



Universidad de San Carlos de Guatemala
Escuela de Ciencias Físicas y Matemáticas
Departamento de Matemática

JUEGOS DE DISPARO DE FICHAS Y LA FÓRMULA DE RIEMANN-ROCH

Alejandro José Vargas De León

Asesorado por Lic. William Roberto Gutiérrez Herrera

Guatemala, Octubre de 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



ESCUELA DE CIENCIAS FÍSICAS Y MATEMÁTICAS

**JUEGOS DE DISPARO DE FICHAS Y LA
FÓRMULA DE RIEMANN-ROCH**

TRABAJO DE GRADUACIÓN
PRESENTADO A LA JEFATURA DEL
DEPARTAMENTO DE MATEMÁTICA
POR

ALEJANDRO JOSÉ VARGAS DE LEÓN
ASESORADO POR LIC. WILLIAM ROBERTO GUTIÉRREZ HERRERA

AL CONFERÍRSELE EL TÍTULO DE
LICENCIADO EN MATEMÁTICA APLICADA

GUATEMALA, OCTUBRE DE 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
ESCUELA DE CIENCIAS FÍSICAS Y MATEMÁTICAS

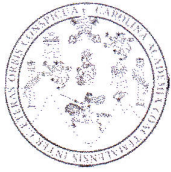


CONSEJO DIRECTIVO

DIRECTOR M.Sc. Edgar Anibal Cifuentes Anléu
SECRETARIO ACADÉMICO Ing. José Rodolfo Samayoa Dardón

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

EXAMINADOR Dra. Mayra Virginia Castillo Montes
EXAMINADOR Lic. Carlos Augusto Morales Santacruz
EXAMINADOR Lic. Francisco Bernardo Raúl De La Rosa



Universidad de San Carlos de Guatemala
Escuela de Ciencias Físicas y Matemáticas



Ref. D.DTG. 006-2016

Guatemala 24 de octubre de 2016

El Director de la Escuela de Ciencias Físicas y Matemáticas de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Coordinador de la Licenciatura en Matemática Aplicada, autoriza la impresión del trabajo de graduación Titulado: **JUEGOS DE DISPARO DE FICHAS Y LA FÓRMULA DE RIEMANN-ROCH** presentado por el estudiante universitario **Alejandro José Vargas de León**.

IMPRIMASE.

MsC. Edgar Aníbal Cifuentes Anleu
Director
Escuela de Ciencias Físicas y Matemáticas



/pec

DEDICATORIA

Ella es una nube que el viento conquistó

Los Jaivas

Así que caminé lentamente por las calles empapadas y ventosas, por encima y en el interior de la perceptible furia de la primavera postergada, viendo golpear en el barro las últimas hojas de los árboles, sintiendo las volteretas, casi visibles, del viento que le tocaba la cara.

Juan Carlos Onetti

A mi madre,

por darme la luz de la vida.

A mi padre,

por darme muchas historias, más de las que mi memoria alcanza a guardar.

A mi abuelo,

por darme el primer problema de matemática que resolví.

A mis amigos,

por darme felicidades que espero poder retribuir.

A quienes amo,

por existir.

AGRADECIMIENTOS

Agradezco a mis maestros, a mi familia, a mis amigos, que muchas veces se han fundido para ser la misma cosa. Todos necesitamos héroes que nos hablen, que nos enseñen y nos alienten. Sin ellos éstas páginas no existirían, sin ellos mi identidad actual no existiría. Por ellos es que no soy una persona aislada, por ellos y por toda la gente que me da su apoyo es que soy parte de grandes procesos, que escucho y hablo en la interminable conversación que se sostiene de una generación a otra usando el lenguaje de la matemática. En este trabajo, gracias a ellos, logro decir un par de palabras. Soy un eslabón en una larga cadena de historias, conocimientos y sentimientos que nace en tiempos remotos. Cadena que espero continúe, en parte a través de mí, hasta más allá de dónde mis ojos alcancen a ver.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	III
LISTA DE SÍMBOLOS	V
OBJETIVOS	VII
INTRODUCCIÓN	IX
1. Un juego de disparo de fichas	1
1.1. Definición del juego	1
1.2. Ejemplos	2
1.3. Algunas interrogantes	3
1.4. Divisores	4
1.5. El laplaciano combinatorio	5
2. El algoritmo de Dhar	11
2.1. Divisores reducidos y el algoritmo de Dhar	11
2.2. El algoritmo de Dhar	15
2.3. Orientaciones de grafo	19
3. Riemann-Roch para grafos finitos	27
3.1. El juego de Brill-Noether	27
3.2. La fórmula de Riemann-Roch	31
3.3. Un criterio para Riemann-Roch	34
4. Ejemplos Numéricos	37
CONCLUSIONES	49
RECOMENDACIONES	51

A. Geometría algebraica	53
A.1. El espacio afín y el espacio proyectivo	54
A.2. Funciones racionales	57
A.3. Sistemas lineales de divisores	59
B. Código fuente	63
Bibliografía	79

ÍNDICE DE FIGURAS

1.1. Ejemplo de disparo	2
1.2. Ejemplo de juegos	2
1.3. Divisores equivalentes	6
2.1. Ejemplo de divisor reducido	13
2.2. Grafo copo de nieve	16
2.3. Grafo hombre de jengibre	19
2.4. Orientaciones de Grafo	21
2.5. Reconstruyendo la orientación	24
3.1. Grafo luna creciente	28
3.2. Un árbol	29
3.3. Un ciclo	33
3.4. Cadena de ciclos	33
4.1. Género 6, promedio de número de divisores de rango 1	42
4.2. Género 6, histograma	42
4.3. Género 6, porcentaje	43
4.4. Género 6, tiempo de procesamiento	44
4.5. Género 8, promedio de número de divisores de rango 1	44
4.6. Género 8, histograma	45
4.7. Género 8, rango y tiempo	45
4.8. Grafo inicial para experimento	46
4.9. Evolución rango 1	47
4.10. Grafo pez tropical	48

LISTA DE SÍMBOLOS

Símbolo	Significado
$:=$	es definido por
G	un grafo conexo, finito, sin bucles, posiblemente con aristas múltiples
$\mathcal{M}(G)$	grupo abeliano de las funciones de valores enteros definidas sobre los vértices de un grafo
\mathbb{Z}	conjunto de enteros
\mathbb{A}	espacio afín
\mathbb{P}	espacio proyectivo
$\mathbf{1}$	función constante de valor 1 sobre los vértices de un grafo
\mathcal{N}	conjunto de los divisores no especiales
\mathcal{O}	orientación de grafo
$\text{Hom}(A, B)$	homomorfismos de A a B
$\text{Div}(G)$	grupo de divisores de G
$\text{div}(f)$	divisor asociado a f
$\text{Prin}(G)$	grupo de divisores principales de G
$\text{Jac}(G)$	el jacobiano de G
$\text{Im}(A)$	imagen de una matriz A
$\text{Div}^d(G)$	divisores de grado d sobre G
$\Delta(f)$	el laplaciano de f
$\Delta_v(f)$	el valor del laplaciano de f en el vértice v
$\text{indeg}_{\mathcal{O}} v$	grado entrante de v en la orientación \mathcal{O}
$\text{outdeg}_{\mathcal{O}} v$	grado saliente de v en la orientación \mathcal{O}
$ D $	sistema lineal asociado al divisor D

OBJETIVOS

General

Hacer una recopilación de resultados respecto a un juego de disparo de fichas e implementar varios algoritmos relacionados a éste.

Específicos

1. Definir un juego de disparo de fichas en grafos conexos y finitos.
2. Describir un algoritmo para decidir si las condiciones iniciales del juego admiten una estrategia ganadora o no.
3. Enunciar y demostrar la fórmula de Riemann-Roch para divisores de un grafo finito y conexo.
4. Implementar los algoritmos descritos en un programa en Perl.

INTRODUCCIÓN

En el transcurso de este trabajo sea G un grafo conexo y finito. Permitimos aristas múltiples, pero no bucles. Desde la década del ochenta surgió interés en estudiar *juegos de disparo de fichas* en G . Hay diversas variantes, pero todas se basan en la idea de distribuir fichas (a veces llamadas granos de arena, o dólares) en los vértices de G y luego definir una *jugada de disparo* como elegir un vértice y hacer que éste mande fichas a sus vecinos, una ficha mediante cada arista adyacente al vértice.

El primer objetivo es estudiar las configuraciones que pueden ser alcanzadas desde una configuración inicial. Analizar el juego está motivado por diversas áreas de estudio, donde el juego surge de manera natural. Puede ser motivado por el estudio de cierto *juego de balanceo* [35]. La generalización es el *juego de disparo de fichas constreñido*, constreñido porque el juego continua mientras los vértices tengan suficientes fichas para disparar [14]. También se puede acercar uno al juego desde la física, en donde surge como un ejemplo de criticidad auto-organizada. Aquí se conoce como el *modelo abeliano de la pila de arena* [18, 33], en donde hay vértices disipadores que nunca disparan, sólo absorben fichas. Hasta ahora consideramos únicamente vértices con cantidades no negativas de fichas. Biggs usa dólares en lugar de fichas, permite que un vértice tenga una cantidad negativa, le llama *el banco*, y dice que el banco puede tener *deuda*. Este es *el juego del dólar* [13].

Siguiendo a Biggs, si un vértice tiene una cantidad negativa de fichas decimos que está en deuda. En la versión de Baker podemos distribuir cualquier número entero de fichas a los vértices, y cualquier vértice puede ser disparado independientemente de si está en deuda o no [10]. Este *juego de disparo de fichas no constreñido*, o sencillamente el juego de disparo de fichas en lo sucesivo, surge por el interés de Baker en geometría p -ádica. En este contexto uno puede asociar un grafo a una curva sobre un campo no arquimediano. La jugada de disparo aparece luego de hacer varias construcciones en la teoría [4, 5, blog, ver la sección de comentarios].

Dado G y una configuración de fichas en G , el juego de disparo de fichas

consiste en sacar a todos los vértices de deuda. Comenzamos la teoría describiendo e ilustrando el juego. Más adelante describiremos un juego relacionado, el juego de Brill-Noether, una variante para dos jugadores.

