2020

A Graph Theoretical Approach to the Dollar Game Problem

Rachel Koch
Virginia Commonwealth University

A Graph Theoretical Approach to the Dollar Game Problem

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

Rachel Koch
Master of Science

Director: Richard Hammack, Professor
Department of Mathematics and Applied Mathematics

Virginia Commonwealth University
Richmond, Virginia
April 2020

Acknowledgements

First and foremost, I would like to thank my family for supporting me on this now eight year journey. Most especially, I would like to thank my children, Adrianna and Bennett Estremera, for helping to pick up the slack at home when mom decided to go back to school.

I would also like to thank my various advisors throughout my academic career. Thank you to Justin Storer and Wendi Dass for having faith in me from the begininning. Thank you to Cassandra Williams for giving me some of the best advice I've ever gotten and helping me stay sane. Thank you to Edwin O'Shea for being my biggest and best advocate. Your support means the world to me. Thank you to Neal Bushaw for giving me my first taste of graph theory and making it so fantastic.

Finally, thank you to Richard Hammack for being a patient, kind, understanding, and encouraging advisor, even when I wasn't always very kind to myself.

Table of Contents

List of Figures

Abstract

In this thesis we consider a problem in Graph Theory known as the Dollar Game. The Dollar game was first introduced by Matthew Baker of the Georgia Institute of Technology in 2010. It is a game of solitaire, played on a graph, and is a variation of chip firing, or sand-piling games. Baker approached the problem within the context of Algebraic Geometry. It is the goal of this paper to provide an overview of the necessary graph theory to understand the problem presented in this game, as well as background on chip firing games, their history and evolution. Finally we will present a variety of results about the Dollar Game from a graph theoretical standpoint.

Chapter 1

Introduction

1.1 Introduction

This thesis explores a particular problem in graph theory, known as the Dollar Game, a game of solitaire played on a graph. It is a variation of chip-firing games, which fall under the branch of graph theory known as algebraic graph theory. Study of chip-firing games expanded rapidly throughout the eighties and nineties, relying heavily on group theory. J. Spencer first began looking into chip-firing games on paths, but the games were quickly expanded to general finite graphs by his successors.

The current form of the Dollar Game was first introduced by Matthew H. Baker of Georgia Institute of Technology [3]. The Dollar Game is a game of solitaire, played on a graph on which each vertex holds an integral number of dollars, not necessarily positive or non-negative. The goal of the game is to get every vertex out of debt through a series of borrowing and lending moves between neighboring vertices. Baker determined necessary conditions for winnable games, and he also determined sufficient conditions for winnable games. He presented proofs of these conditions in the field of Algebraic

Geometry.

Our aim is to expand on Baker's work by providing graph theoretical proofs to the results he found, as well as some of our own results. We also present a variety of algorithms we believe solve this game given certain sufficient conditions. We present our progress in this thesis, as well as our plans for continued work on this problem.

First, we will present a basic overview of graph theory, and the background knowledge necessary to understand the Dollar Game. Next we will provide a history and evolution of the chip firing games that lead to the Dollar Game, and present some interesting results on that topic. Finally we will close with an analysis of the dollar game and present several results we were able to obtain. We will also present our progress towards graph theoretical proofs of the sufficient conditions, and discuss how we would like to continue in the future.

1.2 Background Literature Review: Graph Theory [5]

We begin with a brief overview of the topic of graph theory, focusing on the relevant definitions needed in order to formally introduce chip firing games and the Dollar game in particular.

A **graph** is an ordered pair $G = (V, E)$ consisting of a non-empty set of vertices V and a set of edges E. The set E consists of non-ordered pairs of (not necessarily distinct) vertices. The vertices in the ordered pair are the **endpoints** of the edge. An edge with the same vertex as its endpoints is a **loop**.

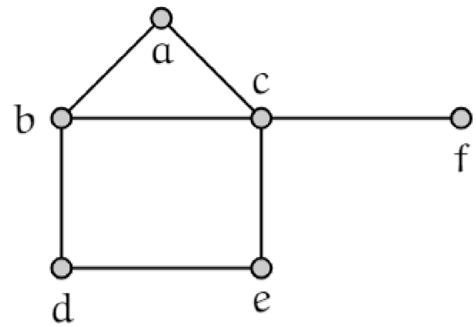

Figure 1.1: A graph G with vertex set $V(G) = \{a, b, c, d, e, f\}$ and edge set $E(G) = \{(a, b), (a, c), (b, c), (b, d), (c, e), (c, f), (d, e)\}$.

Two vertices $u, v \in V(G)$ are said to be **adjacent** if they are the endpoints of an edge in the graph. If two vertices are adjacent, then they are said to be **neighbors**. An edge is said to be **incident** to its endpoints and vice versa. The **degree** of a vertex, $d(v)$, is equal to the number of edges incident to the vertex, with loops counting twice.

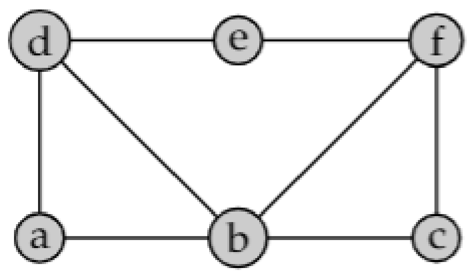

Figure 1.2: The degree of vertex d is 3, while the degree of vertex e is 2.

A **walk** is a list of vertices $v_1 v_2 ... v_k$ such that v_i is adjacent to v_{i+1} for each $1 \leqslant i < k$. A **closed walk** is a walk such that $v_1 = v_k$. A **path** is a walk such that $v_1, v_2, ..., v_k$ are all distinct vertices. A $(u, v)-$**path** is a path the begins at vertex u and ends at vertex v. A **cycle** is a closed walk whose origin and internal vertices are distinct.

A graph is said to be connected if for every pair of vertices $u, v \in V(G)$ there exists a $(u, v)-$path. If a graph is not connected, then its maximal connected subgraphs are the **components** of the graph.

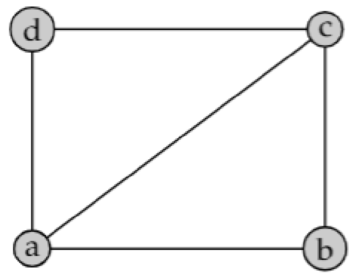

Figure 1.3: The graph G has three cycles: abc, acd and $abcd$.

