The Computational Strength of Chip-Firing Games

Justin Miller
justin.d.miller@dartmouth.edu,
Dartmouth College

Joint with: David Belanger, Damir Dzhafarov,
Liling Ko, and Reed Solomon

April 20th, 2024

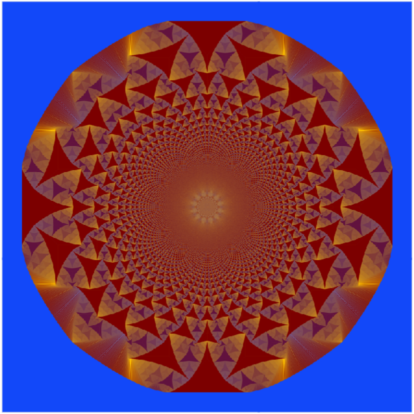
AMS 2024 Special Session
Computability Theory II

Origins of Chip-Firing

The process of chip-firing can be thought of as either a single player game or as a dynamical system. One of the earliest variants of chip-firing is the *abelian sandpile model*, which was motivated by interest in self-organized criticality in physics.

In this setting, we think of the count at each vertex as grains of sand, where the sand slides down if too many grains accumulate at one site. The firing effect generates aesthetic fractal images.

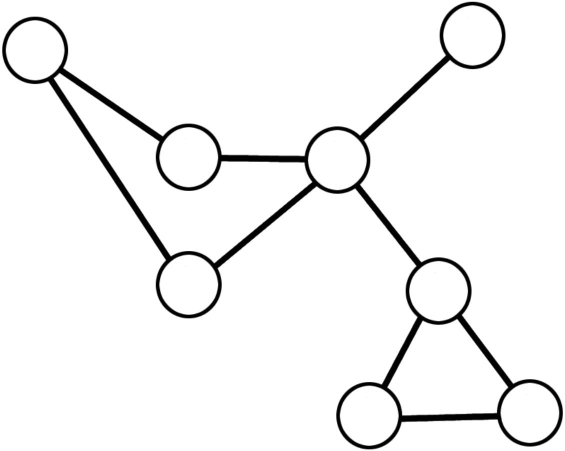
Abelian Sandpiles



The result of starting with 2^{30} chips at the origin of a 2D grid.
[Pegden, Wesley, 2024]

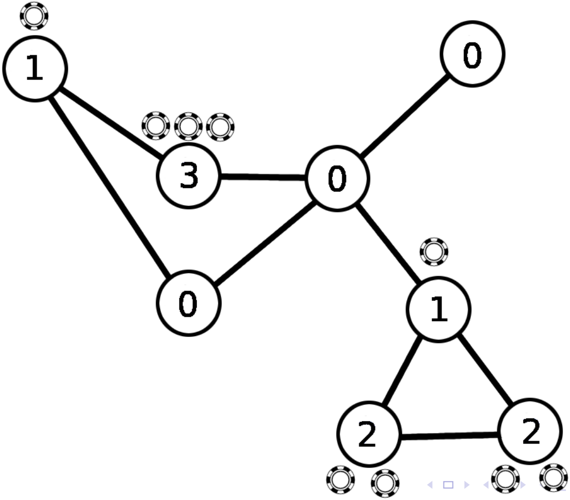
Chip-firing

We start with a graph G with vertices V and edge set E.



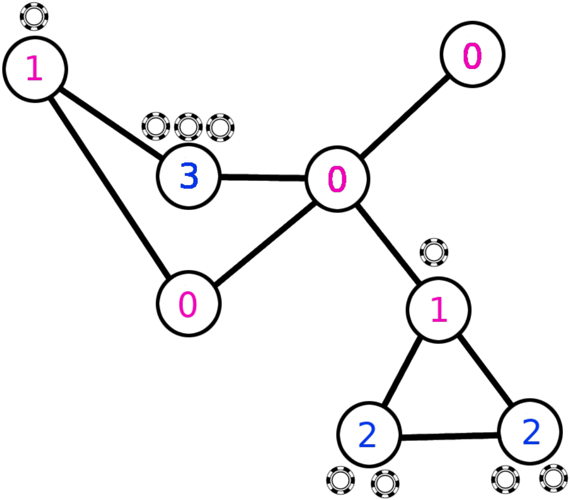
Chip-firing

We have an *initial configuration* of chips, which is a function $c : V \to \omega$ which assigns a natural number of chips to each vertex.