Las conexiones con curvas algebraicas empiezan al introducir un lenguaje similar. Las configuraciones de fichas pueden ser identificadas con *divisores*, elementos del grupo abeliano libre sobre el conjunto de vértices de G , esto se hace también en el caso de curvas algebraicas. La condición de no negatividad para ganar el juego sugiere fuertemente la noción de *divisores efectivos*, entiéndase divisores cuyos coeficientes son no negativos. No todas las configuraciones definen un juego que puede ser ganado. Caracterizamos las configuraciones mediante un procedimiento que da una estrategia ganadora, o garantiza que ninguna existe.

Usamos la jugada de disparo para inducir una relación de equivalencia. Dos configuraciones son equivalentes si se pueden hacer jugadas de disparos en la primera para terminar en la segunda. Las configuraciones con estrategia ganadora son aquellas cuya clase de equivalencia tienen al menos un representante efectivo. Definimos la noción de divisor reducido y mostramos que estos divisores se pueden usar como representantes canónicos y que además resuelven la pregunta de si un juego tiene estrategia ganadora o no. Describimos un algoritmo de tiempo polinomial para calcular los divisores reducidos.

Inicialmente nuestros métodos dependerán fuertemente en la estructura topológica de G . Eventualmente lograremos desarrollar algo más general, el teorema principal de este trabajo es la fórmula de Riemann-Roch para grafos. En lugar de lidiar directamente con la estructura topológica de G , tan sólo es necesario conocer un número, el género de G . En la literatura se le conoce a éste número como *circuit rank* en inglés, o *número ciclomático* en español. Aquí le llamamos *género* del grafo por analogía con la teoría clásica. Riemann-Roch para grafos también responde preguntas en la versión extendida del juego, el juego de Brill-Noether. El orden de ideas de esta exposición sigue principalmente al artículo [10].

1. Un juego de disparo de fichas

1.1. Definición del juego

Esta sección define el juego de disparo de fichas que es el principal objeto de estudio de este trabajo. Denotamos con $V(G)$ y $E(G)$ el conjunto de vértices y el conjunto de aristas de G respectivamente. Cuando no hay peligro de confusión escribimos G en lugar de $V(G)$. Sea $n = |V(G)|$, $m = |E(G)|$. Para denotar una arista e incidente con el vértice v escribimos $v \in e$. Con E_v denotamos el conjunto de todas las aristas incidentes a v , es decir, todas las aristas $e \in E$ con $v \in e$. Escribimos $\deg v = |E_v|$, el grado de v .

El *juego de disparo de fichas no constreñido* comienza con una configuración inicial D de fichas distribuidas en los vértices de G . Denotamos con $D(v)$ al número de fichas en el vértice v . Este número puede ser cualquier entero. Se dice que un vértice con un número negativo de fichas está *en deuda*. Una *jugada de disparo* del vértice v consiste en hacer que v mande fichas a sus vecinos, manda una ficha por medio de cada arista adyacente a v . Una secuencia de jugadas produce una *configuración equivalente*.

El juego termina cuando alcanzamos una configuración sin vértices en deuda. A tal configuración la llamamos una *configuración ganadora*. Una secuencia de jugadas comenzando en D y terminando en una configuración ganadora es llamada una *estrategia ganadora*.

1.2. Ejemplos

Ejemplo 1.1. La figura 1.1 muestra dos configuraciones D y D' que son equivalentes. El número escrito en las cajas muestra el número de fichas en cada vértice. El vértice al centro de D ha sido disparado para obtener D' . Se permiten las aristas múltiples.

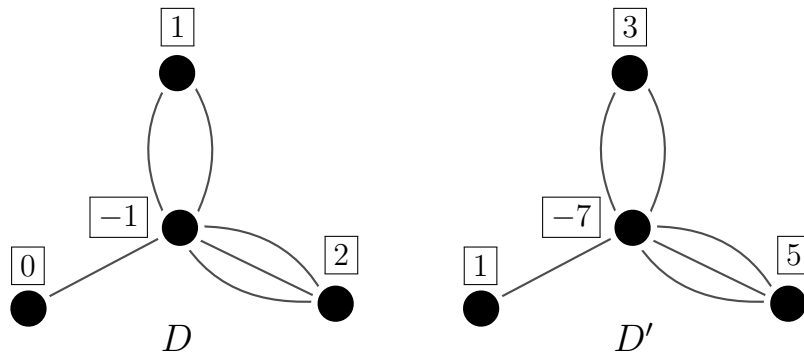


Figura 1.1. Ejemplo de disparo. Elaboración propia.

Ejemplo 1.2. La figura 1.2 muestra dos juegos posibles en el mismo grafo.

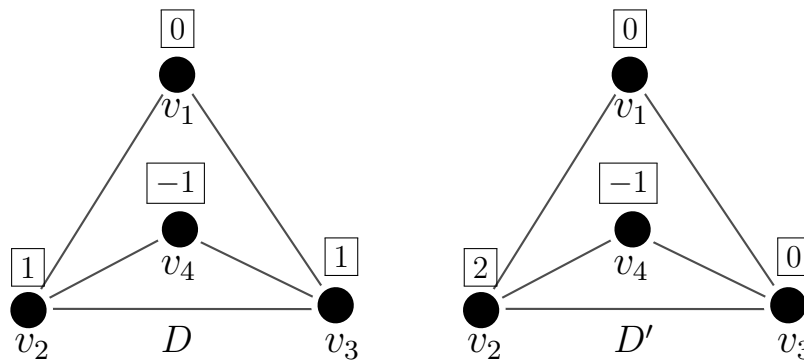


Figura 1.2. Ejemplo de juegos. Elaboración propia.

Disparar v_1, v_2 y v_3 en la configuración D saca a todos los vértices de deuda, así que podemos ganar D . El asunto con D' es más sutil. Sean a, b el número de fichas en v_2, v_3 respectivamente. Se puede demostrar por análisis de casos que disparar cualquier vértice v_i resulta en a', b' fichas en v_1, v_2 , y que estos números cumplen $a - b \equiv a' - b' \pmod{4}$. Uno también observa que cualquier configuración ganadora equivalente a D' tiene una ficha en total, así que $a - b \equiv -1, 0$ o $1 \pmod{4}$, pero D' tiene $a - b \equiv 2 \pmod{4}$. D' no se puede ganar. \square

1.3. Algunas interrogantes

Se puede ganar el juego que comienza en D si existe por lo menos una estrategia ganadora. ¿Acaso es posible ganar todos los juegos? Sea d el número total de fichas en juego, es decir, $d = \sum_{v \in G} D(v)$. Observamos d es invariante bajo la jugada de disparo. Por lo tanto una condición necesaria para que D tenga estrategia ganadora es que $d \geq 0$. ¿Es ésta condición también suficiente?

El ejemplo 1.2 muestra que no es suficiente tener $d \geq 0$, y que además este número por si solo no puede caracterizar las configuraciones que se pueden ganar, ya que tanto D y D' tienen el mismo d . El número de fichas en los ejemplos es relativamente bajo. Tal vez para valores altos d ofrece un criterio para decidir si se puede ganar el juego. Nos preguntamos lo siguiente:

Pregunta 1.3. Para un grafo G , ¿existe una constante c , tal que si una configuración de fichas D tiene al menos c fichas entonces podemos asegurar que existe estrategia ganadora para D ?

Hablando de manera intuitiva, esperamos que las ocurrencias como el ejemplo 1.2 requieran un número *relativamente bajo* de fichas. También esperamos que si d es alto, es decir hay muchas fichas en juego, entonces es posible ganar D sin importar cómo hemos distribuido las fichas. ¿Qué tan grande es este número? El lector puede desarrollar los detalles de la siguiente idea: por el principio de las casillas si se cumple $d \geq 2m - n + 1$ entonces siempre hay un vértices que se puede disparar sin incurrir en deuda. Entonces uno puede hacer una secuencia de disparos arbitrariamente larga, escogiendo siempre vértices que no incurren en deuda. Eventualmente todos los vértices aparecen en esta secuencia, por lo que tenemos una estrategia ganadora. Esencialmente en este caso el juego constreñido nunca para y respondemos de manera positiva la pregunta 1.3. ¿Podemos elaborar más respecto la pregunta 1.3? Sí, ahora que sabemos que c existe, nos preguntamos si tenemos el mejor c posible:

Pregunta 1.4. Para un grafo G , ¿cuál es la constante c más pequeña tal que si D tiene al menos c fichas entonces D se puede ganar?

En el ejemplo 1.2 vemos que $c > 1$ dado que hemos mostrado una configuración con una ficha y sin estrategia ganadora. ¿Qué tal 2? El lector se puede divertir demostrando que 2 cumple. Cualquier configuración con 2 fichas se puede ganar, así que $c = 2$. El mismo ejercicio se puede hacer para el ejemplo 1.1. Primero, D no se puede ganar. Segundo, cualquier configuración con tres fichas se puede ganar, así que $c = 3$.

La constante c es un invariante del grafo, se preserva bajo isomorfismos. Nos

preguntamos si coincide con alguna de las invariantes de grafos ya conocidas. Se requieren muchos ejemplos para ver que c coincide con $g = m - n + 1$, el *rango de circuito* o *número ciclomático*. En el contexto de Riemann-Roch para grafos por analogía llamamos a este número el *género* de G . En particular para los dos ejemplos la constante c coincide con el género.

Observación 1.5. En la literatura hay otra definición para el género de G , y este número usualmente no coincide con nuestra definición de $g = m - n + 1$. En este trabajo cuando decimos «género» nos referimos a g .

¿Es razonable esperar que $c = g$ en general? Este hecho apuntaría a una conexión entre el juego y las propiedades topológicas del grafo. También ahora nuestra cota es $c \leq 2m - n + 1$. Ya que $g = m - n + 1$, demostrar $c \leq g$ sería una gran mejora. Además nos preguntamos, ¿es posible encontrar configuraciones con $g - 1$ fichas y sin estrategias ganadoras? Estas dos preguntas guiarán la investigación inicial del capítulo:

Pregunta 1.6. Dado un grafo G y una configuración D con al menos g fichas, ¿existe siempre una estrategia ganadora para D ?

