2013

Chip Firing Games and Riemann-Roch Properties for Directed Graphs

Joshua Z. Gaslowitz
Harvey Mudd College

Chip Firing Games and Riemann–Roch Properties for Directed Graphs

J. Zachary Gaslowitz

Dagan Karp, Advisor

Dustin Cartwright, Reader

HARVEY MUDD COLLEGE

Department of Mathematics

May, 2013

Abstract

The following presents a brief introduction to tropical geometry, especially tropical curves, and explains a connection to graph theory. We also give a brief summary of the Riemann–Roch property for graphs, established by Baker and Norine (2007), as well as the tools used in their proof. Various generalizations are described, including a more thorough description of the extension to strongly connected directed graphs by Asadi and Backman (2011). Building from their constructions, an algorithm to determine if a directed graph has Row Riemann–Roch Property is given and thoroughly explained.

Contents

Acknowledgments

Great thanks to Spencer Backman, for his suggestions of project ideas and help throughout the project, and to Dagan Karp, my fantastic advisor.

Chapter 1

Tropical Curves

1.1 Motivation

Before looking at Riemann–Roch, we give a (very) brief introduction to Tropical Geometry, which we will use as one possible motivation for studying Riemann–Roch in graphs. The connection is expanded upon in Section 1.5.

1.2 The Tropical Semiring

The Tropical Semiring is comprised of the set $\mathbb{R} \cup \{\infty\}$ along with the operations of tropical addition, $\oplus$, and tropical multiplication, $\odot$, which are defined as

$$x \oplus y = \min(x, y) \qquad\qquad x \odot y = x + y.$$

Notice that both operations are commutative and associative, and that multiplication distributes over addition. This is clear from the definitions of these operations:

$$\begin{gathered} x \odot (y \oplus z) = x \odot y \oplus x \odot z \\ \Updownarrow \\ x + \min(y, z) = \min(x + y, x + z). \end{gathered}$$

This is only a semiring because we will not, in general, have additive inverses. (For example, there is no x such that $3 \oplus x = 10$.)

Upon a moment's consideration, we see that a monomial in the tropical semiring corresponds to a linear function, for example $a \odot x^3 = 3x + a$.

Then a polynomial in n variables in the tropical semiring is the minimum of a finite set of n–dimensional hyperplanes.

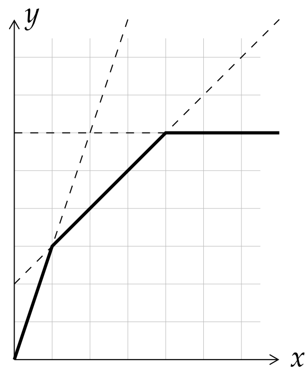
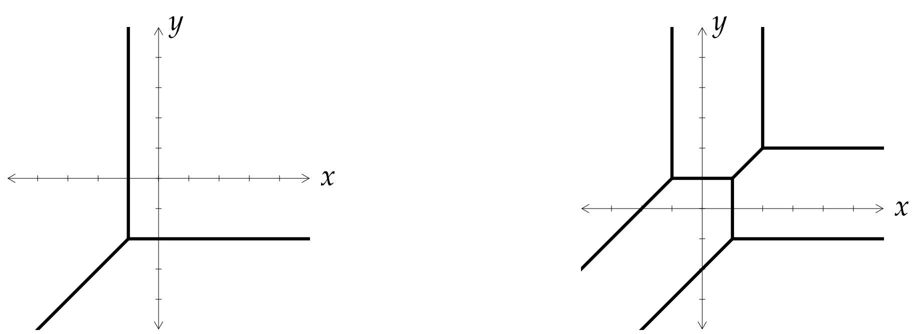


Figure 1.1 The Tropical Polynomial $p(x) = x^3 \oplus 2 \odot x \oplus 6$

1.3 Tropical Curves

An important object of study in tropical geometry is the tropical curve, Tropical curves are defined as the subset of $\mathbb{R}^2$ on which a given polynomial in 2 variables is not differentiable. Intuitively, these are the points in the domain that correspond to to the "corners" of the polynomial, where two or more of planes intersect and form part of the lower hull.



a. A Linear Tropical Curve **b.** A Quadratic Tropical Curve

Figure 1.2 Some examples of tropical curves

1.4 Tropical Curves as Graphs

As is suggested by Figure 1.2, tropical curves can be viewed as having the structure of a graph by interpreting the line segments as edges and intersections as vertices. More specifically, it is common to ignore some of the structure of a particular tropical curve to convert it into a purely graph theoretic object, referred to it as an "Abstract Tropical Curve". One way to do this (as in, for example, Gathmann and Kerber (2007)) is to keep track of the lengths of the line segments (some of which are infinite) and preserve the connections, but ignore the specific embedding in $\mathbb{R}^2$. This connected graph with a number associated with each edge is known as a metric graph. The hope, then, is that we can study tropical curves by studying metric graphs, giving us the flexibility to attack questions about tropical curves from either an algebraic perspective or a combinatorial one.

1.5 Looking Towards Riemann–Roch

There are many surprising tropical analogues of theorems in classical algebraic geometry. One such example is the very important Bezout's Theorem, concerning the number of intersection points between two curves. Given the importance of the Riemann–Roch Theorem in the classical setting, it is natural to ask if there is an analogous result for tropical curves. Motivated by the abstract tropical curves described in the previous section, one way to approach this would be to ask if there are Riemann–Roch style results on graphs. Such results have recently been discovered (see Baker and Norine (2007) and Asadi and Backman (2011), for example), and these will be the main topic of the rest of this thesis.