A **tree** is a connected graph G that doesn't contain any cycles. A tree has $n - 1$ edges, where n is the number of vertices in the graph. The **spanning tree** of a graph G is a subgraph T that has the same vertex set as G, is connected, and is a tree.

Chapter 2

Literature Review: Chip Firing Games

2.1 Origin of Chip Firing Games

A precursor to the chip firing games was studied in a 1986 paper by J. Spencer, who investigated a process of balancing vectors [9]. A non-negative integer N is placed in the center coordinate of a row vector of length N $\{a_1, a_2, ..., a_N\}$, and the other coordinates are padded with zeros. A turn consists of taking each coordinate in the vector with a number $n > 1$, subtracting $2\lfloor \frac{n}{2} \rfloor$, then adding $\lfloor \frac{n}{2} \rfloor$ to the right and to the left [1].

Example 1. Beginning with an initial value of five.

```
Initial Configuration: <0,0,5,0,0>
Step One:              <0,2,1,2,0>
Step Two:              <1,0,3,0,1>
Step Three:            <1,1,1,1,1>
```

Example 2. Beginning with an initial value of six.

```
Initial Configuration: <0,0,0,6,0,0,0>
Step One:              <0,0,3,0,3,0,0>
Step Two:              <0,1,1,2,1,1,0>
Step Three:            <0,1,2,0,2,1,0>
Step Four:             <0,2,0,2,0,2,0>
Step Five:             <1,0,2,0,2,0,1>
Step Six:              <1,1,0,2,0,1,1>
Step Seven:            <1,1,1,0,1,1,1>
```

In general, if N is odd, the process concludes with a string of N consecutive 1's in the center of the vector, and if N is even, the process concludes with a zero in the center, and a string of $\frac{N}{2}$ consecutive 1's to the left and $\frac{N}{2}$ consecutive 1's to the right of the center [4].

Spencer and several others expanded upon this paper in 1989 by converting this process to a game of subdividing and moving piles of discs. The game begins with several piles of discs, all arranged in a line. A legal turn of the game is to choose any pile with more than one disc, and move one disc to the left and one disc to the right [1].

They focused the study of this game on how long it would take the game to terminate (that is, when no pile has more than one disc), and what the final configurations would look like. Although the game is defined for any line of piles of disks, Spencer et al focused on the version that mirrored his initial work with vectors and began each game with the disks all stacked in the center pile. They discovered that the final configuration of this game always mirrored the final configuration of the original version on vectors.

Spencer and his colleagues further found that given any initial configuration of the game, with the discs initially spread over several piles lined in a row, the final configuration would always be the same, regardless of the order in which the turns were taken [1].

We include two examples here, but for clarity we will illustrate the piles of discs using vectors as in the original version of the game.

Example 3. Consider this new configuration of five discs where we will always take the first available move on the left-hand side.

```
Initial Configuration: <0,1,2,2,0>
Step One:              <0,2,0,3,0>
Step Two:              <1,0,1,3,0>
Step Three:            <1,0,2,1,1>
Step Four:             <1,1,0,2,1>
Step Five:             <1,1,1,0,2>
Step Six:             <1,1,1,1,0,1>
```

Now we will play this same configuration, but we will always take the first available move on the right-hand side.

```
Initial Configuration: <0,1,2,2,0>
Step One:              <0,1,3,0,1>
Step Two:              <0,2,1,1,1>
Step Three:            <1,0,2,1,1>
Step Four:             <1,1,0,2,1>
Step Five:             <1,1,1,0,2>
Step Six:              <1,1,1,1,0,1>
```

Notice that we end up with the same final configuration despite taking our moves in a different order.

2.2 Chip Firing Games

The disk stacking game can be exactly represented by placing N chips on the center vertex of a graph that is a path of length $N + 1$, or for the more generalized version, assigning a non-negative integer n of chips to each vertex on a path of sufficient length. As in the previous game, we choose any vertex whose number of chips exceeds its degree per turn (in this case, 2 for internal vertices), and sends one chip along each of its incident edges to its neighbors.

Now, consider an arbitrary, finite, connected, simple graph and label each vertex with a non-negative integer for a total of N chips all together. A legal turn is taken by selecting a vertex v with chip number greater than or equal to its degree, and moving one chip along each of the vertex's incident edges to its neighbors. The number of chips at vertex v decreases by its degree, and each of its neighbors increase their chips by one. The game terminates when there are no more legal moves on the board. The question is whether the game will ever terminate, or if the game will go on infinitely.

Definition 1. Let G be a finite, connected, simple graph. Assign a non-negative integer $C(v)$ to each vertex. We say that a vertex has $C(v)$ chips and refer to this assignment this assignment on all vertices as a **chip configuration** [8].

Definition 2. We say a vertex v is **ready to fire** if the number of chips located at v is greater than or equal to its degree.

Definition 3. Let G be a graph with configuration $C(v)$. Then the configuration $C(v)$ is said to be **stable** if there are no vertices in G that are ready to fire.

Example 4. In the following example, we begin with 6 chips on the graph. We have f fire first, followed by a and d. Then c is ready to fire and does so. Following this move, the game terminates as the graph configuration is now stable, Notice that the total number of chips on the graph never changes.

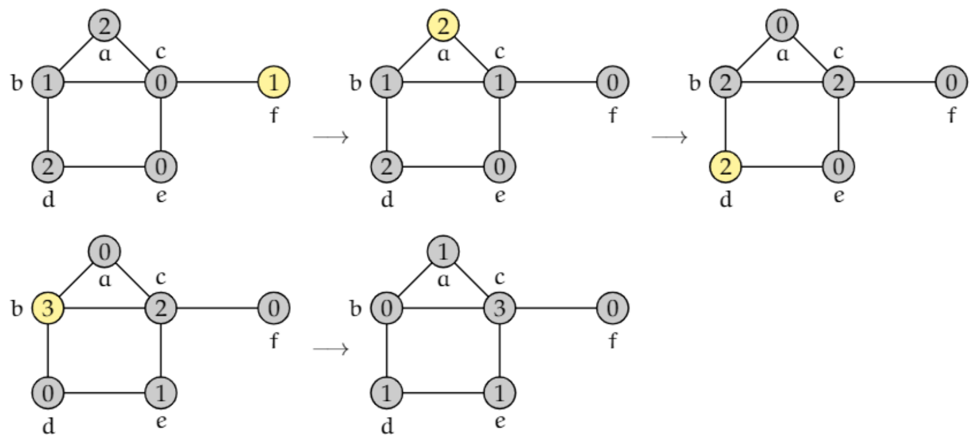

Figure 2.1: Example of a chip firing game.