Pregunta 1.7. Dado un grafo G , ¿existe una configuración D con $g - 1$ fichas y sin estrategia ganadora?

1.4. Divisores

En esta sección definimos un lenguaje que tiene el mismo espíritu que el utilizado en la teoría clásica de divisores. Vemos que el juego de disparo de fichas puede ser representado de una manera natural utilizando este lenguaje, a tal punto que nuestro estudio del juego desde este punto en adelante se realizará estudiando divisores, estos son combinaciones lineales enteras formales sobre el conjunto de vértices $V(G)$. Éste punto de vista es muy fructífero, nos lleva a la fórmula de Riemann-Roch para grafos conexos y finitos. Para los lectores que no están familiarizados con geometría algebraica hemos incluido el apéndice I que sirve como un resumen de la teoría clásica.

Definimos el grupo de divisores $\text{Div}(G)$ del grafo G . El grupo $\text{Div}(G)$ contiene todas las combinaciones lineales formales de los elementos del conjunto $V(G)$ con coeficientes enteros. La operación es la suma formal. Los elementos de $\text{Div}(G)$ son llamados *divisores* en G . A una configuración inicial del juego le asociamos un divisor de la siguiente forma:

$$D = \sum_{v \in G} D(v) (v).$$

Por ejemplo, la configuración de fichas del ejemplo 1.2 se identifica con $D = v_2 - v_3 - v_4$, y si nos referimos al coeficiente en v_1 continuaremos usando $D(v_1) = 0$. Escribimos $E \geq D$ si $E(v) \geq D(v)$ para todo $v \in G$, y decimos que E *domina* a D . Si $E \geq 0$ entonces decimos que E es *efectivo*. Una configuración ganadora se identifica con un divisor efectivo. El soporte de un divisor, denotado por $\text{supp } D$, es el conjunto de vértices v donde $D(v) \neq 0$. El *grado* de un divisor D es su número total de fichas:

$$\deg D = \sum_{v \in G} D(v).$$

1.5. El laplaciano combinatorio

Sea $\mathcal{M}(G)$ el grupo abeliano de funciones valuadas en los enteros definidas en el conjunto de vértices de G , junto con la adición usual de funciones. Para codificar la jugada de disparo definimos el *operador laplaciano* como $\Delta: \mathcal{M}(G) \rightarrow \text{Div}(G)$ con $\Delta(f) := \sum_{v \in G} \Delta_v(f)(v)$, donde:

$$\begin{aligned} \Delta_v(f) &= \deg(v)f(v) - \sum_{e \in E_v, w \in e} f(w) \\ &= \sum_{e \in E_v, w \in e} (f(v) - f(w)). \end{aligned}$$

De la definición observamos que si $a \in \mathbb{Z}$, entonces $\Delta_v(af) = a\Delta_v(f)$. También vemos que para $f, g \in \mathcal{M}(G)$, $\Delta_v(f + g) = \Delta_v(f) + \Delta_v(g)$. Por lo tanto $\Delta(\cdot)$ es un operador lineal sobre los enteros.

Ahora vemos que $\Delta_v(f)$ describe el disparo de un vértice v . Si comenzamos con una configuración D en G , y si disparamos $f(v_i)$ veces el vértice v_i , entonces el divisor resultante E se relaciona por D por medio de la relación:

$$E = D - \Delta(f).$$

¿Qué sucede cuando todos los vértices de un subconjunto no vacío $A \subset G$ son disparados al mismo tiempo? Denotemos con A^c el complemento de A en G . Sea $\chi_A \in \mathcal{M}(G)$ la *función indicador de A* , con valores $\chi_A(v) = 1$ para $v \in A$, y valor cero para los otros casos. También definimos para subconjuntos $A, B \subset G$ el conjunto (A, B) de aristas que tienen un extremo en A y el otro en B . Nos preguntamos, ¿cómo podemos calcular $D - \Delta(\chi_A)$?

Ejemplo 1.8. Sea $D = v_0 + v_2$ en la figura 1.3. Se dispara el conjunto $A =$

$\{v_7, v_8, v_0, v_1, v_2, v_9, v_{10}, v_{11}\}$ para obtener $D' = D + \Delta(\chi_A) = v_3 + v_4 + v_5 + v_6 - v_7 - v_8$. Notamos que solamente las fichas que están en el conjunto de aristas (A, A^c) han sido modificadas.

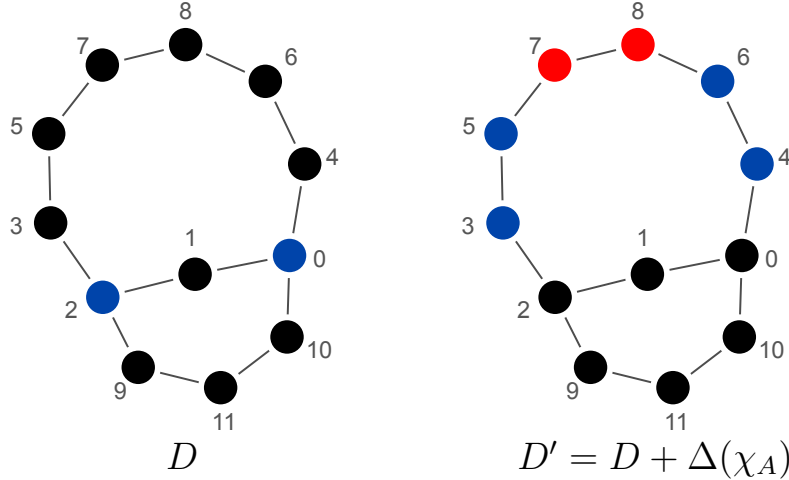


Figura 1.3. Divisores equivalentes. Elaboración propia.

Para $A \subset G$ el conjunto (A, A^c) es el *corte* definido por A . Para $v \in A$ definimos $\text{outdeg}_A v$ como el número de aristas en E_v incidentes a algún vértice en el conjunto A^c . Este número puede ser cero. Como sugiere el ejemplo, en la dinámica del juego al disparar un conjunto A sucede que cada vértice v en A pierde $\text{outdeg}_A v$ fichas. Este disparo simultáneo de todos los vértices del conjunto no vacío A es muy útil para jugar el juego, y para el desarrollo posterior de algoritmos. Escribimos χ_v en lugar de $\chi_{\{v\}}$ y resumimos nuestras observaciones:

Lema 1.9. *Disparar un subconjunto no vacío $A \subset G$ envía fichas desde A hacia A^c , una a través de cada arista en el corte (A, A^c) . Los vértices en $v \in A$ pierden $\text{outdeg}_A v$ fichas, y los vértices en $w \in A^c$ ganan $\text{outdeg}_{A^c} w$ fichas.*

Corolario 1.10. Escribimos $\mathbf{1} = \chi_G$. Se cumple que $\mathbf{1} \in \ker \Delta$.

Del corolario 1.10 y de la linealidad de Δ vemos que si $E = D - \Delta(f)$, entonces $E = D - \Delta(f + a\mathbf{1})$ con $a \in \mathbb{Z}$. En particular esto implica que si $E = D - \Delta(f)$, podemos asumir que $f(v) \geq 0$, y que hay igualdad para por lo menos un vértice. Así mismo podríamos asumir $f(v) \leq 0$, nuevamente con igualdad para un vértice. Este hecho es útil para varias demostraciones.

Por lo tanto no es útil hablar del máximo valor de f . En su lugar nos interesan los vértices donde los extremos son alcanzados. Para $f \in \mathcal{M}(G)$ sea $\text{máx } f = \{v \in G : f(v) \geq f(w) \text{ para todo } w \in G\}$. Definimos $\text{mín } f$ de manera similar. Un lema útil con fácil demostración nos dice:

Lema 1.11. *Sea $A = \text{máx } f$, entonces para $v \in A$ tenemos que $\Delta_v(f) \geq \text{outdeg}_A v$.*

Finalmente en [10] la jugada de disparo dada por $\Delta(\chi_v)$ se llama un *movimiento de crédito* por v , y la jugada de disparo dada por $\Delta(\chi_{G \setminus \{v\}})$ se llama un *movimiento de préstamo* por v . La jugada de crédito es simplemente el disparo de un vértice, y la terminología se explica por « v le está dando crédito a sus vecinos». A la luz del lema 1.9, vemos que la jugada de préstamo tiene sentido también, « v le está pidiendo prestado fichas a sus vecinos». En el ejemplo 1.2 ganamos el juego D al hacer que v_4 realizara una jugada de préstamo.

Con esta terminología podemos interpretar una función $f \in \mathcal{M}(G)$ y su laplaciano $\Delta(f)$ diciendo que la cantidad $f(v_i)$ es el número de jugadas de crédito que v_i realiza si $f(v_i) > 0$, o el número de jugadas de préstamo que v_i realiza si $f(v_i) < 0$, por lo que vemos que el laplaciano codifica la información del juego. Inspirados por el juego de disparo de fichas, decimos que los divisores D, D' son linealmente equivalentes, escribimos $D \sim D'$, si existe $f \in \mathcal{M}(G)$ con $D' = D - \Delta(f)$. Tenemos entonces:

Lema 1.12 (Lema 4.3 en [10]). *Dos divisores D y D' en G son linealmente equivalentes sí y sólo sí existe una secuencia de jugadas en el juego de disparo de fichas que transforme la configuración correspondiente a D en una configuración correspondiente a D' .*

Escribimos $[D]$ para la clase de equivalencia de D , es decir, $[D] = \{D' \in \text{Div}(G) : D' \sim D\}$. La clase $[D]$ contiene todas las posibles configuraciones que pueden ser alcanzadas desde D , así que para estudiar el juego estudiamos el conjunto $[D]$. Además la linealidad de $\Delta(\cdot)$ implica que la equivalencia es lineal, si $D \sim D'$ y $E \sim E'$ entonces $D+E \sim D'+E'$. Sea $\text{Prin}(G) = \Delta(\mathcal{M}(G))$, es decir, divisores de la forma $\Delta(f)$ para algún $f \in \mathcal{M}(G)$. Un conteo elemental demuestra que $\text{deg } \Delta(f) = 0$, lo cuál corresponde al ya mencionado hecho de que la jugada de disparo preserva el número de fichas en juego, así que el grado está bien definido para una clase de equivalencia $[D]$.