Chapter 2

Riemann–Roch, Chip–Firing Games, and Lattices

2.1 A Riemann–Roch Theorem for Finite Graphs

Matthew Baker and Serguei Norine formulated a Riemann–Roch theorem for finite, undirected multigraphs. Recall that a multigraph G is defined to have a set of vertices $V(G)$ and a multiset of edges $E(G)$, where edges connect a pair of vertices and a pair may have multiple edges between them. Their result is analogous to the established Riemann–Roch theorem for Riemann surfaces and algebraic curves.

Their proof is largely combinatorial, and makes use of the chip–firing game described in the following subsection (see Baker and Norine (2007) for more details). Although not always the language used in Baker and Norine (2007), we will define as much as possible by building on the combinatorial interpretation provided by this game.

2.1.1 The Chip–Firing Game of Baker and Norine

In their work, Baker and Norine describe the following chip–firing game. The game is played on a finite, undirected multigraph G. Each vertex has an integer number of chips associated with it (where the vertex is said to be "in debt" if that number is negative). A vertex can either fire, where it sends one chip along each of its edges, or borrow, in which case it receives one chip along each of its edges. The task is then to determine which configurations can, through some sequence of borrowings and firings, bring all vertices out of debt.

2.1.2 Some preliminary definitions

For the following definitions, we consider a finite, undirected multigraph G with $n+1$ vertices labeled $v_1, \ldots, v_{n+1}$.

Definition 2.1. *A* divisor *of the graph G is a vector in $\mathbb{Z}^{n+1}$ that represents the configuration (as in the above chip–firing game) in which the vertex v_i has $D(v_i)$ chips.*

As a side note, Baker and Norine define divisors as elements of $\mathrm{Div}(G)$, the free abelian group on the vertices of G. This definition is equivalent in the case of graphs, but allows them to cleanly define a more general condition for when the Riemann–Roch formula will hold, which we will not discuss.

Definition 2.2. *The* degree *of a divisor D, denoted $\deg(D)$, is the sum of the elements of D, i.e*

$$\deg(D) = \vec{\mathbf{1}} \cdot D.$$

The degree thus counts the total number of chips in the corresponding configuration.

Firing the vertex v_i will always have the same effect: specifically subtracting v_i's degree from its chips and adding to every other vertex one chip per edge between it and v_i. Thus, firing v_i has the effect of subtracting a fixed vector in $\mathbb{Z}^{n+1}$ from the current divisor. This vector is exactly the i^{th} row of the Laplacian of G, defined as follows.

Definition 2.3. *The* Laplacian *of an undirected graph G is defined as $\mathcal{D} - \mathcal{A}$, where $\mathcal{D}$ is a diagonal matrix where $\mathcal{D}_{ii}$ is the degree of v_i and $\mathcal{A}$ is the adjacency matrix of G, i.e. $\mathcal{A}_{ij}$ is the number of edges between v_i and v_j.*

The order in which vertices are fired will not affect the final configuration, so we can, without ambiguity, summarize a sequence of firings as follows.

Definition 2.4. *A* firing strategy, *f, is a vector in $\mathbb{Z}^{n+1}$ that we interpret as a sequence of firings and borrowings where $f(v_i)$ is the difference between the number of times v_i fires and the number of times that it borrows.*

Performing the firing strategy f starting from the divisor D will yield the divisor

$$D' = D - Q^{\mathrm{T}} f,$$

where Q is the Laplacian of the graph in question. The astute reader will note that the Laplacian of an undirected graph is always symmetric. I have chosen to use the transpose in the above equation to parallel what will be necessary in the row chip–firing game, discussed in Section 3.1. Notice that a sequence of firings/borrowings is equivalent to a walk along the lattice spanned by the rows (or columns) of Q.

Definition 2.5. *We say that the divisors D and D' are* equvalent *if you can get from one to the other through a sequence of firings, i.e. $D' - D$ is an element of the lattice spanned by the rows of Q. This is denoted by the equivalence relation $\sim$.*

Definition 2.6. *A divisor is considered* effective *if all vertices have a non-negative number of chips (we say that all of the vertices are out of debt).*

Definition 2.7. *The* linear system *associated to a divisor D is defined as*

$$|D| = \{E \ : \ D \sim E, E \geq \vec{\mathbf{0}}\},$$

the set of equivalent, effective divisors. In the language of the chip firing game, this is the set of configurations that can be reached starting from D that bring all of the vertices out of debt.

We now have the tools needed to define the rank of a divisor.

Definition 2.8. *The* rank *of a divisor D, denoted $r(D)$, is -1 if there is no firing strategy that brings all vertices out of debt. Otherwise, $r(D)$ is the largest non-negative integer r such that any way of removing of r chips from the game still results in a configuration that can be brought out of debt by some firing strategy. Equivalently,*

$$r(D) = \min\{\deg(E) \ : \ |D - E| = \emptyset, E \geq \vec{\mathbf{0}}\} - 1.$$

Theorem 2.9 (Baker and Norine)**.** *For any finite, undirected multigraph G, define the canonical divisor, K, to be the configuration that has $\deg(v) - 2$ chips on each vertex v. Then*

$$r(D) - r(K - D) = \deg(D) + 1 - g,$$

where $g = |E(G)| - |V(G)| + 1$ is the genus of G.

