Chip-Firing Analysis of Stabilization Behaviors, Hitting Times, and Candy-Passing Games

Paul Kominers

1 Context and Inspiration

Math can be an intimidating field. To work on some problems, one must know decades or centuries of background before one can even understand the question. However, what tends to get lost in all of that is that math can be fun, even for the relatively uninitiated. There are problems in mathematics that are discrete (essentially, self-contained) and with some combination of background research, mathematical thought, and appropriate mentoring, they are easily within reach of the high school student.

During my sophomore and junior years in high school, I did work in discrete mathematics, outside the standard (albeit accelerated) math curriculum. I never had an official mentor; mostly I worked alone, but I was always able to enlist the aid of my teachers and some college professors whom I contacted, as well as similarly minded friends (both in high school with me and also some college students) who could review my work.

In the summer of 2007, I was privileged to have the opportunity to attend the Research Science Institute (RSI) at MIT. Every year at RSI, 80 rising high school seniors are brought in from all around the world to conduct research for six weeks in the Boston area. They are each paired with mentors in the area who work in the student's fields or related fields. The math students at RSI (myself among them) always work with graduate students from Harvard or MIT.

I was paired with Amanda Redlich, a combinatorics graduate student at MIT who had attended the University of Chicago as an undergraduate. Because of my prior work in combinatorial game theory, she gave me a few papers to read on rotor-routing and chip-firing, with the hopes that I would be able to construct and solve a game based on one or the other.

An early foray into chip-firing came to an interesting conclusion. I was counting the firings required for a maximized graph with $2k$ vertices and the source and the target vertices placed at opposite ends to stabilize. (I had begun this with the intention of making a game of it, but I soon realized that chip-firing was abelian (commutative) and therefore the game I was making was trivial.) However, each of the graphs took $(\frac{k}{2})^2$ firings to stabilize, which was intriguing.

I gradually collected more information, doing most of my work either in the math common room at MIT, my own room in the dorm that RSI was housed in, or wherever I happened to sit down and have a few minutes. Eventually the pattern generalized; given a maximized cycle of size n and target and source vertices at distance s from one another, it will take $ns - s^2$ firings to stabilize. With another day or two of work, I proved this.

After cycles, the obvious next place to go was trees. After that, I played with theta graphs, unfortunately to no avail. Amanda remembered a problem she had heard Dr. James Tanton talk about (the candy-passing game), and thought it resembled chip-firing. It took the remainder of the program, but I finally broke what was up until then the hardest result in my paper: showing that, in a candy-passing game with $c \geq 3n$, the configuration would eventually settle into a fixed state. (Later I extended this into $c \geq 3n - 2$.)

The weeks went by. I finished my paper and gave my talk, and RSI came to an end. I returned home, where one day I had a flash of inspiration. I had started contemplating another problem while I was at RSI-translating my very first result into a proof of the hitting times between distinct vertices on cycles-but there had been one essential part of the process that I could not understand.

Until it hit me, and, with the block broken, I knew how to solve the problem. My brother and his laptop (with Mathematica) were around, and I immediately spent more than a few hours grinding out, parameter by parameter, a three-part sum function that described the behavior of the system. Finally I had it right; I wrote it up, realized I had it wrong, fixed it, and wrote it up again.

With that, my paper was more or less in the final form that I used for all of the year's science fairs: not only the Intel Science Talent Search, but also the Siemens Competition in Math, Science, and Technology, the Junior Science and Humanities Symposium and our local science fair, as well as the Intel International Science and Engineering Fair. Edits followed, but they were mostly fairly minor. I submitted one portion of my research for publication in the Pi Mu Epsilon Journal; to my great joy, it was accepted [5]. (It is additionally available as arXiv 0709.2156.)