Lemma 4. *[4, Lemma 3.2] If a game terminates, then there is at least one vertex that never fires.*

Proof. Let G be a finite, simple, connected graph with configuration $C(G)$. Suppose the configuration stabilizes, but every vertex fires. Then let v be the vertex which has gone the longest since firing at the time the configuration stabilizes. Then every neighbor of v has fired at least once after v fired, but before the game stabilized. Since v received at least one chip from each of its neighbors, the number of chips on v must be at least its degree. Thus v is ready-to-fire, which contradicts the game stabilizing [4]. $\square$

Notice that in example 3, neither vertex c nor e fired during the game.

Lemma 5. *[4, Lemma 3.1] If a game goes on infinitely, then every vertex fires an infinite number of times.*

Proof. Let G be a finite, simple, connected graph with configuration $C(G)$. Suppose the configuration never stabilizes. Then there exists some vertex v such that v fires an infinite number of time. Let u be a neighbor of v. Then u receives an infinite number of chips from v. But the number of chips on the graph is finite. Thus, u must fire an infinite number of times as well. Since the graph is connected, this argument can be made for every vertex [4]. □

Theorem 6. *[4, Theorem 3.3] Let E be the number of edges in the graph. If $N < E$, then $C(G)$ will stabilize.*

Proof. Let G be a finite, simple, connected graph with configuration $C(G)$. Let N be the number of chips on the graph and let E be the number if edges in the graph. Suppose $N < E$. Now, color the edges each a different color with colors 1,2,...,E. Each time a chip is fired down an edge, color the chip with the edge's color. That edge is now paired with that chip, and the chip can only be fired along that same edge. So every chip either gets colored, or remains stationary on its vertex if that vertex never fires. Since $N < E$, there exists some edge that never gets paired with a chip. That edge is incident with a vertex, and as such, that vertex never fires. Thus, by 5 we know that the configuration must stabilize. □

Theorem 7. *[4, Theorem 3.3] Let E be the number of edges in the graph, and let V be the number of vertices. If $N > 2E - V$, then $C(G)$ will never stabilize.*

Proof. Let G be a finite, simple, connected graph with configuration $C(G)$. Let N be the total number of chips on the board, E the number of edges in the graph and V be the number of vertices in the graph, such that $N > 2E - V$. Suppose $C(G)$ stabilizes. Then for any vertex v in the graph $C(v) \leqslant d(v) - 1$ Thus, $N \leqslant \sum_{v \in V(G)} (d(v) - 1) = 2E - V$. But this contradicts $N > 2E - 1$. Thus, $C(G)$ must always have some vertex v such that $C(v) > d(v)$, and the configuration never stabilizes [7]. □

Theorem 8. *[4, Theorem 3.3] If* $N \geqslant E$*, there is always an initial configuration that leads to a game that never stabilizes.*

Proof. Let G be a finite, simple, connected graph. Assign a direction to each edge so that the graph is acyclical. Now, place chips on each vertex equal to the vertices out-degree, $d^-(v)$. Since every edge contributes one out-degree to one of its endpoints, and $N \geqslant E$, we know this can be done. Since G as a directed graph is acyclic, there exists some vertex v such that $d^-(v) = d(v)$. This vertex is ready-to-fire in G. Fire one chip along each of its incident edges. Now v has no chips, and every neighbor of v has one more chip than previously.

Reverse the orientation of every edge incident to v. This is another acyclic orientation, and as such, there exists a vertex $u \neq v$ such that $d^-(u) \geqslant d(u)$. This vertex is ready-to-fire in G and repeating this process of reversing the orientation of the edges incident to the vertex that just fired immediately after firing ensures there is always a vertex ready-to-fire, and thus the configuration never stabilizes. [7]. □

Example 5. In this example we give an acyclic orientation to our previous chip firing example graph, and then place $d^-(v)$ chips on each vertex. The total number of chips is $N = E$. The vertex b is ready-to-fire, so it sends a chip to each of its neighbors. We then reverse the orientation of the edges incident to b. Now a is ready-to-fire. It fires and we reverse it's incident edges. The vertex c is now ready-to-fire. After reversing the edges incident to c we have two vertices ready to fire, d and f. It doesn't matter which we fire first, so we fire f. The vertex d is not incident to f, so reversing f's incident edges doesn't impact d. Now d fires and we reverse it's incident edges. Again we have two vertices that are ready-to-fire, and again it doesn't matter which we choose to go first. But notice that by the time we get back to b having all outgoing edges, all of its neighbors must have fired in the interim. Thus, it must have enough chips to be ready-to-fire.

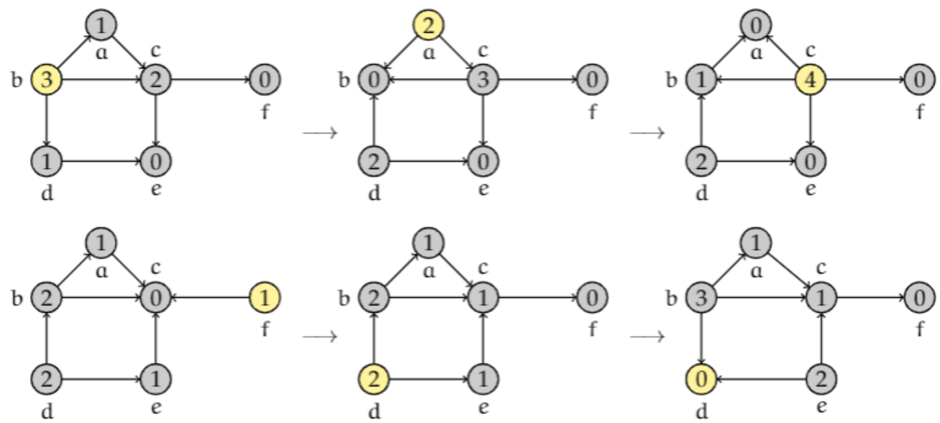

Figure 2.2: Example of the first five moves in an infinite configuration with $N = E$.