Las clases de equivalencia $[D]$ viven en el grupo $\text{Div}(G)/\text{Prin}(G)$. Dado que los elementos $\text{Prin}(G)$ son de grado 0, es una idea útil clasificar a los divisores por su grado. Sea $\text{Div}^d(G) = \{D \in \text{Div}(G) \mid \text{deg } D = d\}$. Como los divisores principales son de grado cero, el grado de una clase de equivalencia está bien definido, y el cociente $\text{Div}(G)/\text{Prin}(G)$ es un grupo graduado por el grado inducido desde $\text{Div}(G)$.

El subconjunto de $\text{Div}(G)/\text{Prin}(G)$ de elementos de grado cero tiene una estructura natural de grupo, está dada por la suma de divisores. Llamamos a este grupo el *jacobiano* del grafo. La elección de este nombre se explica por varios resul-

tados de álgebra de homología y cohomología del artículo [2]. El lector interesado puede consultar la bibliografía sobre el jacobiano de un grafo y encontrar que hay varios resultados muy hermosos que se demuestran con la ayuda del juego de disparo de fichas. El jacobiano es el cociente:

$$\text{Jac}(G) = \frac{\text{Div}^0(G)}{\text{Prin}(G)}.$$

Por la linealidad del laplaciano asociamos una matriz Q al operador $\Delta(\cdot)$. Fijamos una enumeración v_1, v_2, \dots, v_n de los vértices de G . El grupo abeliano $\mathcal{M}(G)$ y el grupo de divisores $\text{Div}(G)$ pueden ser identificados con el conjunto \mathbb{Z}^n con la estructura de un \mathbb{Z} -módulo. También por linealidad el operador $\Delta(\cdot)$ está determinado por $\Delta(\chi_{v_1}), \Delta(\chi_{v_2}), \dots, \Delta(\chi_{v_n})$. Definimos $Q = (q_{ij})$, una matriz de n por n , al decir que q_{ij} es la coordenada número i del vector $\Delta(\chi_{v_j})$.

Para un subconjunto $A \subset \mathbb{Z}^n$ escribimos $QA := \{Qa : a \in A\}$. También denotamos con $\text{Im}(Q) := Q\mathbb{Z}^n$, el conjunto \mathbb{Z} -generado de los vectores columna de Q . Uno puede considerar $\text{Im}(Q)$ como un retículo. Vemos entonces que $[\Delta(f)] = Q[f]$, lo cuál significa:

$$\frac{\text{Div}(G)}{\text{Prin}(G)} \cong \frac{\mathbb{Z}^n}{\text{Im}(Q)}.$$

Esta representación sugiere aproximarse al jacobiano utilizando métodos tales como la forma normal de Smith, y así proceder a estudiar sus propiedades aritméticas [28, 29, 27]. Reproducimos un resultado de dicha teoría. La forma normal de Smith consiste en encontrar una matriz diagonal tal que Q is equivalente por filas y columnas sobre los enteros a la matriz $\text{diag}(s_1, s_2, \dots, s_{n-1}, 0)$. Los enteros s_i son llamados *factores invariantes*. Esto implica:

$$\begin{aligned} \frac{\text{Div}(G)}{\text{Prin}(G)} &\cong \frac{\mathbb{Z}^n}{\text{Im}(Q)} \cong \prod_{i=1}^{n-1} \frac{\mathbb{Z}}{s_i \mathbb{Z}} \times \mathbb{Z}, \\ \text{Jac}(G) &\cong \prod_{i=1}^{n-1} \frac{\mathbb{Z}}{s_i \mathbb{Z}}. \end{aligned}$$

Este resultado es maravilloso porque nos dice que para estudiar $\text{Div}(G)/\text{Prin}(G)$ sólo es necesario trabajar con el grupo $\text{Jac}(G)$. También vemos que $\text{Jac}(G)$ es de orden finito y generado por a lo sumo $n - 1$ elementos. Es realmente notable que después de fijar el grafo G y fijar el número de fichas disponibles obtenemos que el número de juegos distintos es finito. Cada juego tiene infinitas configuraciones, pero el número de juegos con d fichas en total es finito.

Para cerrar esta sección mencionamos un par de palabras acerca del *laplaciano combinatorio* [15, página 54]. Sea $D = (d_{ij})$ la matriz diagonal con $d_{ii} = \deg v_i$. Sea $A = (a_{ij})$ la matriz de adyacencia de G definida como la entrada a_{ij} es el número de aristas entre los vértices v_i y v_j . Nótese que no permitimos bucles en G , por lo tanto A tiene ceros en la diagonal. Uno puede ver que $Q = D - A$. A esta matriz le llamamos el *laplaciano combinatorio*, o también la *matriz de Kirchhoff*.

Kirchhoff estaba interesado en circuitos eléctricos. Él propuso un modelo para el comportamiento de la corriente y el voltaje en un circuito basado en dos leyes. El modelo resuelve el problema utilizando la matriz Q , y de esta forma Kirchhoff fundó el campo de la teoría algebraica de grafos. Entre las propiedades y hechos importantes de la matriz Q , queremos destacar el *teorema de Kirchhoff*. Un *árbol* es un grafo de género cero. Un *árbol generador* de G es un subgrafo T con $V(G) = V(T)$. Sea κ el número de árboles generadores de G . Sea $Q_{(i)}$ la matriz que se obtiene al borrar la fila número i y la columna número i de Q , entonces:

Teorema 1.13 (teorema de Kirchhoff). $\det Q_{(i)} = \kappa$.

Demostración. Teorema 12, capítulo 2 de [15] □

Ahora el teorema de Kirchhoff y algunos hechos de teoría de retículos implica que el orden del jacobiano es el número de árboles generadores, es decir, $\#\text{Jac}(G) = \kappa$. En la siguiente sección exploramos el concepto de divisores reducidos. Demostramos que son representantes canónicos para los elementos de $\text{Jac}(G)$. Uno también puede mostrar una biyección explícita entre los divisores reducidos y los árboles generadores, de donde se sigue otra demostración del teorema de Kirchhoff, esta demostración usa el punto de vista de disparo de fichas. [11]

2. El algoritmo de Dhar

2.1. Divisores reducidos y el algoritmo de Dhar

Dedicamos esta sección a encontrar representantes canónicos para las clases de equivalencia $[D]$ de $\text{Jac}(G)$. Para responder este problema elegimos un vértice base $q \in G$ y buscamos elementos de $[D]$ que son efectivos afuera de q y con sus fichas «lo más cerca posible de q ». Estos representantes canónicos pueden ser computados de manera eficiente, en tiempo polinomial respecto a m y n , mediante el algoritmo de Dhar. El representante canónico tiene dos usos, el primero es resolver la pregunta de determinar si dos divisores son equivalentes o no, y la segunda es determinar si el juego asociado al divisor tiene estrategia ganadora o no.

En el ejemplo 1.1 describimos un divisor particular que no tenía divisores equivalentes que fueran efectivos. El método se fundamentaba en observaciones ingeniosas respecto a la dinámica del juego de disparo de fichas en ese grafo en particular, y luego requería realizar un trabajo de casos. Es claro que tal trabajo de casos no es práctico para grafos con muchos vértices, por lo que aún tenemos la pregunta, ¿qué procedimiento general puede ser aplicado a una configuración inicial de fichas para lograr determinar si tiene una estrategia ganadora o no? Siendo más ambiciosos, pedimos determinar si hay un método constructivo, es decir un método que encuentre al menos una estrategia ganadora en caso de existir.

Comenzaremos con una estrategia intuitiva y más adelante la formalizaremos. Consideremos una configuración de fichas, la identificamos con el divisor D . Primero podemos intentar concentrar la deuda en un sólo vértice $q \in G$. Segundo, podemos intentar usar jugadas que no creen deuda adicional, y con estas jugadas sacar a q de deuda. Decimos que D es *efectivo afuera de q* si $D(v) \geq 0$ para todo $v \in G, v \neq q$. El primer paso que proponemos es una relajación del problema original, porque en lugar de requerir que D sea efectivo, estamos requiriendo que sea efectivo en todos los vértices excepto en uno. Sucede que esta versión relajada siempre es posible.

Lema 2.1. *Sea $D \in \text{Div}(G)$ cualquier divisor, y sea $q \in G$ cualquier vértice. Existe un divisor $E \in \text{Div}(G)$ efectivo afuera de q tal que $D \sim E$.*

Demostración. G es conexo, así que existe una enumeración v_1, v_2, \dots, v_n of $V(G)$, con $v_1 = q$ y tal que v_i está conectada a algún v_j con $j < i$, para $i = 2, 3, \dots, n - 1$. Uno puede obtener tal enumeración por ejemplo tomando $v_1 = q$, luego se enumeran los vecinos de v_1 , luego los vecinos de los vecinos, hasta enumerar todos los vértices.

Tomemos v_n y v_j , con ambos vértices conectados y $j < n$. Ahora se dispara v_j hasta que v_n salga de deuda, luego se dispara $v_{j'}$ hasta que v_{n-1} salga de deuda, con $j' < n - 1$, y se procede en ese orden hasta que v_2 salga de deuda. Esto es posible porque el número de fichas en juego es finita, así que sacar cada vértice v_i de deuda requiere solamente un número finito de jugadas de disparo, y luego de que sacamos a v_i de deuda, ya no lo disparamos, porque ya hemos sacado de deuda a todos los vértices $v_{i'}$ con $i < i'$. La configuración E que resulta satisface las condiciones. \square

Escribimos $G \setminus q$ en lugar de $V(G) \setminus \{q\}$, por conveniencia. La prueba del lema 2.1 es simple y depende únicamente de que G está conectado, pero al mismo tiempo es claro que no estamos siendo eficientes. Al obtener la configuración E es muy probable que tendremos una deuda $E(q)$ más grande de la necesaria. En el camino para demostrar la fórmula de Riemann-Roch vamos a demostrar que siempre podemos obtener un divisor E que satisfaga las condiciones y que adicionalmente $E(q) \geq -g + \deg D$. Por ahora, ¿cómo podemos arreglar esta aparente ineficiencia? Por el lema 1.9 empezariamos a disparar todos los subconjuntos $A \subset G \setminus q$, tales que al disparar no se produzca nueva deuda, y si tenemos suerte esto eliminará la deuda en q . Antes de demostrar que este proceso intuitivo funciona, vamos a definir cómo se mira el resultado final.