My Intel STS project, like all of my math work, was discrete; I read several papers' worth of background, but that was it, and not all of those eventually turned out to be useful. I could have done exactly as good a project having read only one or two. My work was inductive and intuitive and elegant and mostly self-contained. I believe that too many good brains are turned off to math because they never realize that mathematics can be elegant, interesting, or deep. Everyone, before he or she gives up on math, should crack open something outside of the high school norm. High school math can require a lot of rote learning–which is fine, insofar as it teaches the mechanics of and discipline for computation. But we cannot forget how much more–how much beauty, how much power, how much uniqueness–remains to be found in mathematics.

2 Introduction

2.1 Graph Theory

A *graph* G consists of a set $V(G)$ of vertices and a set $E(G)$ of edges connecting pairs of vertices. A vertex $v_1 \in V(G)$ which is connected by an edge to another vertex $v_2 \in V(G)$ is said to *neighbor* or *be adjacent to* v_2 [1]. The degree $(\deg v)$ of a vertex $v \in V(G)$ is the number of neighbors of vertex v.

A *path* from v_0 to v_n in G is an ordered set $\{v_0, \ldots, v_i, \ldots, v_n\}$ of adjacent vertices v_i in $V(G)$ such that $v_i \neq v_j$ for every $i \neq j$. The number of edges in the shortest path between two vertices is called the *distance* between those vertices. A graph is *connected* if, for any two vertices $v, v' \in V(G)$, there exists a path from v to v'.

We will define further graph-theoretic terminology as needed. A complete introduction to graph theory can be found in [1].

2.2 Random walks

The *random walk* is a commonly studied problem in which a bug is placed on a lattice or graph and allowed to wander at random. On a number line (or integer lattice of dimension 1), the probability that the bug will at some point return to its starting point is 1. On increasingly complex lattices and graphs, however, this probability decreases [2].

Generally, random walks on graphs are approximated by computing the expected *hitting time*, or probable number of random moves required to go from one vertex to another. Although random walks are useful in mathematics and computer science, probabilistic systems do not offer sufficient precision for some applications. There are, however, several emerging methods of deterministically simulating random walks which can be used to more efficiently compute hitting times [4, 6].

2.3 Chip-firing

One such deterministic simulation uses a process known as *chip-firing*. In chip-firing, we start with an arbitrary finite graph G. At each vertex $v \in V(G)$ we place an arbitrary number of chips g_v. If $g_v > (\deg v)$, then v *fires*, distributing evenly among its neighboring vertices the maximum number of chips that can be so distributed [6]. In other words, v distributes h chips to each neighboring vertex, where h is the greatest integer such that $h \cdot (\deg v) \leq g_v$.

Chip-firing settings often designate *target* vertices, which "absorb" chips and so never fire. Because they will never fire, we may treat such vertices as having zero chips, regardless of how many they have actually received from their neighbors. A chip-firing configuration that has no vertex capable of firing is called *stable*. A stable configuration such that one more chip placed at any non-target vertex will make that vertex fire is said to be *maximized*; in other words, a maximized graph is a graph such that each vertex $v \in V(G)$ has $(\deg v - 1)$ chips [6].

It is important to note that the chip-firing process is *commutative* (see [3, 6]). This means that a configuration's end-behavior does not vary with the order in which vertices fire. (By end-behavior, we mean whether the configuration will ever stabilize, and, if it stabilizes, the arrangement of chips among the vertices.) Also, vertices may be fired simultaneously.

Past research has focused on properties of chip-firing, as well as the ending configurations produced by certain starting configurations (see [3, 4, 6, 7]). In this paper we develop a formula to count the number of firings required to stabilize maximized cycles. We present applications of this idea to hitting time analysis. (In the full STS paper, which there is unfortunately insufficient room to present here, we additionally: develop a formula to count the number of firings required to stabilize arbitrary maximized trees after the addition of one chip; examine a special chip-firing game called the "candy-passing game," and show that when the number of chips is at least $3n - 2$, where n is the number of vertices, the configuration will eventually become fixed; and give a conjecture extending this result and discuss this result's application to the PageRank algorithm.)