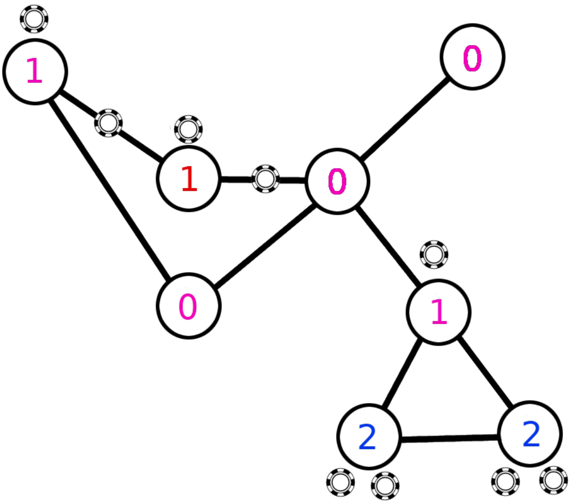
Chip-firing

A vertex is *fireable* if it has at least as many chips as neighbors.



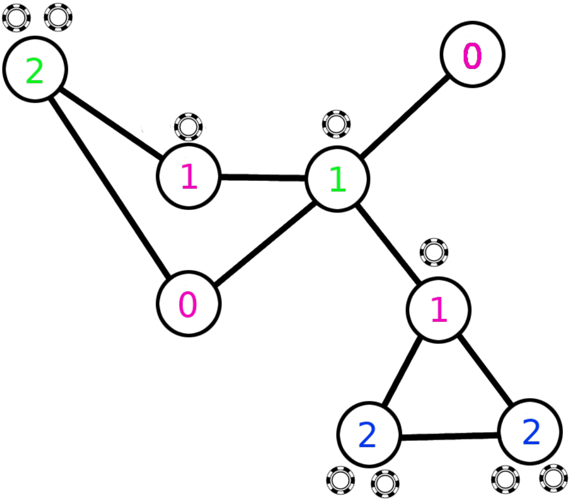
Chip-firing

When a vertex is fired, or played, it sends one chip to each neighbor.



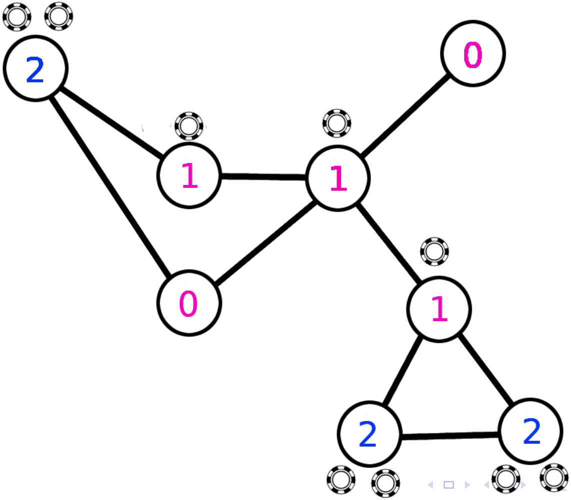
Chip-firing

When a vertex is fired, or played, it sends one chip to each neighbor.