Definición 2.2. Sea $q \in G$. Un divisor $D \in \text{Div}(G)$ está q -reducido si D es efectivo afuera de q y para cualquier conjunto no vacío $A \subset G \setminus q$ existe un vértice $v \in A$ tal que $\text{outdeg}_A v > D(v)$.

Ejemplo 2.3. La figura 2.1 presenta dos divisores equivalentes en el mismo grafo. Consideremos v_0 . El divisor D está v_0 -reducido. Esto se verifica con la definición al enumerar los 16 posibles subconjuntos de $G \setminus v_0$ y ver que disparar cualquiera de ellos crea deuda. Ahora consideremos v_3 . Encontramos que D no está v_3 -reducido. Al enumerar los 16 subconjuntos posibles de $G \setminus v_3$ se encuentra que el conjunto $G \setminus v_3$ se puede disparar sin crear deuda. Al disparar $G \setminus v_3$ obtenemos D' , equivalente a D . Este D' cumple con estar v_3 -reducido. \square

El ejemplo anterior nos enseña que la elección del vértice base para reducir importa, podemos tener un divisor reducido respecto a cierto vértice, pero esto no quiere decir que esté reducido respecto a otros vértices. También pareciera ser posible que si un divisor D no está reducido, el disparar los subconjuntos que no crean deuda

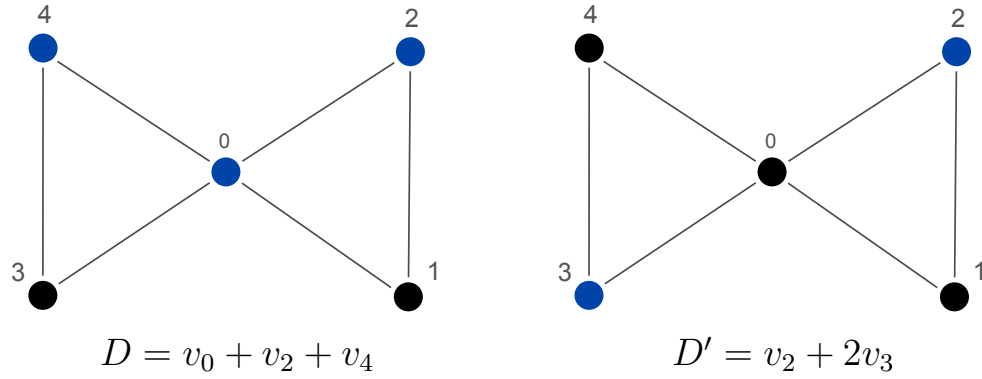


Figura 2.1. Ejemplo de divisor reducido. Elaboración propia.

eventualmente nos da un divisor reducido equivalente a D . Otra pregunta relevante es, una vez encontrado un divisor reducido equivalente a D , ¿es este divisor único? Veremos que la respuesta es afirmativa.

En términos del juego, D está q -reducido si es efectivo afuera de q y cualquier disparo posible de subconjuntos no vacíos $A \subset G \setminus q$ crea deuda. En nuestro método intuitivo esperaríamos que ejecutar el paso 2 varias veces eventualmente termine en un divisor que está q -reducido. Una vez obtenemos el divisor q -reducido, ¿qué información obtenemos de él? Demostraremos que si D está q -reducido, entonces D tiene una estrategia ganadora si y sólo si $D(q) \geq 0$.

Aparte de la utilidad en el juego de disparo de fichas, en el contexto de divisores, estamos diciendo que dado un divisor D , siempre existe un divisor equivalente que está q -reducido, ¡y es único! Entonces los divisores q -reducidos pueden ser utilizados como representantes canónicos de las clases de equivalencia que hemos estado trabajando. Tener estos divisores canónicos nos permite responder la pregunta de decidir si dos divisores son equivalentes o no.

Teorema 2.4 (Proposición 3.1 en [10]). *Fijemos un vértice $q \in V(G)$. Entonces para cada $D \in \text{Div}(G)$, existe un único divisor $D_q \in \text{Div}(G)$ que está q -reducido tal que $D \sim D_q$.*

Esbozo de la demostración. Por el lema 2.1 podemos asumir que D es efectivo afuera de q . En la demostración solamente disparamos conjuntos que no creen deuda. Para un conjunto no vacío $A \subset G$ escribimos $\text{deg}_A D = \sum_{v \in A} D(v)$ para el número de fichas en A .

Primero notamos que los vértices de $S = \{q\}$ nunca disparan, así que el número $\text{deg}_S D$ solamente puede crecer. Supongamos que D no está q -reducido, entonces

existe un conjunto no vacío A , $\{q\} \not\subset A$, que puede ser disparado, lo disparamos para obtener una nueva configuración. Mientras que el divisor no esté q -reducido podemos continuar disparando conjuntos. Supongamos que este proceso continua. Como el número de fichas en el grafo es finito, y ningún vértice afuera de S puede estar en deuda, entonces hay un máximo para el número de fichas que pueden haber en S , esto significa que de algún paso en adelante siempre tendremos $\deg_S D = a$, $a \in \mathbb{Z}$. Así que en ese punto ninguno de los vecinos de S dispara, y le agregamos a S sus vértices vecinos, y repetimos el argumento hasta que S crece a ser todo el grafo, por conexidad.

Para demostrar unicidad supongamos que tenemos divisores $D, D' \in \text{Div}(G)$ que están q -reducidos, tales que $D \sim D'$ y $D \neq D'$. Entonces existe una función $f \in \mathcal{M}(G)$ no constante tal que $D' = D - \Delta(f)$. Sea $A = \text{máx } f$, el conjunto donde f alcanza su máximo. Si $q \in A$ entonces escribimos $D = D' - \Delta(-f)$, para que ahora q esté en el conjunto donde $-f$ es mínimo, y como $-f$ es no constante entonces el conjunto donde $-f$ es mínimo es diferente del conjunto $\text{máx } -f$. Así que luego de intercambiar D con D' si es necesario, podemos asumir que $q \notin A$. Entonces para todo $v \in A$:

$$0 \leq D'(v) = D(v) - \Delta_v(f).$$

Por el lema 1.11 tenemos $\Delta_v(f) \geq \text{outdeg}_A v$, así que $D(v) \geq \text{outdeg}_A v$, por lo que podemos disparar A en D y tenemos $q \notin A$, lo cual contradice que D está q -reducido. \square

Corolario 2.5. Sean $D, D' \in \text{Div}(G)$ divisores y sean D_q, D'_q sus divisores equivalentes que están q -reducidos. Entonces $D \sim D'$ sí y sólo sí $D_q = D'_q$.

Demostración. En caso contrario estaríamos contradiciendo la unicidad de los divisores que están q -reducidos. \square

Corolario 2.6. Sea $D \in \text{Div}(G)$ un divisor y D_q su divisor equivalente que está q -reducido, entonces el juego representado por D tiene una estrategia ganadora si y sólo si $D_q(q) \geq 0$.

Demostración. Una implicación es trivial. Para la otra, supongamos que D es equivalente a un divisor efectivo E , entonces D es equivalente al divisor E_q , la q -reducción de E . En el proceso de calcular la q -reducción de E , notamos que q nunca dispara, así que permanece positivo a lo largo de todo el proceso. Finalmente, por unicidad $D_q = E_q$. \square

2.2. El algoritmo de Dhar

En vista de los dos últimos corolarios, para poder desarrollar aplicaciones prácticas necesitamos encontrar un algoritmo para calcular el divisor q -reducido. Tal algoritmo debe ser capaz de detectar eficientemente si un divisor está q -reducido. Si no está q -reducido, entonces el ejemplo 2.3 nos sugiere que deberíamos obtener un conjunto no vacío $A \subset G \setminus q$ que pueda ser disparado sin crear deuda, y que al iterar este procedimiento podamos llegar al único divisor que está q -reducido.

La definición de q -reducido requiere constatar que cada conjunto no vacío de $G \setminus q$ cumple la condición. Este número de conjuntos crece exponencialmente en n . Presentamos un algoritmo cuyo tiempo es polinomial en m, n , que se constituye como una piedra angular en este trabajo. Presentamos primero un criterio para decidir rápidamente que un divisor está q -reducido, el algoritmo 1, y más tarde iteramos dicho criterio, de acuerdo con [11].

Algoritmo 1: El algoritmo de Dhar

Datos: Un vértice $q \in G$, un divisor $D \in \text{Div}(G)$ efectivo afuera de q .

Resultado: VERDADERO si D está q -reducido, en caso contrario retorna un subconjunto no vacío $A \subset G \setminus q$ el cuál puede ser disparado sin que ningún vértice $v \neq q$ entre en deuda.

Sea $i = 0$, $B_0 = \{q\}$, $A_0 = G \setminus B_0$;

mientras $A_i \neq \emptyset$ **hacer**

$B_{i+1} = \{v \in A_i : D(v) < \text{outdeg}_A v\}$;

si $B_{i+1} = \emptyset$ **entonces**

 Regresar A_i y Parar;

sino

 Poner $A_{i+1} = A_i \setminus B_{i+1}$;

$i \leftarrow i + 1$;

fin

fin

Regresar VERDADERO.