Theorem 9. *[4, Theorem 3.3] If* $N \leqslant 2E - V$*, the there is always some initial configuration that leads to a stable game.*

Proof. Let G be a finite, simple, connected graph. Suppose $N = 2E - V$. Then for each $v \in V(G)$ let $C(v) = d(v) - 1$. (Recall that $\sum_{v \in V(G)} (d(v) - 1) = 2E - V$. If $N < 2E - V$, simply remove $(2E - V) - N$ chips from the vertices of your choosing. Since $C(v) < d(v)$ for all v, the configuration is stable [7]. $\square$

Interestingly, like the vector/disk balancing games, given an initial configuration, either every chip firing game is infinite, or it terminates in the exact same configuration, regardless of the order the turns are taken in [4].

2.3 The Bank Game

A variation of the chip firing games is known as the Bank variant. It works almost the same as the regular chip firing games, except that it introduces the concept of bank.

Definition 10. If a vertex v in G is designated as a **bank**, then v cannot fire unless no other non-bank vertex in G is ready-to-fire [8].

This version established the beginning of thinking of the chips as a form of currency. The bank vertex stores chips until the "economy" becomes stagnant, and then revitalizes the economy in this circumstance.

Definition 11. A chip configuration on a graph G is said to be **bank stable** if the only vertex ready-to-fire is a bank vertex.

Theorem 12. *A bank game configuration on a graph* G *will always become (bank) stable.*

Proof. Let G be a finite, simple, connected graph with configuration $C(G)$. Let $v \in V(G)$ be a designated bank and suppose the game never stabilizes. Then v never fires. But this contradicts 5. Therefore the game must become (bank) stable. $\square$

In 1999, N.L. Biggs of the London School of Economics introduced a further bank variant he called the Dollar Game. In Biggs' variant, the bank vertex is allowed to go into debt, or have a negative number of chips. Not only is the bank vertex allowed to go into debt, the bank must fire if at any point it has a positive number of chips. The game stabilizes if no non-bank vertex is ready-to-fire and the bank has a non-positive balance [2]

Chapter 3

The Dollar Game

In 2007, Matthew Baker introduced another variant of the dollar game, and it is this version of the game that will concern us for the remainder of this thesis. In Baker's dollar game, the chips are still replaced with dollars and the game represents an economy of sorts. Instead of there being bank vertices that can go into debt, any vertex, in this case representing households or individuals, can go into debt. At the beginning of the game, vertices are assigned any integer amount, either negative, zero or positive. Negative integers represent a vertex being in debt, and a positive integer represents a vertex having a surplus of funds. The goal of the game is to get every vertex out of debt through a combination of lending and borrowing moves.

Definition 13. A vertex may make a **lending move** by giving each of its neighbors one dollar. The lending vertex does not necessarily have to have the dollars to accommodate this and may go into debt to complete this move.

Note that this is equivalent to the chip firing move in the chip firing games with the exception that now vertices can go into debt. However, there is an additional inverse move for this game.

Definition 14. A vertex may make a **borrowing move** by borrowing a dollar from each of its neighbors. The neighboring vertices do not necessarily have to have the funds to cover this, and may go into debt as a result of this move.

It's important to note that while a vertex may go as far into debt (or in the positive) as necessary, like the chip firing games, the overall total amount of money on the board never changes. Additionally, for a game to winnable, each of its components must be winnable, so we will only be considering finite, connected graphs.

Definition 15. Let G be a graph. Assign an integer $D(v)$ to each vertex. We say that a vertex has $D(v)$ dollars and refer to this assignment on all vertices as a **dollar configuration** of G denoted $D(G)$.

Theorem 16. *If a game on graph G is winnable, then the total number of dollars on the board must be non-negative.*

Proof. Let G be a graph with dollar configuration $D(G)$. Suppose G is winnable. Then at the end of the game we have $D(v) \geqslant 0$ for all $v \in V(G)$. Thus $\sum_{v \in V(G)} D(v) \geqslant 0$ at the end of the game. But the total number of dollars on the graph never changes, thus $\sum_{v \in V(G)} D(v)$ must always be non-negative. $\square$

Lemma 17. *The results of a lending move can be achieved exactly through a series of borrowing moves.*

Proof. Let G be a graph with dollar configuration $D(G)$. Let x be a vertex in H. Suppose we want x to lend a dollar to each of it's n neighbors. Then we want each of x's neighbors to gain a dollar, and x to lose n dollars. Then for each $v \in H \setminus \{x\}$, borrow one dollar, and consider the outcome for a vertex v.

Case 1: v is a vertex that is not adjacent to x. Then v borrows a dollar from each of its k

neighbors, and gains k dollars. But v also has k neighbors borrow from it, so v loses k dollars. This is a total net gain of 0 dollars.

Case 2: v is adjacent to x. Then v borrows a dollar from each of its j neighbors to gain j dollars. However, v also has $j-1$ neighbors borrow a dollar from it, so v loses $j-1$ dollars. This is a net gain of 1 dollar.

Case 3: $v = x$. Then v borrows no dollars from it's neighbors and gains 0 dollars. Additionally, v has n neighbors borrow one dollar each, so v loses n dollars. This is a net loss of n dollars.

This is equivalent to x lending each neighbor a dollar. Therefore, any lending move can be achieved through a sequence of borrowing moves. $\square$

Corollary 18. *If a game on a graph* G *with initial configuration* $D(G)$ *is winnable, then it can be won through only making borrowing moves.*

Lemma 19. *Borrowing moves are commutative, so it does not matter which order they are taken in.*

Proof. Let G be a graph with initial dollar configuration $D(G)$. Let x, y be vertices of G. Let the degree of x, $d(x) = n$ and $d(y) = m$. Suppose x borrows a dollar. Then all of x's neighbors decrease by one dollar and x increases by n dollars. Then y borrows a dollar. Neighbors of y decrease by one dollar and y increases by m dollars. In total, shared neighbors of x and y decrease by two dollars, and neighbors in the symmetric difference decrease by one dollar. If x and y increase by n and m dollars respectively, unless they are neighbors. In that case x increases by $n-1$ dollars and y increases by $m-1$ dollars.

Now suppose y borrows a dollars first Then all of y's neighbors decrease by a dollar and y increases by m dollars. Then x borrows a dollar, so all of x's neighbors decrease by a dollar and x increases by n dollars. In total, shared neighbors of x and y decrease by two dollars, and neighbors in the symmetric difference decrease by one dollar. If x and y increase by n and m dollars respectively, unless they are neighbors. In that case x increases by $n-1$ dollars and y increases by $m-1$ dollars. This is the same outcome as when x borrowed first. Therefore, borrowing moves are commutative. $\square$

Since lending moves can be converted to borrowing moves, and borrowing moves are commutative, we can assume all moves happen simultaneously.