2.2 Generalization to Lattices

Omid Amini and Madhusudan Manjunath took the work of Baker and Norine (2007) in a different direction. The linear system associated with a divisor, and by extension its rank, is characterized entirely by the lattice spanned by the rows of the Laplacian matrix of a given graph. By rephrasing some of the definitions, then, this can be seen as a statement about certain lattices having the Riemann–Roch property. Amini and Manjunath use this to define a Riemann–Roch property for any full rank sub–lattice of the root lattice, A_n (i.e. the integer points normal to $\vec{\mathbf{1}}$). This corresponds to the same property as Baker and Norine's Riemann–Roch theorem for graphs when the lattice is generated by the rows of some graph's Laplacian. However, it is not a property that all lattices, in general, will have. Instead of the combinatorial approach taken by Baker and Norine (2007) of studying a chip–firing game, Amini and Manjunath introduce a geometric characterization of lattices that have the Riemann–Roch property, giving necessary and sufficient conditions for when a lattice will have this property. Specifically, a full rank sub–lattice of the root lattice has the Riemann–Roch property if and only if it is uniform and reflection invariant, see Amini and Manjunath (2010). This is all we will say at the moment, because in the next chapter we will look more closely at these constructions as we look at a generalization provided by Asadi and Backman (2011).

Chapter 3

Riemann–Roch for Directed Graphs

Throughout this chapter, we will let G to be a strongly connected, directed multigraph with $n+1$ vertices labeled $v_1, \ldots, v_{n+1}$.

Following with Asadi and Backman (2011), we discuss two separate Riemann–Roch properties, each corresponding to a lattice built from the directed Laplacian of G (Definition 3.1). In doing so they generalize the lattice results of Amini and Manjunath (2010), and utilize connections to chip–firing games analogous to those of Baker and Norine (2007). Except where specified, definitions and theorems follow Asadi and Backman.

3.1 The Row Chip Firing Game

We first point out that, on a directed graph, we can play a chip firing game that is analogous to the one used by Baker and Norine. We will again assign an integer number of chips to each vertex, but since we now have directed edges, we must make a choice as to what a firing should be. A natural choice would be for a vertex to send a chip along each of its outgoing edges, losing as many chips as its outdegree, when it fires. We again define borrowing to be the opposite of firing, i.e. the vertex receives a chip along each of its outgoing edges.[1] Just as in the undirected version, de-

[1] This may seem like a less than intuitive notion of borrowing, and perhaps you think it would be better to keep our definition of firing, but borrow by receiving a chip along each incoming edges. This does seem fairly natural, as the movement of chips respect the direction of the edges in both cases, and I encourage you to explore such a game. However, this variant is not discussed in the Asadi and Backman (2011), so while it would undoubtably be

scribed in Section 2.1, firing a vertex v_i corresponds to subtracting a fixed vector from the divisor. Analogously, that vector is exactly the i^{th} row of the directed Laplacian matrix Q, which is defined similarly to the undirected version (see Definition 2.3).

Definition 3.1. *The* Laplacian *of a directed graph G is defined as $\mathcal{D} - \mathcal{A}$, where $\mathcal{D}$ is a diagonal matrix where $\mathcal{D}_{ii}$ is the outdegree of v_i and $\mathcal{A}$ is the adjacency matrix of G, i.e. $\mathcal{A}_{ij}$ is the number of directed edges from v_i to v_j.*

We again think about a sequence of firings as a vector $f \in \mathbb{Z}^{n+1}$, where

$$f(v_i) = \text{\# of times } v_i \text{ fires} - \text{\# of times } v_i \text{ borrows},$$

and just as before, starting at a divisor D, carrying out firing strategy f yields

$$D' = D - Q^{\mathrm{T}} f.$$

The graph G the Row Riemann–Roch property if and only if the lattice spanned by the rows of directed Laplacian of G has the Riemann–Roch property.

3.2 The Column Chip Firing Game

There is another very natural lattice that we can build from the directed Laplacian – specifically the lattice spanned by its columns, rather than its rows. (Note that these two lattices are identical for an undirected graph, because the Laplacian is symmetric.) This begs the question "What is the corresponding chip firing game?" Similar to the row chip firing game above, we want the columns to be the vector by which a divisor is shifted when the corresponding vertex is fired. Thus, the column chip firing game defines firing as sending one chip along each incoming edge, but still losing chips equal to its outdegree, and again borrowing is the just the inverse. This game is interesting, in that the number of chips is not necessarily conserved. None the less, a combinatorial explanation is given, which will be discussed shortly.

Now, we would like to say that the G has the column Riemann–Roch property if this lattice does, and we will, but there is a problem. This lattice is also n dimensional, but it is no longer in general normal to $\vec{\mathbf{1}}$ (which is, in essence, a statement that this chip firing game does not conserve chips).

interesting, I am not sure of any connections it would have to these or other Riemann–Roch results.

The next result will show us that this lattice will be normal to some strictly positive integral vector, and will also be important in the study of the row chip–firing game. After that we will see how Backman and Asadi were then able to extend the lattice results of Amini and Manjunath to apply to the lattices associated to these column chip–firing games.

3.3 A Preliminary Result

The following lemma appears in Asadi and Backman (2011).

Lemma 3.2 (Lemma 3.1 in Asadi and Backman (2011))**.** *A directed graph G is strongly connected if and only if there exists a vector $R \in \mathbb{N}^{n+1}$, unique up to multiplication by a real constant, such that $Q^T R = 0$.*

We now present an equivalent statement of this lemma, more in the language of the row chip firing game, and a modified and expanded version of the original proof.[2]

Lemma 3.3. *For any directed graph G with Laplacian Q, there exists a firing strategy for the row chip firing game $f \neq \vec{\mathbf{0}}$ such that $Q^T f = \vec{\mathbf{0}}$. Furthermore, G is strongly connected if and only if all such firing strategies have either all vertices firing or all vertices borrowing and the nullspace of Q^T is one dimensional.*

Proof. Consider any directed graph G with $n+1$ vertices and Laplacian matrix Q. Because firing a vertex preserves the number of chips, $Q\vec{\mathbf{1}} = \vec{\mathbf{0}}$. This implies that $\text{rk}(Q) \leq n$. Thus, the rows of Q are not all linearly independent, so $Q^\text{T} f = \vec{\mathbf{0}}$ has a nontrivial solution. Note that any such f can be scaled to vector in $\mathbb{Z}^{n+1}$.