El algoritmo de Dhar admite una interpretación ilustrativa, que será empleada más adelante cuando usemos argumentos que involucren el algoritmo de Dhar. Imaginemos que un incendio inicia en q , y que este incendio se esparce a través de aristas adyacentes. En el vértice v colocamos $D(v)$ bomberos para que puedan parar los incendios entrantes. Un bombero puede prevenir que el vértice se quemé cuando hay únicamente un incendio entrante, dos bomberos pueden prevenir que el vértice

se queme si hay a lo sumo dos incendios entrantes, etc. Si en algún punto hay más incendios entrantes que bomberos en un vértice, entonces los bomberos escapan y el vértice se quema, por lo que el incendio se continua esparciendo mediante las aristas adyacentes. El algoritmo itera en varios pasos, en el paso i los vértices en B_i se queman porque no tienen suficientes bomberos, y debemos quitar estos vértices de A_i , el conjunto de vértices que no se han quemado.

Ejemplo 2.7. Sea G el grafo mostrado en la figura 2.2, le llamamos el grafo *copo de nieve*. Mostramos que para el divisor $D = v_6 + v_7 + v_9 + v_{10} + v_{11}$ el copo de nieve se quema al comenzar un fuego en v_1 .

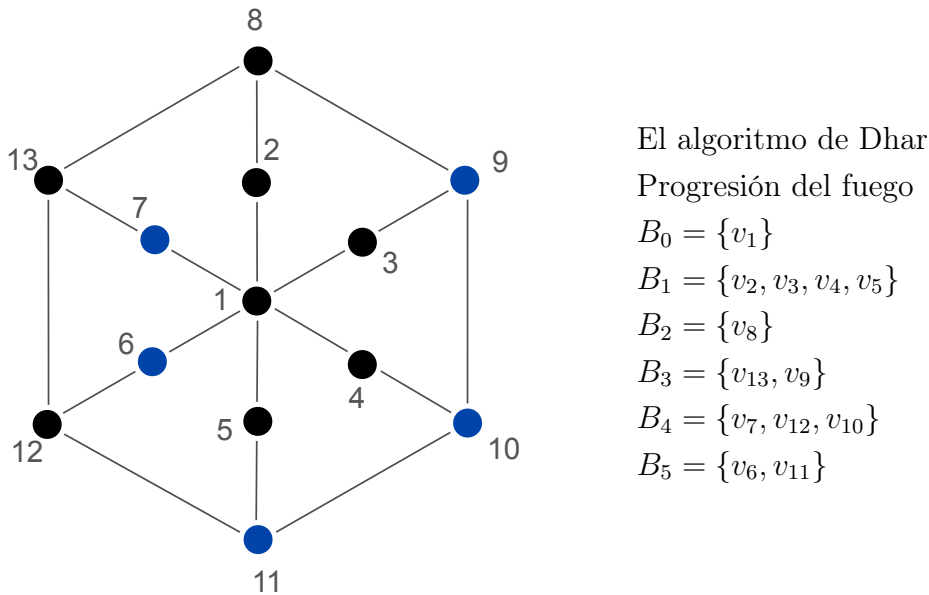


Figura 2.2. Grafo copo de nieve. Elaboración propia.

Por lo tanto por el corolario 2.6 el juego con configuración inicial $D' = D - v_1$ no tiene estrategia ganadora. Claramente este criterio es más fácil de utilizar que la definición de v_1 -reducido. El lector puede volver a ejemplos anteriores que hemos propuesto, aplicar el algoritmo de Dhar y encontrar algunos sorpresas placenteras (se puede intentar quemar el vértice en el centro del ejemplo 1.1, y el vértice v_4 en el ejemplo 1.2). También el lector puede continuar, comenzando fuegos en el ejemplo del copo de nieve, para demostrar que D está reducido respecto a todos los vértices, de donde uno infiere que el juego que se identifica con D tiene exactamente una configuración ganadora. \square

Es claro que si el algoritmo se detiene antes que el incendio consuma todo el grafo, entonces obtenemos un conjunto no vacío A_i que puede ser disparado sin crear

deuda. También es relativamente sencillo demostrar que si el incendio consume el grafo entonces el divisor es reducido. Como el número de vértices disponibles para quemar es finito, entonces el algoritmo termina.

Lo maravilloso del algoritmo de Dhar es su naturaleza constructiva. El algoritmo nos dice que el divisor D es reducido, y si no lo es entonces nos da un conjunto no vacío que previene que D sea reducido. ¿Qué sucede si disparamos este conjunto? Con este procedimiento obtenemos el algoritmo iterado de Dhar, algoritmo 2.

Algoritmo 2: Algoritmo iterado de Dhar

Datos: Un vértice $q \in G$, un divisor $D \in \text{Div}(G)$ efectivo afuera de q .

Resultado: El único divisor q -reducido D_q con $D_q \sim D$

Sea $D_0 = D$, $i = 0$;

repetir

 Repetir el algoritmo de Dhar, asignar a A_i la salida del algoritmo;

si $A_i = \text{VERDADERO}$ **entonces**

 | Regresar D_i ;

sino

 | Asignar $D_{i+1} = D_i - \Delta(\chi_{A_i})$;

 | $i \leftarrow i + 1$

fin

hasta $A_i = \text{VERDADERO}$;

El algoritmo iterado de Dhar termina por un argumento similar al utilizado en el esbozo del teorema 2.4. Notamos que hablando de una manera informal, en cada paso del algoritmo iterado de Dhar estamos escogiendo el conjunto A_i más grande posible que puede ser disparado, así que esperamos que este algoritmo sea bastante eficiente.

Un algoritmo completo para calcular el divisor q -reducido que funcione para cualquier $D \in \text{Div}(G)$ es descrito y analizado en [11]. El proceso que describen es de tres pasos. Dado un divisor $D \in \text{Div}(G)$ el primer paso encuentra un divisor $D' \sim D$ que satisfaga $|D'(v)| \leq \deg v$ para todo $v \neq q$.

El segundo paso encuentra un divisor $D'' \sim D'$ que sea efectivo afuera de q tal que el número de fichas en v está acotado por $\deg v$. Finalmente el tercer paso es el algoritmo de Dhar como lo hemos descrito.

El análisis que ellos realizan del algoritmo emplea métodos de energías. En particular demuestran que los tres pasos juntos se pueden realizar en tiempo polinomial que depende de n y m . Los mismos métodos implican que si nos saltamos el paso

de acotar el número de fichas en cada vértice entonces el algoritmo de Dhar, de la manera como lo hemos descrito aquí, tiene una complejidad polinomial que depende de m , n y $\deg_{G \setminus q} D$, para un divisor D efectivo afuera de q . Para nuestros propósitos teóricos en este trabajo es suficiente con tener el algoritmo de Dhar.

Ejemplo 2.8. Terminamos esta sección con un ejemplo más elaborado en donde reducimos un divisor usando el algoritmo iterado de Dhar. Sea G el grafo de la figura 2.3, le llamamos el *grafo hombre de jengibre*. El ejemplo ha sido computado con el programa incluido en el apéndice II.

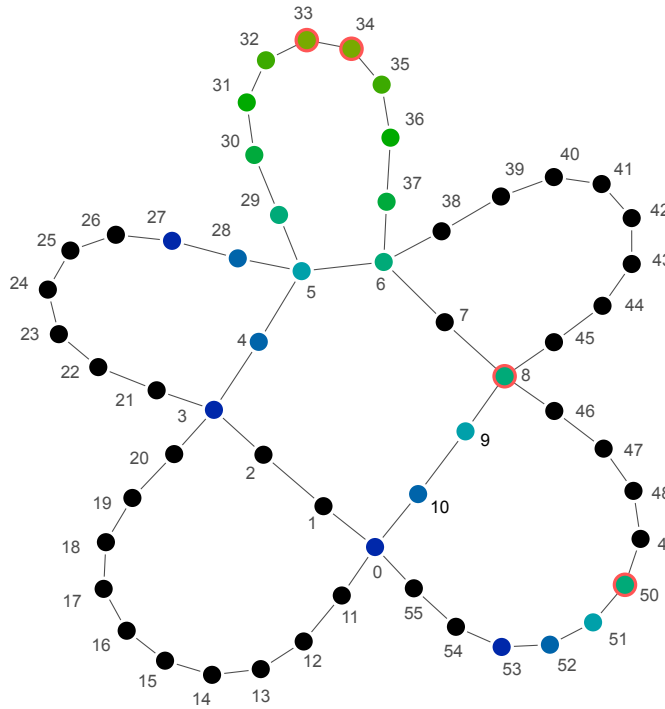
```
cat links.txt
0,1 0,10 0,11 0,55 1,2 2,3 3,4 3,20 3,21 4,5 5,6
5,28 5,29 6,7 6,37 6,38 7,8 8,9 8,45 8,46 9,10
11,12 12,13 13,14 14,15 15,16 16,17 17,18 18,19
19,20 21,22 22,23 23,24 24,25 25,26 26,27 27,28
29,30 30,31 31,32 32,33 33,34 34,35 35,36 36,37
38,39 39,40 40,41 41,42 42,43 43,44 44,45 46,47
47,48 48,49 49,50 50,51 51,52 52,53 53,54 54,55

perl chipfiring.pl

chipfiring: links-file links.txt
chipfiring: reduce 33 1v0 1v3 1v27 1v53
```

Inicialmente en el grafo hombre de jengibre el divisor $D = v_0 + v_3 + v_{27} + v_{53}$ está v_0 -reducido. Procedemos a reducirlo respecto al vértice v_{33} . Usamos colores para ayudar al ojo a seguir el movimiento de las fichas en su recorrido para llegar al divisor v_{33} -reducido D_7 . El último divisor está marcado con un borde rojo. Son siete iteraciones para alcanzar D_7 , y además notamos que $D_7(v_{33}) = 1$, así que $D - v_{33}$ tiene una estrategia ganadora. Si al inicio de este trabajo hubiéramos intentado ganar el juego dado por $D - v_{33}$, probablemente nos hubiéramos encontrado bastante perdidos respecto a cómo proceder. Inicialmente las fichas están bastante distantes de v_{33} , pero gracias al algoritmo iterado de Dhar vemos que sólo es necesario disparar siete conjuntos para alcanzar v_{33} . \square

El algoritmo de Dhar, como se puede apreciar en el ejemplo, ilustra el significado de la frase «los divisores que están q -reducidos son efectivos afuera de q y *tienen sus fichas tan cerca como es posible de q* ». Cada iteración del algoritmo de Dhar trae las fichas más cerca de q , porque estamos disparando conjuntos A_i con $q \notin A_i$. Si deseamos formalizar el concepto de *traer cerca*, podemos introducir una energía



Algoritmo de Dhar

Secuencia de Divisores

$$D_0 = v_0 + v_3 + v_{27} + v_{53}$$

$$D_1 = v_4 + v_{10} + v_{28} + v_{52}$$

$$D_2 = 2v_5 + v_9 + v_{51}$$

$$D_3 = v_6 + v_8 + v_{29} + v_{50}$$

$$D_4 = v_8 + v_{30} + v_{37} + v_{50}$$

$$D_5 = v_8 + v_{31} + v_{36} + v_{50}$$

$$D_6 = v_8 + v_{32} + v_{35} + v_{50}$$

$$D_7 = v_8 + v_{33} + v_{34} + v_{50}$$

Figura 2.3. Grafo hombre de jengibre. Elaboración propia.

potencial, y entonces los divisores reducidos minimizan la energía potencial.