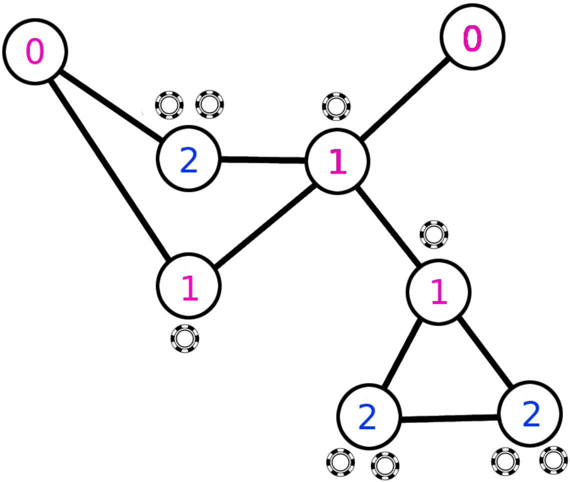
Chip-firing

After the vertex is fired, we have a new chip configuration for which the same rules apply.



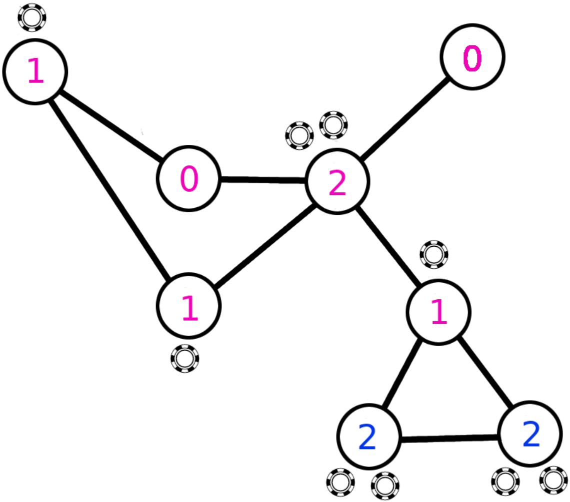
Chip-firing

We continue playing while there are playable vertices.



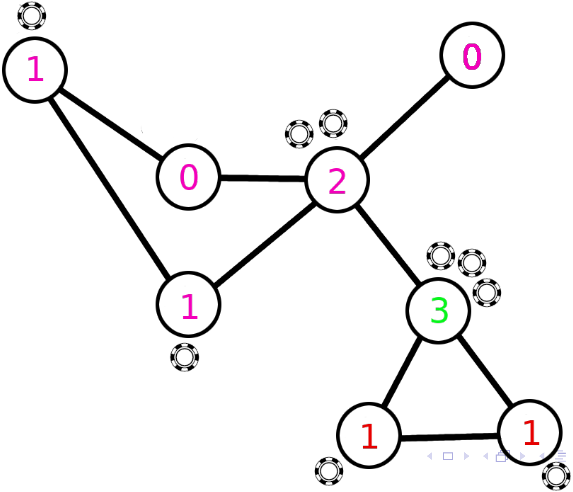
Chip-firing

We continue playing while there are playable vertices.



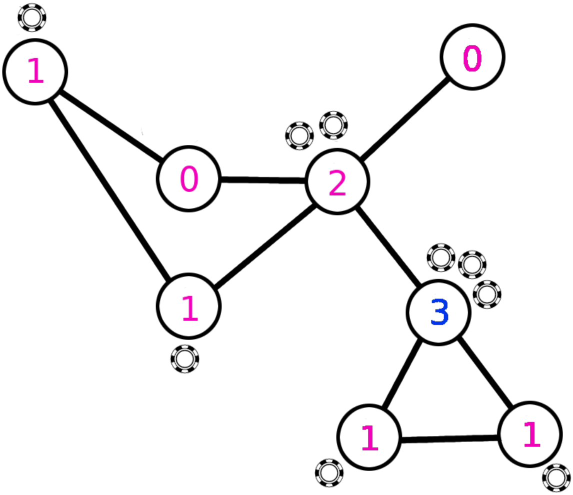
Chip-firing

If there are multiple vertices playable at once, it is straightforward to prove that they can fire in either order, or simultaneously, and achieve the same result.