3 Firings to Stabilize Cycles

In this section, we count the number of firings required to stabilize maximized cycles with a single target vertex after a single additional chip is placed at a given non-target vertex.

We examine n-cycles, connected graphs G having n vertices such that each $v \in V(G)$ has exactly two neighbors. An example of a maximized 6-cycle with one additional chip and a single target vertex is given in Figure 1:

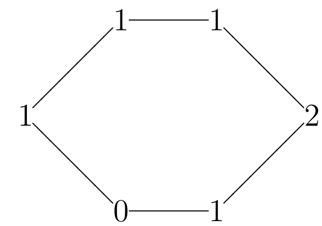

Figure 1: The number of chips on each vertex in a starting configuration of a 6-cycle with $s = 2$ in Theorem 3.1. The vertex with zero chips is the target vertex; the vertex with two chips is v_s.

Theorem 3.1. *Suppose we have a maximized n-cycle with one target vertex v_0. We place an additional chip at a vertex v_s at distance s from v_0. The number of firings required to stabilize the configuration is*

$$ns - s^2. \tag{1}$$

Proof. We number the vertices consecutively, $v_0, v_1, v_2, \ldots, v_s, \ldots, v_k, \ldots, v_{n-1}$. (For an example of such a labeling, see Figure 2.) We color the additional chip placed at v_s blue.

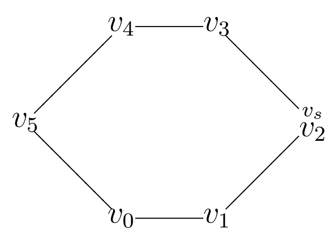

Figure 2: A sample labeling of a 6-cycle.

Because chip-firing is commutative, the number of firings required for stabilization using a specific ordering of firings is the same as for any other ordering. Therefore, we may order the firings without changing the number of firings required for stabilization. We assign the ordering:

- The vertex containing the blue chip is fired if and only if it is the only vertex on the graph that may be fired.
- When the vertex v_k containing the blue chip fires, the blue chip moves to $v_{(k+1) \bmod n}$.

We call the period of time beginning with one firing of the vertex containing the blue chip and ending just before the next firing of the vertex containing the blue chip a *round* of firing.

Claim 3.2. *At the end of any round of firing t, there is exactly one empty vertex, located at v_t.*

Proof. We show the claim by induction, starting at round $t = 0$. When the blue chip is first placed, its vertex v_s is the only vertex capable of firing. Once it fires, v_s, which began with two chips and fires two chips, is empty, and the adjacent vertices are both capable of firing. Now there is exactly one vertex not containing the blue chip which is capable of firing, v_{s-1}; it fires. Now v_{s-2} can and does fire, adding a chip to v_{s-1} and v_{s-3} and becoming empty itself. If we continue these firings, eventually v_2 will be empty and v_1 will fire, leaving v_1 empty and only v_{s+1} capable of firing, meaning that we have started a new round of firing. Now the blue chip is at $v_{s+1} = v_{s+t}$ and v_1 is empty.

It follows from an analogous argument that at the beginning of any round t the vertex containing the blue chip is v_{s+t} and the empty vertex is at v_t. □

The vertex containing the blue chip fires once in each round of firings. The blue chip moves $n - s$ times, or once for every vertex it passes over on its way to the target vertex. In each round of firing, every vertex between the vertex containing the blue chip and the empty vertex fires once. Since the empty vertex follows the blue chip at distance s, there are s firings in each round, which means that our total number of firings is $s(n-s) = ns - s^2$. $\square$

The following corollary is implied by the proof of Theorem 3.1.