Un lector curioso puede intentar reducir, en el último ejemplo del hombre de jengibre, el divisor dado respecto a todos los vértices. El lector curioso notará que la reducción de D siempre termina con una ficha en el vértice elegido para calcular la reducción. Esta característica es algo muy especial de este D , y dicha característica se convertirá en el tema central de nuestras investigaciones posteriores.

2.3. Orientaciones de grafo

Ahora enfocamos nuestra atención a la tarea de encontrar juegos que no pueden ser ganados. En esta sección exploramos la idea de inducir un divisor desde una orientación de grafo. Uno encuentra dichas técnicas dispersas en toda la literatura, por ejemplo en el influyente artículo de Björner, Lóvasz y Shor que inició el estudio del juego de disparo de fichas [14, Teorema 2.3]. El gran resultado de esta sección es mostrar que ciertas orientaciones especiales responden nuestras preguntas de motivación 1.6 y 1.7, es decir vamos a demostrar que toda configuración con al menos g fichas se puede ganar, y vamos a demostrar la existencia de configuraciones con $g - 1$ fichas que no se pueden ganar.

En la literatura los divisores reducidos también son estudiados bajo el concepto de *funciones de G -parqueo* y una buena referencia para las funciones de parqueo es [12]. Si uno desea generalizar la teoría de esta sección entonces la fuente definitiva es [3], donde el concepto de orientaciones parciales es formulado. La mayoría de las ideas de esta sección vienen de esos dos artículos.

Una orientación de grafo \mathcal{O} le da una dirección a cada arista $e \in E(G)$ al especificar que uno de los vértices incidentes a e sea la *cabeza* y que el otro vértice sea la *cola*. Vamos a denotar con e^+ a la cabeza de la arista, y con e^- a la cola. Decimos que la arista está orientada *hacia* e^+ y que está orientada *saliendo de* e^- . Para un vértice $v \in G$ y una orientación \mathcal{O} definimos el *grado entrante* $\text{indeg}_{\mathcal{O}} v$ como el número de aristas orientadas hacia v , y de manera similar el *grado saliente* $\text{outdeg}_{\mathcal{O}} v$ como el número de aristas orientadas saliendo de v . Cuando no hay peligro de confusión omitimos especificar la orientación de grafo y escribimos $\text{indeg } v$ y $\text{outdeg } v$. A una orientación \mathcal{O} le asociamos el divisor $D_{\mathcal{O}} \in \text{Div}(G)$ con $D_{\mathcal{O}}(v) = \text{indeg } v - 1$ para todo $v \in G$.

Ejemplo 2.9. Con la figura 2.4 mostramos cuatro orientaciones distintas en el grafo G . Los divisores asociados están escritos abajo de la figura.

Nuestras siguientes meditaciones se van a basar en las cuatro orientaciones $\mathcal{O}_a, \mathcal{O}_b, \mathcal{O}_c, \mathcal{O}_d$. Empezamos notando que \mathcal{O}_c ha sido construido al tomar \mathcal{O}_a y revertir la dirección de todas sus aristas. Tal orientación se llama *la orientación revertida* de \mathcal{O}_a y la denotamos con $\bar{\mathcal{O}}_a$. Uno observa que el divisor $D_{\mathcal{O}_a} + D_{\bar{\mathcal{O}}_a}$ es independiente de la orientación \mathcal{O}_a y es igual al divisor $K \in \text{Div}(G)$ con $K(v) = \text{deg } v - 2$ para todo $v \in G$. A este divisor le llamamos el *divisor canónico* y tendrá un rol importante más tarde.

Ahora consideramos las orientaciones $\mathcal{O}_a, \mathcal{O}_b$, las cuáles son orientaciones diferentes pero sus divisores asociados coinciden. Esto implica que la correspondencia que va de orientaciones a divisores no es inyectiva. Nos preguntamos cuáles son los mecanismos que permiten que esto suceda.

Para un grafo G definimos un *camino* como una secuencia de aristas e_1, e_2, \dots, e_i que conectan una secuencia de vértices v_0, v_1, \dots, v_i donde todos los vértices son distintos excepto, posiblemente v_0 y v_i siendo los mismos. Un camino con i aristas tiene *longitud* i . La secuencia de aristas es importante por la posibilidad de aristas múltiples entre dos vértices. Para una orientación \mathcal{O} un *camino dirigido* es un camino donde cada vértice de la secuencia es la cabeza de la arista anterior y la cola de la próxima arista. Por ejemplo en la orientación \mathcal{O}_a un ejemplo de camino dirigido está dado por $\langle v_4, v_1, v_0, v_3, v_5 \rangle$. Un *ciclo* es un camino con $v_0 = v_i$. Un *ciclo dirigido* es un

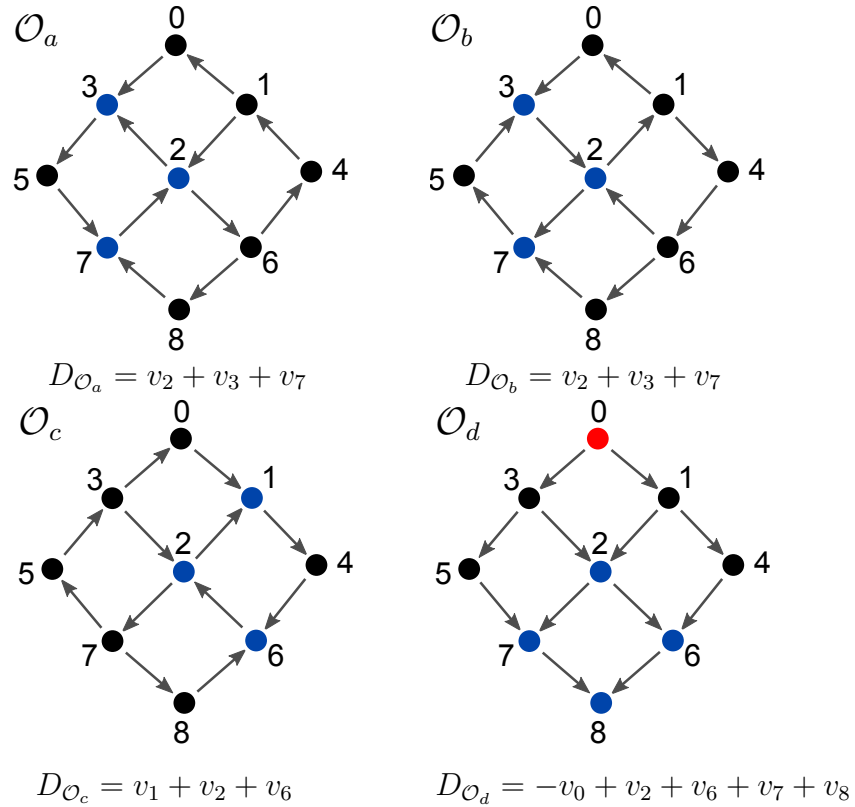


Figura 2.4. Orientaciones de Grafo. Elaboración propia.

camino dirigido con $v_0 = v_i$. Al caminar un ciclo dirigido uno nunca verá un cambio de orientación en las aristas. Un ejemplo de un ciclo dirigido en la orientación \mathcal{O}_b es $[v_3, v_2, v_7, v_5]$.

Ahora observamos que \mathcal{O}_b ha sido obtenido de \mathcal{O}_a al revertir la dirección de dos ciclos de \mathcal{O}_a , el primero siendo $[v_3, v_5, v_7, v_2]$ y el segundo $[v_1, v_2, v_6, v_4]$. Llame-mos a una orientación *acíclica* si no tiene ciclos dirigidos. Uno puede construir tales orientaciones al tomar una enumeración de los vértices v_1, v_2, \dots, v_n y orientar cualquier arista e incidente a v_i, v_j con $i < j$ al hacer que v_i sea la cola y v_j la cabeza. Uno puede demostrar que dicho procedimiento no produce ciclos dirigidos, así que el conjunto de orientaciones acíclicas de G es no vacío.

Sería interesante si *revertir-ciclos* fuera el único mecanismo mediante el cual podemos obtener el mismo divisor asociado a diferentes orientaciones de grafo, porque esto implicaría que la correspondencia de orientaciones a divisores es inyectiva para las orientaciones acíclicas, dado que estas no tienen ciclos para revertir.

Lema 2.10. *Sean \mathcal{O} y \mathcal{O}' dos orientaciones de grafo, entonces $D_{\mathcal{O}} = D_{\mathcal{O}'}$ si y sólo si existe una secuencia finita de operaciones de revertir ciclos que transforma la*

orientación \mathcal{O} en \mathcal{O}' .