We now assume that G is strongly connected, and consider any firing strategy f that makes $Q^\text{T} f = \vec{\mathbf{0}}$. Let V^+ be the set of vertices that fire under this strategy, i.e. those v_i for which $f(v_i) > \vec{\mathbf{0}}$. There cannot be a net flow of chips out of V^+, or else there would have to be at least one $v_i \in V^+$ with $(Q^\text{T} f)(v_i) < 0$, in contradiction to our choice of f. This is only possible if, in particular, there are no edges from any firing vertex to any non-firing vertex. But because G is strongly connected, this means that V^+ is either empty or contains all vertices. This also rules out the possibility of f having both 0 and negative components, for negating such an f would give a firing strategy that had both positive and 0 components, impossible because

[2]There was an error in the original proof, and I found it helpful to work through the details myself to better understand it. This exploration is what appears here.

$Q^{\mathrm{T}}(-f) = \vec{\mathbf{0}}$. Thus, as desired, any non-zero firing strategy that does not move any chips will either fire all vertices, or have them all borrowing.

To show $\dim(\ker(Q^{\mathrm{T}})) = 1$, we consider any two non-zero firings strategies f_1 and f_2 satisfying $Q^{\mathrm{T}} f_1 = Q^{\mathrm{T}} f_2 = \vec{\mathbf{0}}$. Because neither is all zero, we know that they have no zero components. But then the linear combination $f_2(v_1) f_1 - f_1(v_1) f_2 = \vec{\mathbf{0}}$, because the v_1 component is 0. Thus, any such f_1 and f_2 will be linearly dependent.

To prove the other direction, we now assume that G is not strongly connected. Then we can partition the vertices into maximal strongly connected components $V_1, V_2, \ldots, V_m$ such that no element of V_1 has outgoing edges to any vertex outside of V_1. This guarantees that any firings in V_1 will not affect the vertices in other components. Thus, we can extend the firing strategy that has no effect on V_1 (which exists from applying the above result to the subgraph defined by V_1) to one for all of G by not firing or borrowing at any other vertex. Thus, all graphs that are not strongly connected will have a firing strategy that does not move any chips, and contains both zero and non-zero elements.

These together prove the desired result. □

This lemma tells us that each of the columns, and thus the lattice spanned by the columns, is normal to this vector which may not be $\vec{\mathbf{1}}$, but will be positive and integral in each component. The work of Amini and Manjunath only considers lattices normal to $\vec{\mathbf{1}}$, so in order to describe the column Riemann–Roch property as the Riemann–Roch property of the associated lattice, Asadi and Backman extended their results to lattices normal to any $R \in \mathbb{N}^{n+1}$. This will be expanded upon in the next section.

This lemma also tells us that in playing the row chip firing game, there are always nonzero firing strategies that do not move any chips. This motivates us to define an equivalence relation on firing strategies.

Definition 3.4. *Two firing strategies (for the row chip firing game) are considered equivalent, denoted $f \approx f'$, if $Q^{\mathrm{T}}(f - f') = \vec{\mathbf{0}}$.*

Remark 3.5. *While this is not a difficult definition to understand, the interpretation is a bit more transparent when it is pointed out that this is just saying that $Q^T f = Q^T f'$; firing strategies are equivalent if they move around chips in the same way.*

Note that in the undirected version of this game, we did not have the same flexibility in which firing strategies don't move any chips. Indeed, such strategies are always multiples of $\vec{\mathbf{1}}$: if you fire every vertex once, then along each edge, one chip has been pushed in each direction, so there is no net change. This turns out to be quite a significant difference between the directed and undirected versions of this game (see Theorem 3.19).

We now point out a divergence from the notation used in Asadi and Backman (2011). They use R (as introduced in the original statement of the above lemma) throughout their discussion of the row chip firing game to refer to the smallest of the strictly positive integer firing strategies guaranteed by this lemma (although the minimality is not explicitly stated). Throughout their article, R is used to denote a vector to which a lattice is normal, and this is that vector for the lattice spanned by the columns of Q. However, when interpreting it as a firing strategy for the row chip–firing game (whose lattice will, recall, be normal to $\vec{\mathbf{1}}$), I will instead refer to the smallest positive firing strategy that does not move any chips as $\hat{f}$.

3.4 Riemann–Roch for Lattices

We will now be a bit more specific with what it means for a lattice to have the Riemann–Roch property, and describe briefly how Asadi and Backman extended the results of Amini and Manjunath to apply this to, in particular, the lattice associated with the column chip firing game.

First, a bit of notation. Fix $R \in \mathbb{N}^{n+1}$. We define $H_R \subset \mathbb{R}^{n+1}$ to be the hyperplane $\{x \in \mathbb{R}^{n+1} \ : \ x \cdot R = \vec{\mathbf{0}}\}$. Then Λ_R refers to the integral points of H_R, i.e. $H_R \cap \mathbb{Z}^{n+1}$. Finally fix some Λ, a full rank sublattice of Λ_R. Notice that the lattice associated with the row chip firing game is a full rank sublattice of $\Lambda_{\vec{\mathbf{1}}}$ and the lattice associated with the column chip firing game is a full rank sublattice of Λ_R, where $R = \hat{f}$.