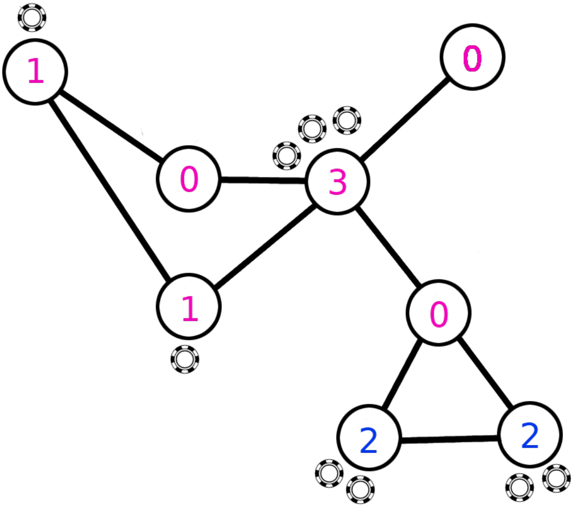
Chip-firing

If the game continues indefinitely, then we (the player) win.



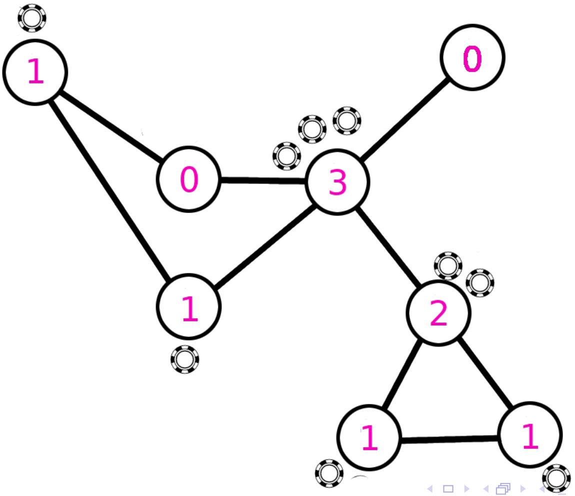
Chip-firing

If the game continues indefinitely, then we (the player) win.



Chip-firing

If the game reaches a stage where no vertices can be played, the house wins. (I.e., we lose.)



The Formal Game of Chip-Firing

An instance of the *chip-firing game* is an ordered pair $\langle G, c \rangle$ where G is a graph with vertices V and edges E, and $c : \omega \to V$ is an initial configuration of chips.

A *winning play* for $\langle G, c \rangle$ is a function $p : \omega \to V$ such that $p(0)$ is playable in c, and for all n, $p(n)$ is playable after all previous $p(i)$'s have been played. $\langle G, c \rangle$ is winnable if it has a winning play.

For our purposes, we are interested in countable, locally finite G. We wish to analyze the Turing degree of p compared to the degrees of G and c, particularly if the latter two are computable.

Not Losing the Game

Theorem ([Klivans, 2018])

If c_1 and c_2 are legal states of the chip-firing game $\langle G, c\rangle$ reached by firing finitely many vertices, then there is a a legal state d which can be reached from both c_1 and c_2.

This is known as the *confluence* property of chip-firing. A fundamental corollary: there are no “bad” plays. Unlike other games where certain plays can lose you the game, firing a vertex in this variant of the chip-firing game can never render a winnable state unwinnable.

Firing a vertex can only render the fired vertex unplayable: all other vertices either remain static or gain chips. In other words, all other vertices are as fireable as they were on the previous stage.

Strongly Locally Finite Graphs

Definition

A graph G is strongly locally finite if there is a total computable function $f : \omega \to \omega$ such that there is an edge between v_i and v_j only if $j \leq f(i)$.

Lemma (Belanger, Dzhafarov, Ko, M., Solomon)

If $\langle G, c \rangle$ is a computable, winnable chip-firing game for G strongly-locally finite graph, then there is a computable winning play.

Proof.

Since G is strongly-locally finite, it is uniformly computable to determine which vertices are playable at each state in the game. Since $\langle G, c \rangle$ is winnable, there is a playable vertex at any turn during the game. Therefore, the computable winning play is to play the least playable vertex on each turn. □

Locally Finite Graphs

If G is not strongly-locally finite, then it is not computable in general to determine which vertices are playable on a given turn. However, it is computable from the graph's jump and c.