Corollary 3.3. *Once the cycle in Theorem 3.1 has become stable, each non-target vertex should contain a single chip, with the exception of v_{n-s}, which will be empty.*

4 Hitting Times on Cycles

The *elementary excitation-relaxation operation* is defined as follows: given a graph G with one target vertex v_0, place a chip at a *source vertex* v_s and fire until the graph stabilizes. If we perform the elementary excitation-relaxation operation m times on the same vertex v_s of a maximized cycle or tree, the graph will eventually return to its maximized state. Suppose that it takes m elementary excitation-relaxation operations for the graph to return to its maximized state and that M chip-motions (note that we count motions, not firings) occur over those m elementary excitation-relaxation operations. Then the ratio $\frac{M}{m}$ is equal to the *hitting time* for a random walk from v_s to v_0 [6]. (The definition of a hitting time is given in the introduction.)

We denote by $i = 0, 1, \ldots$ the index of a given iteration of the elementary excitation-relaxation operation. A vertex other than v_0 that is empty between iterations of the elementary excitation-relaxation operation is called a *break vertex*, denoted by v_b.

Theorem 4.1. *Given an n-cycle, the hitting time from two distinct vertices at distance s from one another is given by*

$$ns - s^2. \tag{2}$$

Proof. We number the vertices as in Theorem 3.1, declaring a target vertex at v_0 and a source vertex at v_s. We first show that the break vertex moves s vertices on every iteration of the elementary excitation-relaxation operation. This will allow us to count the number of iterations of the elementary excitation relaxation operation and, by extension, the number of firings in each iteration.

Claim 4.2. *After iteration i of the elementary excitation-relaxation operation, $v_{(n-si) \bmod n}$ is empty.*

Proof. We first show the claim when $|b - s| < s$ at the beginning of iteration i of the excitation-relaxation operation. Since no chips will move past either v_0 or v_b in the course of the iteration, we may temporarily identify v_b with v_0 by collapsing vertices $v_b, \ldots, v_1$ to v_0; thus, we may consider the graph to be a cycle of size $n - b$ with a single target vertex for the course of this iteration. If $|b - s| < s$ and $b < s$, then the distance from v_s to the new v_0 during iteration i is $s - b$. At the beginning of iteration $i + 1$, v_b will be at the vertex $v_{n-(s-b)} = v_{n-s+b}$. In this case, v_b has moved $s - b$ vertices from v_0 and skipped over the b vertices between v_b and v_0, moving a total of $b + s - b = s$ vertices.

We now show the claim when $|b - s| < s$ and $b > s$. For the same reasons as above, we may once again temporarily identify v_b with v_0 by collapsing vertices $v_b, \ldots, v_{n-1}$ to v_0. (Also note that we collapse a different set of vertices. This is simply to avoid collapsing v_s to v_0.) This gives us a cycle of size b with one target vertex v_0 and on which the distance from the new target vertex to v_s will be $b - s$. At the beginning of iteration $i + 1$, then, the break vertex has moved $b - s$ vertices from v_0 to v_{b-s}. We thus find the total distance it has moved to be $b - (b - s) = s$.

We now show the claim when $|b - s| \geq s$. Consider an iteration of the elementary excitation-relaxation operation on a cycle of size $n' \geq 2s$. After this iteration, by Corollary 3.3, $v_{n'-s}$ has zero chips. As above, we may temporarily identify $v_{n'-s}$ with v_0 by collapsing vertices $v_{n'-s}, \ldots, v_{n-1}$ to v_0, giving us a cycle with $n' - s$ vertices. We may then treat the next iteration of the elementary excitation-relaxation operation as the first iteration of the elementary excitation-relaxation operation on an $(n-s)$-cycle, and our induction is shown. $\square$

If v_b moves s vertices on every iteration, where s does not divide n, it will move $\frac{n}{\gcd(n,s)}$ times before it reaches its starting point. Thus, there are $\frac{n}{\gcd(n,s)}$ iterations. At the beginning of the last iteration, the break vertex is at v_s, so there are $\frac{n}{\gcd(n,s)} - 1$ iterations in which the break vertex is some vertex other than v_s.

