goles. Parallel chip firing games on graphs. *Theoretical Computer Science*, 92(2):291–300, Jan. 1992. ISSN 0304-3975.

- A. Björner, L. Lovász, and P. W. Shor. Chip-firing games on graphs. *European Journal of Combinatorics*, 12:283–291, 1991.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. MIT Press, 2009. ISBN 0262533057.
- L. Dall’Asta. Exact solution of the one-dimensional deterministic fixed-energy sandpile. *Physical Review Letters*, 96:058003, Feb 2006.
- D. Dhar. The abelian sandpile and related models. *Physica A: Statistical Mechanics and its Applications*, 263(1-4):4–25, 1999. ISSN 0378-4371. Proceedings of the 20th IUPAP International Conference on Statistical Physics.
- A. Fey and L. Levine. private correspondence, 2011.
- A. Fey, L. Levine, and D. B. Wilson. Approach to criticality in sandpiles. *Physical Review E*, 82:031121, Sep 2010.
- E. Goles and M. Margenstern. Universality of the chip-firing game. *Theoretical Computer Science*, 172(1-2):121–134, 1997.
- E. Goles and E. Prisner. Source reversal and chip firing on graphs. *Theoretical Computer Science*, 233(1-2):287–295, 2000. ISSN 0304-3975.
- T.-Y. Jiang. On the period lengths of the parallel chip-firing game, Mar. 2010. [arXiv:math.CO/1003.0943](https://arxiv.org/abs/math/1003.0943).
- M. A. Kiwi, R. Ndoundam, M. Tchunte, and E. Goles. No polynomial bound for the period of the parallel chip firing game on graphs. *Theoretical Computer Science*, 136(2):527–532, 1994.
- P. M. Kominers and S. D. Kominers. Candy-passing games on general graphs, II, July 2008. [arXiv:math.CO/0807.4655](https://arxiv.org/abs/math/0807.4655).
- L. Levine. Parallel chip-firing on the complete graph: Devil’s staircase and Poincaré rotation number, Nov. 2008. [arXiv:math.CO/0811.2800](https://arxiv.org/abs/math/0811.2800).
- Z. Scully, T.-Y. Jiang, and Y. X. Zhang. Motors and Impossible Firing Patterns in the Parallel Chip-Firing Game, 2013. Submitted, [arXiv:math.CO/1211.6786](https://arxiv.org/abs/math/1211.6786).