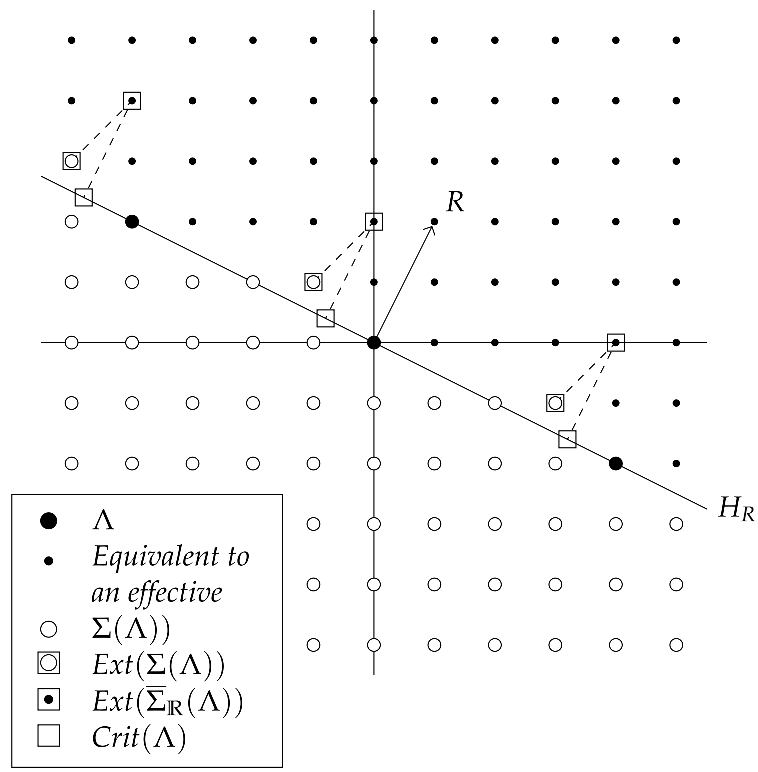
We now generalize the degree of a divisor as follows:

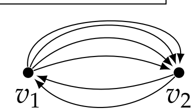
Definition 3.6. *The degree of a divisor D, denoted* $\deg_R(D)$ *is defined to be $R \cdot D$.*

Notice that when $R = \vec{\mathbf{1}}$, and when we can interpret the lattice as coming from a chip firing game, this definition coincides with the definition used by Baker and Norine (2007), namely the total number of chips on the graph. Note that the degree of a divisor is constant under translation by a vector in Λ.

This suggests a fairly natural combinatorial interpretation for the column chip-firing game: if each vertex v has its own currency, so that 1 of v's chips is worth $R(v)$ chips in a universal currency, then the total value in the universal currency (which is exactly what is captured by $\deg_R$) is preserved by firings of the column chip-firing game.

Example. *We include this example, illustrating some of the definitions in this section, as they apply to the lattice spanned by $(4, -2)$, a full rank sublattice of $\Lambda_{(1,2)}$. Although these constructions depend only on the lattice, we also provide a graph who's column chip firing game uses this lattice.*



$G =$ 

$$Q = \begin{bmatrix} -4 & 4 \\ 2 & -2 \end{bmatrix}$$

Just as before, two divisors are equivalent if they differ by a point in Λ, a divisor is effective if each entry is nonnegative, and the linear system associated with a divisor D is the set of equivalent, effective divisors:

$$|D| = \{E \,:\, D \sim E, E \geq \vec{\mathbf{0}}\}.$$

We now define the rank of a divisor in a more general context than before.

Definition 3.7. *The* rank *of a divisor D, denoted $r(D)$, is still given by*

$$r(D) = \min\{\deg(E) \ : \ |D - E| = \emptyset, E \geq \vec{\mathbf{0}}\} - 1.$$

With the generalized notion of degree, we must proceed a little more carefully: $r(D)$ is -1 if D is not equivalent to an effective divisor, and otherwise, $r(D)$ is the largest non-negative integer r such that for all E with $\deg(E) = r$, $D - E$ still results in a configuration that is equivalent to an effective divisor.

Definition 3.8. *We define $\Sigma(\Lambda)$ to be $\{D \in \mathbb{Z}^{n+1} \ : \ D \not\geq p, \forall p \in \Lambda\}$, or in the language of the chip firing game, $\Sigma(\Lambda)$ is the set of divisors that cannot be formed by adding chips to a divisor in Λ.*

Remark 3.9. *If Λ is associated to one of our chip-firing games, $\Sigma(\Lambda)$ is exactly the set of divisors that are not equivalent to any effective divisor. To see this, note that for any divisor D,*

$$\begin{aligned} E \sim D \text{ is effective} &\iff D - E = \lambda \in \Lambda \\ &\iff D = \lambda + E \\ &\iff D \notin \Sigma(\Lambda). \end{aligned}$$

Definition 3.10. *The set of extreme divisors of $\Sigma(\Lambda)$ is denoted $Ext(\Sigma(\Lambda))$ and is defined as follows:*

$$Ext(\Sigma(\Lambda)) = \{\nu \in \Sigma(\Lambda) \ : \ \deg_R(\nu) \geq \deg_R(p), \forall p \in N(\nu) \cap \Sigma(\Lambda)\},$$

where $N(\nu)$ is the set of divisors formed by adding or removing a single chip from ν.