Lemma (Belanger, Dzhafarov, Ko, M., Solomon)

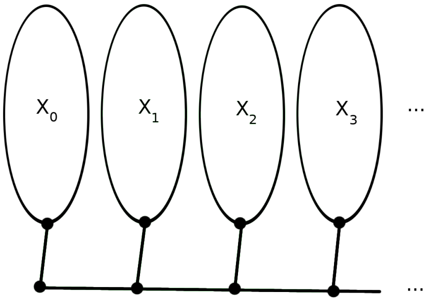
If A is computed by every winning play of some winnable, computable chip-firing game $\langle G, c\rangle$, then A is Δ^0_2.

Theorem (Belanger, Dzhafarov, Ko, M., Solomon)

If A is Δ^0_2, then there is a computable, winnable chip-firing game $\langle G, c\rangle$ such that the Turing degrees of winning plays for $\langle G, c\rangle$ form exactly the cone above A.

Proof Idea

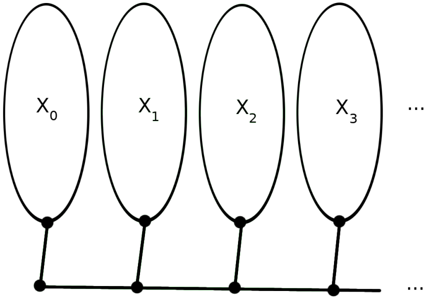
Given a Δ^0_2-set X, we construct a computable graph G like the picture below and a computable chip configuration c on G with the property that the subgraph X_n contains a single playable vertex if $n \in X$, and no other vertices are ever playable.



However, a winning strategy for this game computes an infinite subset of X, not necessarily X itself.

Fixing the Proof

Given a Δ^0_2-set A, we construct the same graph as before for $X = graph(f)$, where f is a Δ^0_2 modulus for A.



A winning strategy is then an infinite subset of $graph(f)$, but any infinite subset of the graph of a modulus computes a function which dominates the modulus function, and therefore computes A.

An Objection

One might object to the spirit of the previous construction: we are only engaging with the rules of the game at a surface level, with infinitely many isolated, playable vertices littered throughout the graph, but the chips are never interacting in any way. We are never forced to move chips to a specific vertex in order to make it playable, we simply play vertices which are already playable from the start however we wish.

Strong Winnability

Definition

A chip-firing game $\langle G, c\rangle$ is strongly winnable if it is winnable and there are only finitely many vertices playable in the starting configuration c.

Strong winnability captures the essence of this objection to our construction, and the natural question is how Turing degrees of strong winning plays compare to Turing degrees of winning plays which are not strong.

Strongly Winnable Games

Theorem (Belanger, Dzhafarov, Ko, M., Solomon)

For any ω-c.e. set A, there is a strongly winnable, computable chip-firing game $\langle G, c\rangle$ such that the Turing degrees of winning plays for $\langle G, c\rangle$ are upwards-closed in the cone above A.

It is open whether the Turing degrees for winning strategies are exactly the cone above A. It is also open whether or not this can be proved for any Δ^0_2-set A to bring strong winnability in line with winnability. The proof fails strongly for sets which are not ω-c.e.

Further Work

- Given a chip-firing game $\langle G, c\rangle$, which configurations $\tilde{c} : V \to \omega$ can be realized by legal plays starting from c? Which can be realized computably?
- Given graphs which are not locally finite, define a variant of the chip-firing game where each vertex is assigned an ordinal number γ of chips. The choice of how to play a limit ordinal adds an extra dimension of strategy to the game that likely yields additional computational strength. (In particular, it is now possible to make a losing play.)

Thank you!

Klivans, C. J. (2018).
The mathematics of chip-firing.
Chapman and Hall/CRC.

Pegden, Wesley (2024).
Single-source sandpile.
[Online; accessed April 15th, 2024].