Definition 20. Let T be a connected acyclic graph (tree). Let P be a path in T. Let x_i be a vertex on P. Then define the **x_i-subtree** to be the subgraph of T consisting of x_i and the branches of of x_i that do not lie along P.

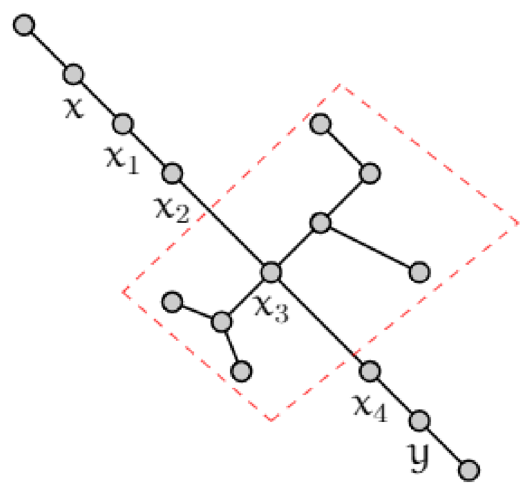

Figure 3.1: Example of an x_i-subtree. The x_3-subtree is outlined in the red box.

Lemma 21. *For a game on a tree graph T, a dollar can be moved from any one vertex to any other vertex while maintaining dollar amounts on all other vertices in the tree.*

Proof. Let T be a tree with dollar configuration $D(G)$. Suppose there is one dollar on vertex x we wish to move to vertex y. Since T is a connected tree, there exists a unique (x, y)-path. Let $P = xx_1...x_{n-1}y$ be such a path. Then the distance from x to x_i along path P is i. For each x_i along P and its corresponding x_i-subtree, borrow i dollars. For y which is distance n from x, and its corresponding y-subtree, borrow n dollars.

Consider vertex v after these moves have taken place.

Case 1: $v = x$. Then x decreases by one dollar.

Case 2: $v = x_i$ for some x_i along P. Then x_i borrows $[d(x_i)]i$ dollars. But it has $[d(x_i) - 2]i + (i-1) + (i-2) = [d(x_i)]i$ dollars borrowed from it, for a net change of zero dollars.

Case 3: $v \in$ the x_i-subtree. Then v borrows $[d(v)]i$ dollars, but also has $[d(v)]i$ dollars borrowed from it, for a net change of zero dollars.

Case 4: $v = y$. Then y borrows $[d(y)]n$ dollars. But it also has $(n-1) + [d(y) - 1]n = [d(y)]n - 1$ dollars borrowed from it, for a net change of one dollar.

Therefore we have moved one dollar from x to y without changing the dollar amounts on any other vertices. □

Corollary 22. *If a game on a tree T has configuration $D(T)$ such that the total number of dollars on the graph is non-negative, then the game is winnable,*

Proof. This follows immediately from the fact that we can move any positive dollar amount to any other vertex without impacting the dollar amounts on the other vertices. □

Example 6. In this game, we wish to move the \$1 that is currently on the vertex x to the vertex y so that y can get out of the negative. We need to leave all other dollar amounts on vertices unchanged, or one of them may go into the negative. We begin by having vertex x_1 borrow \$1 since it is distance one from x. Then vertex x_2 and all vertices on the x_2-subtree borrow \$2. This leaves all of the vertices on the subtree, excluding x_2, with unchanged dollar amounts, since they each have an even exchange of \$2 with their neighbors. Then x_3 borrows \$3, and finally, y and the y-subtree borrows \$4. Again, the additional vertices on the y-subtree remain with unchanged dollar amounts since it's an even exchange of currency between neighbors.

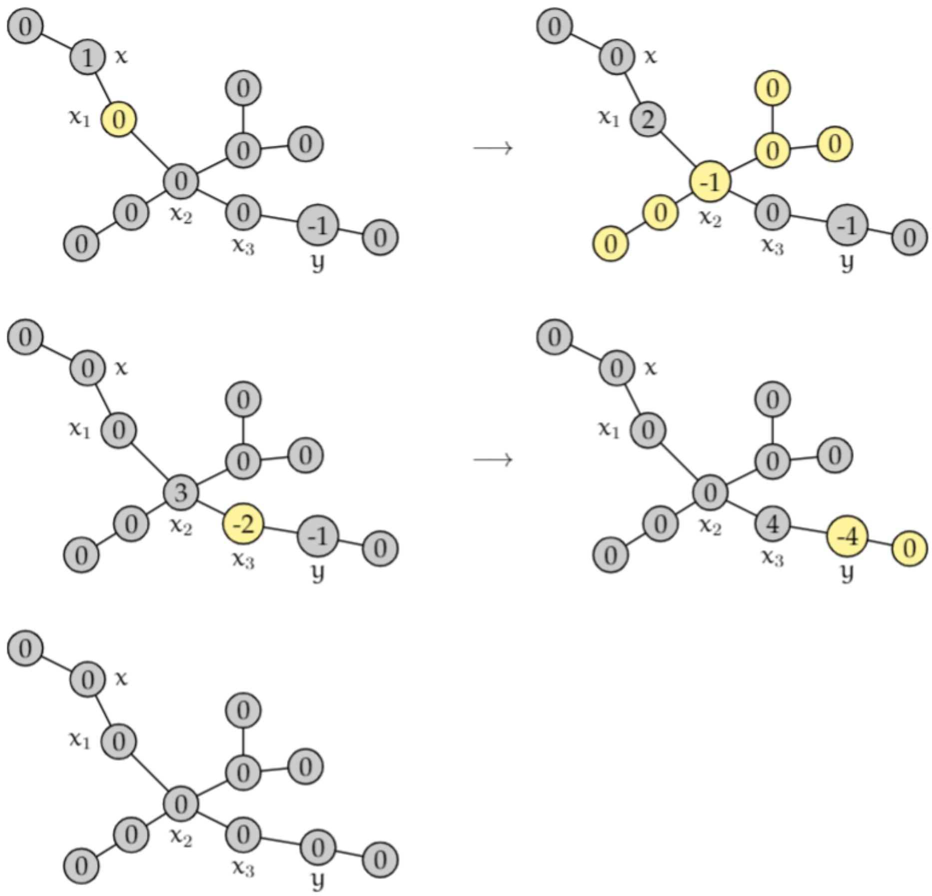

Figure 3.2: Example of moving one dollar from one vertex to another on the tree.