Remark 3.11. *Interpreted in the language of the chip-firing games, an extreme divisor is a divisor which is not equivalent to any effective divisor, but adding even one chip to any vertex yields a divisor which is.*

Definition 3.12. *We say Λ is uniform if all extreme divisors have the same degree.*

Asadi and Backman also define the sets $Ext(\overline{\Sigma}_{\mathbb{R}}(\Lambda))$ and $Crit(\Lambda)$. We will not include the formal definitions here, however it is shown that the elements of $Ext(\overline{\Sigma}_{\mathbb{R}}(\Lambda))$ are exactly the extreme divisors shifted by $\vec{\mathbf{1}}$, or more precisely

$$Ext(\overline{\Sigma}_{\mathbb{R}}(\Lambda)) = \{E + \vec{\mathbf{1}} \ : \ E \in Ext(\Sigma(\Lambda))\}.$$

Also $Crit(\Lambda)$, the set of critical points of Λ, is obtained by projecting each element of $Ext(\overline{\Sigma}_{\mathbb{R}}(\Lambda))$ onto H_R orthogonally. We will not use these constructions except to define the following.

Definition 3.13. *The lattice* Λ *is reflection invariant if for some* $c \in \mathbb{R}^{n+1}$*,*

$$-Crit(\Lambda) = Crit(\Lambda) + c.$$

Definition 3.14. *We say that* Λ *has the Riemann–Roch property if there exists a canonical divisor* K*, with* $\deg(K) = 2g - 2$*, such that*

$$r(D) - r(K - D) = \deg(D) + g - 1$$

for all divisors D*.*

Asadi and Backman (2011) then show that this is equivalent to Λ being uniform and reflection invariant, just as it was in Amini and Manjunath (2010).

This is interesting and important in that it establishes a fairly surprising relationship between the rank, which is relatively difficult to compute, and the degree, which is quite easy to get a hold of.

Let $\mathcal{R}$ be the diagonal matrix with entries $\mathcal{R}_{ii} = R(v)$. Then Asadi and Backman (2011) show that

$$\mathcal{R}\Lambda_R \subset \Lambda_{\vec{1}},$$

and that $\mathcal{R}\Lambda$ will be a full rank sublattice of $\Lambda_{\vec{1}}$ that is uniform and reflection invariant if and only if Λ is. This lets them conclude the following.

Theorem 3.15. *For any full rank sublattice* Λ *of* Λ_R*,* $\mathcal{R}\Lambda \subset \Lambda_{\vec{1}}$ *has the Riemann–Roch property if and only if* Λ *does.*

We will make use of this in Section 3.6.6 to reduce testing for the column Riemann–Roch property to testing for the row Riemann–Roch property of a related graph.

3.5 Reduced Divisors

Before presenting an algorithm for determining if a graph has the Riemann–Roch property, we introduce reduced divisors, which will be very useful for tying the row chip firing game to the geometry of the associated lattice.

Definition 3.16. *A firing strategy* f *is called valid with respect to some vertex* v^* *if* $f(v^*) = 0$ *and* $0 \leq f(v) \leq \hat{f}(v)$ *for all* $v \in V(G) \setminus v^*$*.*

Definition 3.17 (Definition 3.3 in Asadi and Backman (2011))**.** *A divisor* D *of a graph* G *is* v^**-reduced if:*

(i) for all $v \in V(G) \setminus v^*, D(v) \geq 0$,

(ii) carrying out any valid firing strategy will bring some vertex other than v^* into debt.

Remark 3.18. *If a divisor D is v^*-reduced, then $D(v)$ is less than the out degree of v for all $v \in V(G) \setminus v^*$. If this were not the case, then $f = \chi_{\{v\}}$ would contradict (ii).*

In essence, we are picking one vertex to be a sink, and we have poured as much as we can into it without any vertex going into debt.

We will need the following lemma, regarding the existence of reduced divisors.

Lemma 3.19 (Lemma 3.8 in Asadi and Backman (2011))**.** *For all vertices v and any divisor D, there are exactly $\hat{f}(v)$ v-reduced divisors equivalent to D.*

3.6 Testing for the Riemann–Roch Property

We will now go over how to test for the Row Riemann–Roch Property. We will later see how we can test for the Column Riemann–Roch Property by performing a simple transformation on your graph and checking if that one has the Row Riemann–Roch Property (described in 3.6.6).

This algorithm draws upon the row chip firing game to locate the extreme divisors of the lattice, up to equivalence, from which uniformity and reflection invariance can be tested directly.

3.6.1 The Algorithm

- Calculate $\hat{f}$
- To locate the extreme divisors:
  - Fix some vertex v^* and examine all divisors that give -1 chips to v^* give all other vertices a nonnegative number of chips no larger than their outdegree
  - For each such divisor D, use Dhar's algorithm to determine the set of v^*-reduced divisors equivalent to D. From this, we find out whether or not D is equivalent to an effective divisor, and whether or not it is itself reduced with respect to v^*.

  - Collect those divisors that are not equivalent to effective divisors, are v^*-reduced, and for which adding even one chip to any vertex yields a divisor equivalent to an effective. This finds all extreme divisors, up to equivalence.
- The lattice is uniform if all extreme divisors have the same number of chips.
- If it is, test for reflection invariance:
  - Fix one of your extreme divisors, E^*.
  - For each of the extreme divisors, E, in your list:
    - Define $c_E = E^* + E$.
    - Test each extreme divisor, E', in the list, to see if $-E' + c_E$ is itself extreme.
  - If, for any E, this holds for all E', then the lattice is reflection invariant, and if no such E exists, then it is not.
- If you find the lattice to be both uniform and reflection invariant, then the graph has the row Riemann–Roch property.

3.6.2 Locating Extreme Divisors

The task of locating the extreme divisors is based on the characterization given by the following lemma.

Lemma 3.20 (Lemma 3.10 of Asadi and Backman (2011))**.** *Let G be a directed graph and let D be a divisor. Then*

(i) D is equivalent to an effective divisor if and only if there exists a v^*-reduced divisor $D' \sim D$ such that D' is effective;

(ii) Suppose D is not equivalent to an effective divisor. Then D is an extreme divisor if and only if for any $v \in V(G)$, there exists a v-reduced divisor $D' \sim D$ such that $D'(v) = -1$.