There are $\frac{(n-2s)}{\gcd(n,s)} + 1$ "outside" iterations $o = 0, 1, \ldots, \frac{(n-2s)}{\gcd(n,s)}$ where $|b-s| \geq s$. (The one additional iteration comes from the iteration in which $b = 0$.) At the beginning of each of these iterations, the distance from v_0 to the source v_s is s and the size of the cycle is $n-b$, where $b = o \cdot \gcd(n,s)$.

There are $\frac{s}{\gcd(n,s)}$ "lesser" iterations $l = 1, 2, \ldots, \frac{(s)}{\gcd(n,s)} - 1$ where $|b-s| < s$ and $b < s$. At the beginning of each of these iterations, the distance from v_0 to the source v_s is $s-b$ and the size of the cycle is $n-b$, where $b = l \cdot \gcd(n,s)$.

There are $\frac{s}{\gcd(n,s)}$ "greater" iterations $g = 1, 2, \ldots, \frac{(s)}{\gcd(n,s)} - 1$ where $|b-s| < s$ and $b > s$. At the beginning of each of these iterations, the distance from v_0 to the source v_s is $b-s$ and the size of the cycle is b, where $b = g \cdot \gcd(n,s)$.

Therefore, the hitting time is given by

$$\frac{2}{\frac{n}{\gcd(n,s)}}\left(\sum_{o=0}^{\frac{n-2s}{\gcd(n,s)}}(s(n-o\gcd(n,s))-s^2)\right.$$
$$+\sum_{l=1}^{\frac{s}{\gcd(n,s)}-1}((s-l\gcd(n,s))(n-l\gcd(n,s))-(s-l\gcd(n,s))^2)$$
$$\left.+\sum_{g=1}^{\frac{s}{\gcd(n,s)}-1}((g\gcd(n,s))(g\gcd(n,s)+s)-(g\gcd(n,s))^2)\right)=$$
$$\frac{2\gcd(n,s)}{n}\left(-s^2(-1+\frac{s}{\gcd(n,s)})+s^2(\frac{s-\gcd(n,s)}{\gcd(n,s)})-\frac{s(2s-n)(2s-\gcd(n,s)-n)}{2\gcd(n,s)}\right.$$
$$\left.+sn(-1+\frac{s}{\gcd(n,s)})-\frac{sn(s-\gcd(n,s))}{2\gcd(n,s)}-s^2(1+\frac{-2s+n}{\gcd(n,s)}+sn(1+\frac{-2s+n}{\gcd(n,s)})\right)$$
$$=ns-s^2.$$

□

5 Conclusion and Future Directions

In this paper, we developed formulae for the number of firings required to stabilize cycles and for the hitting times between distinct vertices on cycles. In the future, we would also like to derive formulae to describe the firings required for general undirected or directed graphs to stabilize. We would also like to do further work on the candy-passing game (the current work on which is not presented in this paper).

References

[1] B. Bollobás, *Modern Graph Theory*, Springer, New York, 1998.

[2] W. Feller, *An Introduction to Probability Theory*, **1**, John Wiley, New York, 1957.

[3] J. Jeffs and S. Seager, The chip-firing game on n-cycles, *Graphs and Combinatorics* **11**, 1995, pp. 59-67.

[4] M. Kleber, Goldbug variations, *Mathematical Intelligencer*, **27 (1)**, 2005, pp. 1-20.

[5] P. Kominers, The Candy-Passing Game for $c \geq 3n-2$, *Pi Mu Epsilon Journal*, **12 (8)**, 2008m pp. 459-460.

[6] J. Propp, A whirling tour of chip-firing and rotor-routing, DIMACS Workshop on Puzzling Mathematics and Mathematical Puzzles, `http://jamespropp.org/tour.pdf`, 2007.

[7] J. Tanton, Today's puzzler, *The St. Marks Institute of Mathematics Newsletter*, November 2006, pp. 1-2.