We have seen that having a nonnegative total dollar amount is necessary for a winnable game, and we have shown that this condition is sufficient for a winnable game on a tree. However, this condition is not sufficient in general for arbitrary graphs that contain cycles.

3.1 The Dollar Game with Cycles

A non-negative total of dollars on the graph is a necessary condition for the game to be winnable, but it is not sufficient.

Example 7. The following initial configuration on the triangle graph is not winnable, despite there being a non-negative total on the graph.

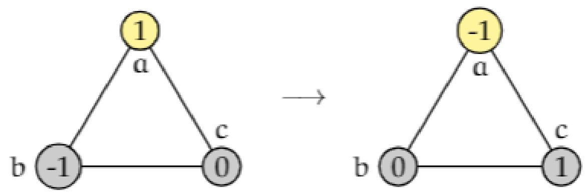

Figure 3.3: Example of an unwinnable configuration with total dollars equal to zero on a triangle graph.

Notice that if the positive vertex a lends to its neighbors, we end up with a clockwise rotation of the original configuration. If the negative vertex borrows, the configuration rotates counterclockwise. No matter how we try to structure our lending and borrowing moves, we will always end up with some rotation of the initial configuration eventually.

We can actually prove this configuration is unwinnable. Since we can look at the borrowing moves of the game as if they happen simultaneously, then we can prove that the above example is not winnable. Say vertex a borrows x dollars from its two neighbors, vertex b borrows y dollars from its two neighbors, and vertex c borrows z dollars from its two neighbors. Then for there to be a winnable solution, it would need to satisfy the following system of equations:

$$\begin{cases} 2x - y - z + 1 = 0 \\ -x + 2y - z - 1 = 0 \\ -x - y + 2z = 0 \end{cases}$$

Solving this system of equations we obtain $x = z - \frac{2}{3}$ and $y = z - \frac{1}{3}$. Since vertices can only borrow whole dollar amounts, this is an unwinnable game.

Example 8. The following configuration on the triangle graph is clearly winnable, even though it has the same total dollar amount as our first triangle graph in the previous example. If a lends \$1, then the game is won. Alternatively, c could borrow \$1 followed by b borrowing \$1.

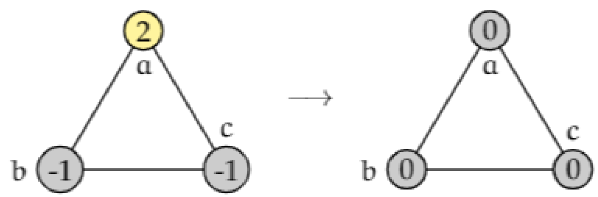

Figure 3.4: Example of a winnable configuration with total dollars equal to zero on a triangle graph.

It might be tempting to think that the above configuration is only winnable because the debt is split evenly between a's neighbors. However, there are examples where that is not the case, and our proof that the first triangle configuration wasn't winnable holds the key. The first configuration wasn't winnable because it required non-integer solutions. In fact, it required rational solutions where the divisor was three.

Example 9. It may then come as no surprise that the following configuration does have a solution.

Solving the following system of equations:

$$\begin{cases} 2x - y - z + 3 = 0 \\ -x + 2y - z - 3 = 0 \\ -x - y + 2z = 0 \end{cases}$$

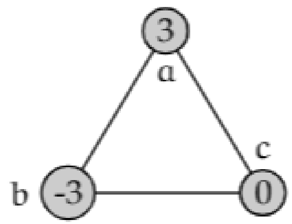

Figure 3.5: Another example of a winnable configuration with total dollars equal to zero on a triangle graph.

we obtain $x = z - 1$ and $y = z + 1$. One possible solution to this is $x = 0, y = 2, z = 1$. In order for the figure, b borrows \$2 and then c borrows \$1.

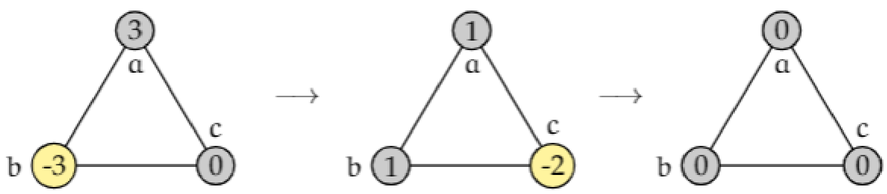

Figure 3.6: Solution to triangle example.

Theorem 23. *A dollar configuration with total zero dollars on a triangle graph is winnable if and only if the difference between initial dollar amounts of any two vertices is a multiple of three.*

Proof. Let $D(G)$ be an initial dollar configuration on a triangle graph G such that the total dollar amount on the graph is zero dollars. Let vertex a have A dollars, vertex b have B dollars and vertex c have C dollars. Let a borrow x dollars, b borrow y dollars, and c borrow z dollars. Then the following system of equations must have an integer solution

$$\begin{cases} 2x - y - z + A = 0 \\ -x + 2y - z + B = 0 \\ -x - y + 2z + C = 0 \end{cases}$$

where $A + B + C = 0$. Solving this system of equations we obtain:

$$
\begin{aligned}
A - C &= \frac{-4a - 2b}{6} = \frac{-(2a + b)}{6} \\
b - c &= \frac{-a - 2b}{3}.
\end{aligned}
$$

This implies that $(2a + b) = 3k$ for $k \in \mathbb{Z}$ and $(a + 2b) = 3J$ for $J \in \mathbb{Z}$. Then $2a + b = 3k$ implies $b = 3k - 2a$. Substituting into $A + B + C = 0$ we obtain

$$
\begin{aligned}
A + (3k - 2A) + C &= 0 \\
3k - A + C &= 0 \\
-A + C &= -3k \\
A - C &= 3k.
\end{aligned}
$$

This implies that $A \equiv C \bmod 3$. Additionally $A + 2B = 3J$ implies $A = 3J - 2B$. Again, substituting into $A + B + C = 0$ we obtain

$$
\begin{aligned}
(3J - 2B) + B + C &= 0 \\
3J - B + C &= 0 \\
-B + C &= -3J \\
B - C &= 3J.
\end{aligned}
$$

This implies $B \equiv C \bmod 3$, which implies that $A \equiv B \bmod 3$. Thus, the difference between the initial dollar amounts of any two vertices is a multiple of three. $\square$

3.1.1 The Impact of the Graph's Betti Number

Definition 24. For a graph G, let V be the number of vertices of G, E the number of edges of G. Then the **Betti number** of G is $E - V + 1$. This can also be thought of as the minimum number of edges one would need to delete from G to obtain a tree [3].