Remark 3.21. *This is basically just saying that if you add one chip anywhere it becomes equivalent to an effective divisor.*

The statement of (ii) reminds us that all divisors equivalent to an extreme divisor must be extreme themselves, so to get a hold of the infinitely many extreme divisors, we really only need to find a representative from each equivalence class of extreme divisors. To get started on that path, notice that (ii) implies that, in particular, every equivalence class of extreme divisors contains a v^*-reduced divisor with -1 chips at v^* for any choice of v^*.

We now fix any vertex v^* to use throughout the rest of the algorithm. The remark after Definition 3.17 asserts that for any v^*-reduced divisor, all vertices other than v^* must have a nonnegative number of chips less than their out degree. In our search for the extreme v^*-reduced divisors with -1 chips at v^*, this observation allows us to restrict our attention to a finite number of candidates.

Before proceeding, we take a moment to discuss the choice of which vertex to select as v^*. In some sense it doesn't matter, because you will come to the same conclusions either way, but there may very well be computational advantages to one over the other. This is not an avenue that I explored rigorously, but my advice would be to choose a vertex with maximum out degree. This is because the number of candidates for extreme v^*-reduced divisors discussed above is the product of the out degrees of the other vertices. A naive implementation would look at each of these candidates and would clearly benefit from this choice, although we will shortly describe ways to cut down this number, so it is less clear how important this choice will be.

We now discuss the generalized Dhar's algorithm, a tool that will be useful in identifying exactly which of the candidate divisors are extreme.

3.6.3 Reducing

Asadi and Backman generalized an algorithm found by Dhar , which can be used to identify whether or not a divisor is v-reduced. If it is, the algorithm will also find all equivalent v-reduced divisors, and if it is not, the algorithm will terminate on a valid firing strategy that will move it closer to being reduced.

The generalized Dhar's algorithm begins with any divisor D that is nonnegative on all vertices except possibly v^* and produces a decreasing sequence of firing strategies as follows:

$$f_0 = \hat{f}$$

and for $t > 0$, pick a vertex v such that $(D - Q^{\mathrm{T}} f_{t-1})(v) < 0$, if one exists, and set

$$f_t = f_{t-1} - \chi_{\{v\}}.$$

If no such v exists, then set

$$f_t = f_{t-1} - \chi_{\{v^*\}}$$

if $f_{t-1}(v^*) > 0$, and terminate otherwise, returning the sequence up to this point.

Asadi and Backman (2011) then prove the following theorem.

Theorem 3.22. *Let D be a divisor satisfying condition (i) in Definition 3.17. Then:*

(i) the divisor D is v^* reduced if and only if the generalized Dhar's Algorithm terminates at $f_{\vec{\mathbf{1}}\cdot\hat{f}} = 0$.

(ii) if D is a v^*-reduced divisor then for each $0 \leq t < \vec{\mathbf{1}} \cdot \hat{f}$ such that $f_{t+1} = f_t - \chi_{\{v^*\}}$, $D - Q^T f_t$ is a v^*-reduced divisor.

If the algorithm terminates on step t^* with firing strategy f_{t^*}, then it must be that carrying out f_{t^*} on D will not put any vertices other than possibly v^* into debt and must have $f_{v^*}(v^*) = 0$. Furthermore, $f_{v^*}(v) \geq 0$ for all $v \in V(G) \setminus v^*$. Otherwise, some vertex would be the first one to start borrowing, but no vertex that starts out out of debt will go into debt without firing and without any of its neighbors borrowing.

Thus, if this algorithm terminates before reaching $f_{\vec{\mathbf{1}}\cdot\hat{f}}$, then D is not reduced, and f^* is a valid firing strategy that will bring D closer to being reduced. We can apply this firing strategy to D and rerun the algorithm, repeating until we find the reduced divisors that are equivalent to D.

3.6.4 Locating Extreme Divors, cont.

We now have all of the tools we need. Of the candidates for v^*-reduced divisors described previously, the ones that we will pull out as representatives of the extreme divisors are those that are v^*-reduced, are not equivalent to effective divisors, and for which the divisors obtained by adding one chip to any vertex are all equivalent to effective divisors.

One way to carry out the computations is as follows. First, create a table that has an entry for each divisor D that has $D(v^*) = -1$ and has $D(v)$ nonnegative and less than or equal to the out degree of v for all other vertices v. (Note that we have included divisors with as many chips as the out degree of the vertex even though these are not going to be reduced. This is because it allows us to prove, for any divisor that could be reduced, that adding a single chip does in fact yield a divisor equivalent to an effective.) In this table, we will keep a running classification of these divisors as either extreme, ruled out, or untested (the default value).

We now look at each divisor of this form, in a non increasing order of total number of chips. For a given divisor, D:

- If we have already ruled the divisor out, then pass it by.
- If this divisor has already been marked as extreme or if running the generalized Dhar's algorithm finds it to be reduced and not equivalent to any effective divisors, then mark it as extreme and rule out all divisors that are reachable by removing chips. Note that this one must be extreme, because we have looked at all divisors with one more chip and if any of them were found to be extreme, this would have already been ruled out.
- Otherwise, we can rule out this divisor, but we should continue to use the generalized Dhar's algorithm as described in the preceding section to determine if it is equivalent to any effective v^*-reduced divisors. If not, then we can rule out all It will be extreme, but we don't care. It will be equivalent to a reduced divisor that also has -1 chips at v^*, so mark that one as extreme.

3.6.5 Testing for Uniformity and Reflection Invariance

Testing for uniformity is quite easy. The lattice will be uniform if the extreme divisors located in the previous step all have the same total number of chips, which is to say they all have the same degree. Recall that every extreme divisor is equivalent to one of these, and that equivalent divisors will have the same number of chips (because firing in the row chip firing game preserves this). Thus, checking only these extreme divisors is sufficient.

Testing for reflection invariance is a little bit more involved than for uniformity. Recall that a lattice is reflection invariant if negating the set of critical points only translates the set, or more precisely $-Crit(\Lambda) = Crit(\Lambda) + c$. To apply this definition directly, we would need to find these critical points by first adding $\vec{\mathbf{1}}$ to each extreme divisor, then projecting onto $H_{\vec{\mathbf{1}}}$, the real hyperplane in which Λ lives.

If we already know the graph to be uniform, we can simplify this somewhat. First of all, adding $\vec{\mathbf{1}}$ to a divisor does not affect where it goes under projection onto $H_{\vec{\mathbf{1}}}$, so we can ignore this step. In particular, projecting an extreme divisor E yields $E - \frac{\deg(E)}{n+1}\vec{\mathbf{1}}$. This leads to the second observation:

Lemma 3.23. *For uniform* Λ*:*

(i) Λ *is reflection invariant if and only if there exists* $C \in \mathbb{Z}^{n+1}$ *such that for all extreme divisors* E*,* $C - E$ *is also extreme.*

(ii) *If* C *satisfies (i), than any* $C' \sim C$ *will as well.*

Proof. If all extreme divisors have the same degree, then for any such divisor, projection onto $H_{\vec{1}}$ corresponds to translation by the fixed vector $t = -\frac{\deg(E)}{n+1}\vec{\mathbf{1}}$.

Thus, we will have reflection invariance if and only if there exists some c such that for every extreme divisor E, there exists an extreme divisor E' such that

$$-(E - t) = (E' - t) + c,$$

or equivalently

$$-E = E' + (c - 2t).$$

□

Being able to test for reflection invariance without ever looking at critical points notably allows us to work exclusively with integers, sidestepping the risk of rounding errors should we ask a computer to carry out this algorithm.

We now fix an extreme divisor E^*. This lemma guarantees that, if Λ is uniform and reflection invariant, we can map it to one of the extreme divisors found previously. It is thus sufficient to check, for each extreme divisor E located above, if $C = E^* + E$ satisfies Lemma 3.23(i). If any do, then Λ is reflection invariant, otherwise it is not.

To determine, for a fixed E^* and E, if C satisfies the condition in Lemma 3.23(i), we need only confirm it for the set of extreme divisors found above (or any set of representatives from each equivalence class of extreme divisors). Note that if take an extreme divisor E' from our list and compute $C - E'$, the result will not necessarily appear in our list even if it is extreme. However, we can use the generalized Dhar's algorithm in the same way as before to find the set of equivalent reduced divisors. We already know that $C - E'$ will be extreme if and only if it is equivalent to one of the reduced extreme divisors on our list.

Note that if we find $C - E'$ to be equivalent to an extreme divisor on our list, we do not need to separately check that divisor, because if $C - E' \sim E''$, then $C - E'' \sim E'$. We also know, by construction, that E^* and E will work without testing them explicitly.

3.6.6 Testing for the Column Riemann–Roch Property

The following theorem reduces testing for the column Riemann–Roch property to testing for the row Riemann–Roch property in a related graph, at which point you can use the above algorithm.

Theorem 3.24 (Theorem 3.18 in Asadi and Backman (2011))**.** *Let G be a strongly connected directed graph with Laplacian Q and let G' be the Eulerian directed graph with Laplacian $Q^T\mathcal{R}$ where $\mathcal{R}$ is as defined in Section 3.4. Then G has the column Riemann–Roch property if and only if G' has the row Riemann–Roch property.*

The proof of this theorem follows from Theorem 3.15.

Chapter 4

Future Work

An immediate next step in this project is to implement this algorithm and begin building up a database of small graphs and whether or not they have each of these Riemann–Roch properties.

Some things to keep in mind when writing this program:

- There may be a more clever way to rule out candidate divisors while searching for the extreme divisors. For example, whenever we find the v^*-reduced divisors equivalent to some candidate D, we can take each of them, add an appropriate number of chips to v^* to find one of our candidates that we know will be v^*-reduced. The one that started out with the most chips at v^* will yield a candidate that is not equivalent to an effective divisor, and all the others will yield candidates that are. This allows us to rule out divisors that can be formed by removing chips in any configuration and those formed by adding chips in any configuration, respectively. It is quite possible that you can take advantage of this information to better optimize the step of finding extreme divisors.

- Many aspects of this algorithm can be parallelized easily. For example, you could simultaneously test multiple graphs at once, or multiple different choices for E while checking for reflection invariance without

- If you are trying to build up a database of small graphs, take care to enumerate the graphs so that you hit everything (so that running forever would eventually hit graphs with arbitrarily many vertices, as well as arbitrarily many edges on any number of vertices), and skips over as many isomorphic graphs as possible.

Other good directions would include attempting to characterize those directed graphs that have each of the Riemann–Roch properties, as well as finding the density of graphs that have each of these properties.

Bibliography

Amini, O., and M. Manjunath. 2010. Riemann–Roch for sub–lattices of the root lattice a_n .

Asadi, A., and S. Backman. 2011. Chip–firing and Riemann–Roch theory for directed graphs .

Baker, M., and S. Norine. 2007. Riemann–Roch and Abel–Jacobi theory on finite graphs .

Gathmann, A., and M. Kerber. 2007. A Riemann–Roch theorem in tropical geometry .