Note that for a graph G, if we consider $T =$ a spanning tree of G, then T has $v - 1$ edges. Then there are $e - (v + 1)$ edges of G that are not contained in T. If we add any of those remaining edges to T we will introduce a cycle to the graph.

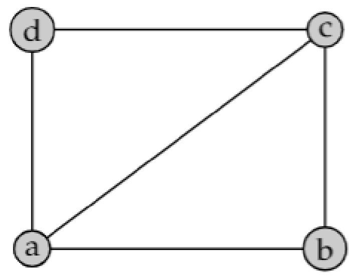

Figure 3.7: The Betti number of G is 5 - 4 + 1 = 2. If we remove edges dc and ac we will have a tree.

Theorem 25. *[3] For a graph G with initial dollar configuration $D(G)$, if the total dollar amount is greater than or equal to the graph's Betti number, then the game is winnable.*

Baker proved this theorem in his paper with Algebraic Geometry using Riemann-Roch and Abel-Jacobi theory. It was our aim to provide a graph theoretical proof of this theorem.

First, notice that if G is a tree, then the Betti number $E - V + 1 = (V - 1) - V + 1 = 0$. This supports the fact that a tree is winnable with any nonnegative dollar total on the graph.

We will show this for a single cycle with total dollars on the graph equal to the Betti number, which for a cycle is one.

Proposition 26. *If* C *is a cycle with dollar configuration* $D(C)$ *such that the total number of dollars on the cycle is one, then we can move a dollar from positive vertices to a negative vertex without bringing other vertices into debt.*

Proof. Let C be a cycle with dollar configuration $D(C)$ such that the total number of dollars on the cycle is one. If there are no negative vertices then the game is already won. So assume there is a negative vertex v. For clarity, orient the cycle so v is at the bottom in approximately the six o'clock position. There must be at least one positive vertex. Let x be the closest positive vertex to v traveling clockwise from v around the cycle and let y be the closest positive vertex to v traveling counterclockwise around the cycle. Note that it may be that $x = y$. In that case, $x = y$ would have at least \$2 since the total must be equal to one.

Let P_1 be the path $x = x_o x_1 x_2 ... x_n = v$ be the (x, v)-path on the left of the cycle and let P_2 be the path $y = x_0 y_1 y_2 ... y_m = v$ be the (y, v)-path on the right hand side of the cycle. Let n be the length of P_1 and let m be the length of P_2. Without loss generality, suppose $n \leqslant m$. Then have all vertices on the (x, y)-path that does not run through v lend n dollars. Then for each i, vertex x_i lends $n - i$ dollars. Additionally, for each i, vertex y_i lends $(n - i)$ dollars, unless $(n - i)$ is negative, in which case the vertex lends nothing.

Consider vertex w after these moves.

Case one: $w = x$ or $w = y$. Then w lends n dollars to each of its two neighbors, but gets back n dollars from one neighbor and $n - 1$ dollars from the other neighbor. If w starts with d dollars, then w finishes with $d - 2n + n + (n - 1) = d - 1$. So, each of the two positive vertices lose one dollar.

Case two: w is a vertex on the (x, y) path that does not go through v. Then w lends n dollars to each of its neighbors, but also receives back from each of its neighbors. This is an even exchange of money, so the dollar amount on w remains unchanged.

Case three: w lies on P_1. Then $w = x_k$ for some k. Suppose w starts with d dollars. Then w lends $n-k$ to each of its two neighbors, but gains $n-k+1$ dollars from one neighbor, and gains $n-k-1$ from the other neighbor. So w is left with $d-2(n-k)+(n-k+1)-(n-k-1) = d$ dollars, so w total dollars remains unchanged.

Case four: $w = v$ or w lies on P_2 and is the first vertex that doesn't lend money. Then w receives one dollar from its neighbor and does not lend anything back, so w's dollars increase by one. If $w = v$ and $n = m$ then v is the first vertex that doesn't lend for both P_1 and P_2. In that case, v increases by two dollars.

Case five: w lies on P_2 and does not lend, not does it receive anything from its neighbors. Then clearly its dollar amount doesn't change.

Therefore, positive dollars can be moved to a negative vertex without bringing any other vertices into debt. □

Example 10. See figure 3.8. In the following example, we have two vertices, e and i with positive dollar amounts, and one vertex, a with a negative dollar amount. We wish to move a dollar to the negative vertex a. We begin by having all vertices on the (e,i)-path that does not run through a each lend the dollar amount equal to $\min\{d(e,a), d(i,a)\}$ which in this case is $d(i,a) = 3$. The vertices on the interior path do not change their dollar amount, since this is an even exchange of currency. The vertices e and i are each in the negative by two dollars from the excess money loaned to d and j respectively.

In the second step, vertices d and j each lend one dollar less than they received, so in this case, they each lend two dollars. This pays back the debt to vertices e and i respectively. Additionally vertices c and k increase to two dollars each, and d and j are each at negative one dollar from the excess lent to c and k respectively.

Finally, in the third step, vertices c and k each lend one dollar less than they received, so in this case, they each lend one dollars. This pays back the debt to vertices d and j respectively. Additionally vertices b and a increase by one dollar, giving b one dollar and a coming out of the negative to zero dollars.

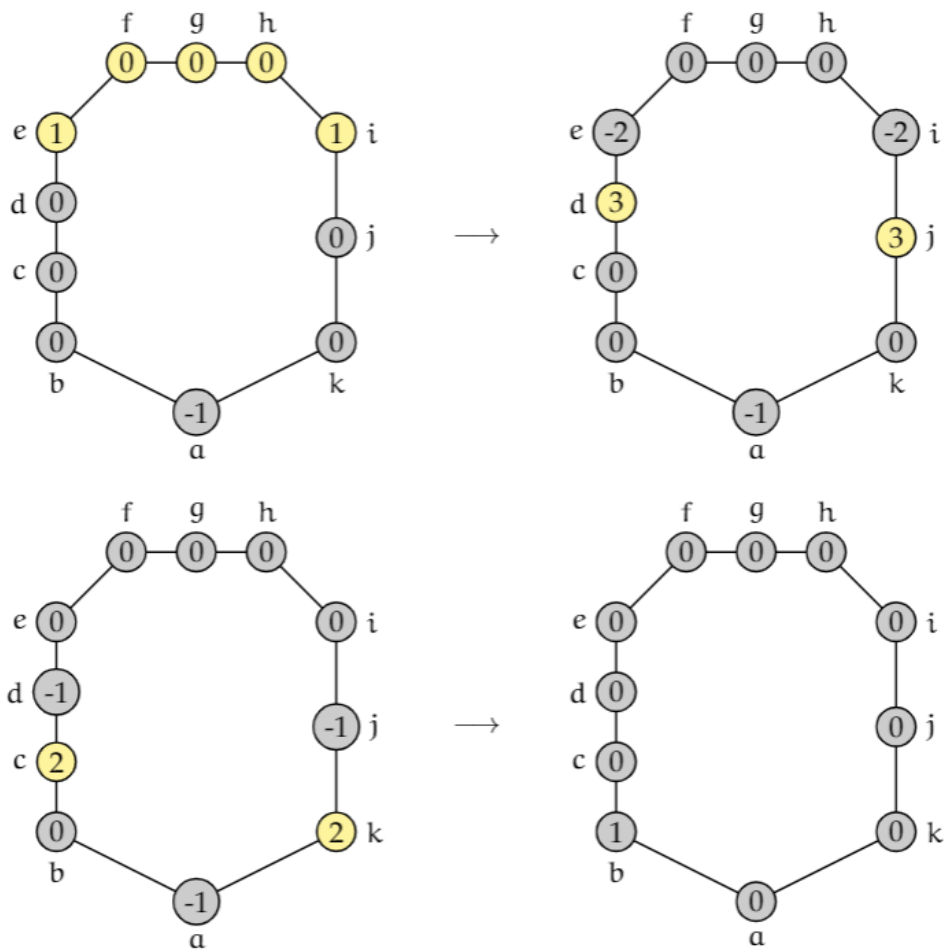

Figure 3.8: Example of moving a dollar to a negative vertex on a cycle.

3.1.2 Attempts at a General Proof

We have made several attempts at a general proof. At best we were able to prove that we can get the graph down to having only one vertex in the negative. We will present two of these proofs here. The first attempt at a proof was induction on the Betti number of the graph.

Proposition 27. *Given a connected graph* G *with dollar configuration* $D(G)$*, if the total number of dollars on the graph is equal to or exceeds the Betti number of the graph, then there is a sequence of borrowing moves that leads to at most one negative vertex on the graph.*

Proof. We know this is true for Betti number equal to zero by Theorem 22. Now suppose it is true for graphs with Betti number B and consider a graph with Bettie number $B+1$. Consider subgraph $G' = G - e$ where e is a non cut edge. Let x, y be the endpoints of e. Then the Betti number is decreased by one, and by the induction hypothesis, we know there is a solution.

Apply this solution to the original graph G by having all of the vertices in G borrow the same amount that they borrowed in the solution for G'.

Case one: In the solution, both x and y borrow the same dollar amount. Then replacing the edge creates an even exchange between the two, and their values don't change. The solution still works and the game is won.

Case two: In the solution, x and y borrow different amounts. Without loss of generality suppose x borrows more than y. If y does not have the excess currency available to lend this difference to x, y will now be in debt, but it will be the only vertex in debt. □

The second proof was an attempt to induct on the number of vertices in the graph.

Proof. Suppose we have a simple connected graph with $n = 2$ vertices with dollar configuration $D(G)$ such the the total dollars on the graph is greater than or equal to the Betti number of the graph. Then the graph is P_2 and we know by Theorem 22 there is a sequence of borrowing moves that will win the game.

Assume this is true for all graphs with k vertices. Suppose G has $k+1$ vertices. Choose an edge with endpoints x and y. Suppose x has a dollars and y has b dollars. Create G' by contracting the edge xy, and combining the dollar amounts from both vertices. This may create multiple edges in the graph G'. These are needed to maintain the vertex degrees from the original graph. This new combined vertex has $(a+b)$ dollars. By the induction hypothesis, there is a sequence of borrowing moves the solves the game. Uncontract the edge, and utilize the same borrowing moves in the original graph. Have vertices x and y both borrow the same amount as the combined vertex in the solution from G'.

We know that all vertices besides x and y are positive in G, since vertex degrees were maintained in G'. However, in general, while the new dollar amounts on x and y sum to a positive amount, it is possible that one of the two vertices is negative. $\square$

While there is still a lot that we don't completely understand about the Dollar Game on more complex arbitrary graphs, we will continue to work towards a general graph theoretical proof for arbitrary graphs, and hope to make further progress in the near future.

Bibliography

[1] R. Anderson, L. Lovasz, P. Shor, J. Spencer, E. Tardos, and S. Winograd, "Disks, balls and walls: analysis of a combinatorial game", *The American Mathematical Monthly* **vol 96, no. 6** (1989) 481–493.

[2] N. L. Biggs, "Chip-firing and the critical group of a graph", *Journal of Algebraic Combinatorics* **vol 9, no. 1** (1999) 25–45.

[3] M. Baker and S. Norine, "Riemann-roch and abel-jacobi theory on a finite graph", *Advances in Mathematics* **vol 215, no. 2** (2007) 766–788.

[4] A. Bjorner, L. Lovasz, and P. W. Shor, "Chip-firing games on graphs", *European Journal of Combinatorics* **vol 12, no. 4** (1991) 283–291.

[5] J. A. Bondy, U. S. R. Murty *et al*, *Graph Theory with Applications*, volume 290, Macmillan, London, 1976.

[6] Reinhard Diestel, *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*, Springer-Verlag, Berlin, third edition, 2005.

[7] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*, Springer Science & Business Media, New York, 2013.

[8] G. Musiker, "Algebraic combinatorics lecture notes 29-31 ", *MIT OpenCourseware* (2009).

[9] J. Spencer, ″Balancing vectors in the max norm″, *Combinatorica* **vol 6, no. 1** (1986) 55–65.

Vita

Rachel Koch was born in Abington, Pennsylvania on December 19, 1980. She graduated from Piedmont Virginia Community College with an Associate's degree in computer science in May of 2016. She graduated with a Bachelor's degree in Mathematics from James Madison University, and she continues her studies to pursue a Master's degree in Mathematics from Virginia Commonwealth University.