Demostración. Sea r_0 el número de aristas cuya orientación en \mathcal{O}' es el reverso de su orientación en \mathcal{O} . Supongamos que $\mathcal{O} \neq \mathcal{O}'$, así que $r_0 > 0$. Tomemos una arista e_0 que apunta hacia un vértice v en \mathcal{O} , y que en \mathcal{O}' está saliendo de v . La condición $D_{\mathcal{O}} = D_{\mathcal{O}'}$ implica que $\text{indeg}_{\mathcal{O}} v = \text{indeg}_{\mathcal{O}'} v$, así que tiene que haber al menos una arista e_1 saliendo de v en \mathcal{O} y apuntando hacia v en \mathcal{O}' para que se mantenga la igualdad. Uno repite el argumento con la arista e_1 y obtiene otra arista e_2 con la orientación revertida, y así uno camina un camino a través de las aristas e_0, e_1, e_2, \dots , donde cada arista e_i tiene su orientación de \mathcal{O} revertida respecto a la orientación de \mathcal{O}' . Eventualmente visitamos un vértice dos veces, así que a partir de este vértice inicia un ciclo dirigido, que revertimos para obtener una orientación \mathcal{O}_1 , cuyo número r_1 de aristas de \mathcal{O}' cuya orientación está revertida en \mathcal{O}_1 es estrictamente menor que r_0 , así que después de revertir un número finito de ciclos, que nos da una secuencia de orientaciones $\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_i$ llegaremos a $r_i = 0$ y habremos terminado. \square

Corolario 2.11. El mapeo {orientaciones acíclicas en G } $\rightarrow \text{Div}(G)$ dado por $\mathcal{O} \mapsto D_{\mathcal{O}}$ es inyectivo.

Sabiendo de esta inyección, se vuelve relevante la pregunta de determinar el conjunto imagen de este mapeo. Llamamos a tales divisores *moderadores*, es decir, divisores $\nu \in \text{Div}(G)$ con $\nu = D_{\mathcal{O}}$ donde \mathcal{O} es acíclica. Para tener una idea de cómo caracterizar a los moderadores vamos a enfocar nuestra atención a \mathcal{O}_d . Esta orientación es acíclica y notamos que $D_{\mathcal{O}_d}(v_0) = -1$. El vértice v_0 es una *fuelle*, un vértice que no tiene aristas apuntando hacia él. Si \mathcal{O} es una orientación y q es una fuente entonces $D_{\mathcal{O}}(q) = -1$. Toda orientación acíclica tiene una fuente. El concepto converso de un vértice sin aristas salientes se llama *sumidero*.

Lema 2.12. Sea \mathcal{O} una orientación acíclica, entonces existe al menos un vértice $q \in G$ que es una fuente para \mathcal{O} .

Demostración. Supongamos que \mathcal{O} no tiene fuentes, entonces $\bar{\mathcal{O}}$ no tiene sumideros. Realizamos una caminata orientada en $\bar{\mathcal{O}}$ empezando en cualquier vértice v (una caminata orientada es la que define un camino orientado). Es posible continuar caminando indefinidamente porque la orientación no tiene sumideros, así que cada vez que uno entra a un vértice u hay por lo menos una arista por la cual uno puede salir. Así que nuevamente hay un vértice que podemos visitar dos veces, y desde aquí comienza un ciclo dirigido en $\bar{\mathcal{O}}$, así que \mathcal{O} tiene ciclos dirigidos también, contradicción. \square

Dado que todos los moderadores tiene al menos un vértice con -1 fichas, nos preguntamos si un juego que comienza en un moderador se puede ganar o no. En el grafo del último ejemplo observamos que \mathcal{O}_d es acíclica, y que incendiar v_0 quema todo el divisor $D_{\mathcal{O}_d}$, así que para este grafo en particular, el divisor $D_{\mathcal{O}_d}$ responde la pregunta 1.7 porque $\deg D_{\mathcal{O}_d} = 3$, el género g de G es 4, y el juego que comienza en $D_{\mathcal{O}_d}$ no puede ser ganado. Esto sugiere que el juego que comienza en ν , con ν un moderador, no puede ser ganado. Si esto fuera cierto entonces respondemos la pregunta 1.7 porque un conteo fácil nos da que $\deg D_{\mathcal{O}} = g - 1$ para todas las orientaciones.

Pensando en responder la pregunta 1.7, llamamos a un divisor $\nu \in \text{Div}(G)$ *no especial* si $\deg \nu = g - 1$ y ν no es equivalente a algún divisor efectivo. Dado un divisor $D \in \text{Div}(G)$, el *sistema lineal* asociado a D es el conjunto $|D| = \{E \in \text{Div}(G) : E \sim D, E \geq 0\}$ de todos los divisores efectivos que son equivalentes a D . Entonces ν es no especial si y sólo si $|\nu| = \emptyset$ y $\deg \nu = g - 1$. Deseamos probar que uno puede construir divisores no especiales mediante las orientaciones acíclicas.

Siguiendo la filosofía de concentrar deuda en un vértice q , vamos a considerar únicamente orientaciones acíclicas con una única fuente en q . Vemos que dado un divisor especial D , al reducirlo respecto q obtenemos un divisor D_q que cumple que $D_q(q) \leq -1$. Considerando D_q , al comenzar un fuego en q este quema todo el grafo. Es una idea natural el intentar asociar una orientación a la manera como el fuego quema D_q , dado que el fuego tiene un origen en q y luego se esparce *alejándose* de un vértice de origen, yendo *hacia* otros vértices. Con eso en mente, proponemos una biyección entre orientaciones acíclicas con una fuente, y divisores q -reducidos.

Ejemplo 2.13. La figura 2.5 muestra que prestar atención a la manera como el fuego consume un divisor reducido D permite recuperar una orientación \mathcal{O} tal que $D_{\mathcal{O}} = D$. En la secuencia de figuras, tomamos $D = 3v_2 + v_8$. Cuando un vértice se incendia, hacemos que todas las aristas que quemaron el vértice apunten hacia él. El lector puede revisar que $D_{\mathcal{O}} = D$ como deseábamos. \square

Teorema 2.14. *Fijemos un vértice $q \in G$. El siguiente mapeo es una biyección:*

$$\{\mathcal{O} \text{ acíclica, una fuente } q\} \rightarrow \{D \text{ no especial y } q\text{-reducido}, D(q) = -1\}$$

dado por $\mathcal{O} \mapsto D_{\mathcal{O}}$.

Demostración. Dividimos la prueba en dos partes, cada una con una idea principal.

Parte 1: Primero mostramos que los elementos en la imagen del mapeo son divisores que están q -reducidos. Sea \mathcal{O} una orientación acíclica con una única fuente

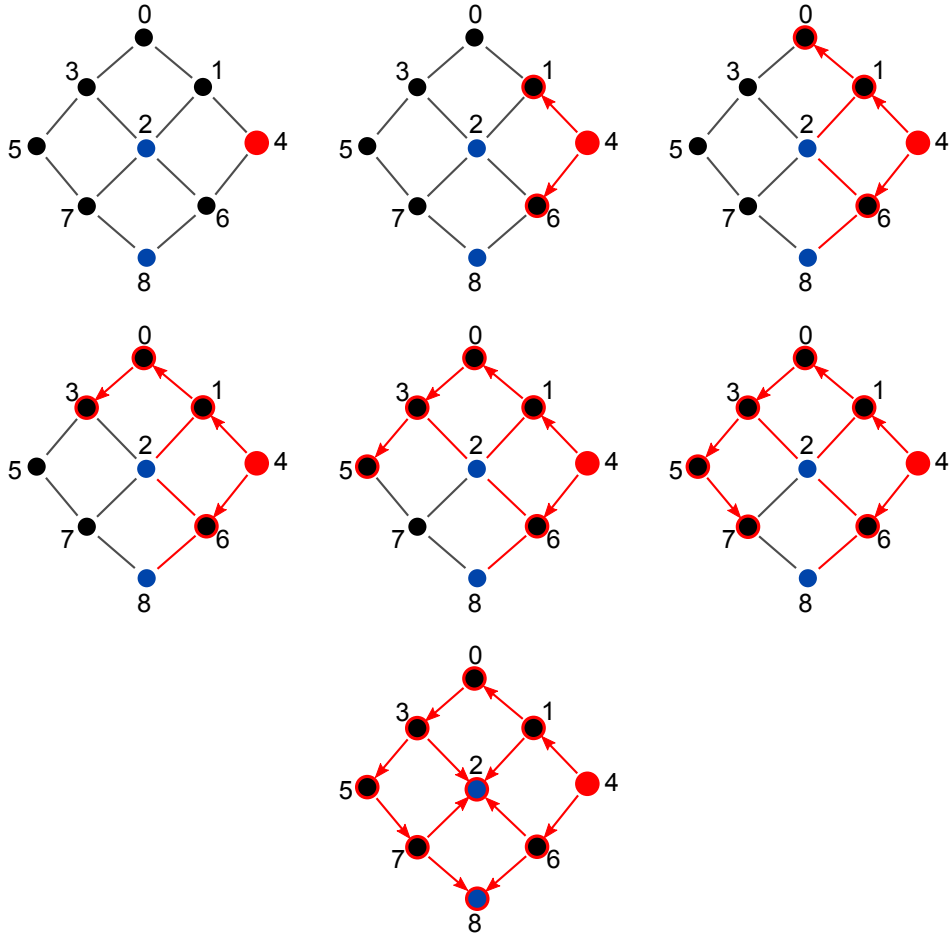


Figura 2.5. Reconstruyendo la orientación. Elaboración propia.

en q , entonces $D_{\mathcal{O}}$ es efectivo afuera de q . Tenemos que demostrar que $D_{\mathcal{O}}$ se quema cuando empezamos un incendio en q . Analicemos los pasos del algoritmo de Dhar, donde en el paso i tenemos que el conjunto A_i de vértices aún no se ha quemado.

Consideremos el subgrafo $G[A_i]$ con el conjunto de vértices igual a A_i . Orientemos las aristas de $G[A_i]$ con la misma orientación que tienen en \mathcal{O} de G , esto le da una orientación acíclica a $G[A_i]$, la cuál por el teorema 2.12 tiene un vértice v_i que es una fuente. Dado que la única fuente de \mathcal{O} es q , y q nunca es un elemento de A_i , entonces el vértice v_i se vuelve una fuente en $G[A_i]$ precisamente porque quitamos A_i^c , es decir, todas las aristas que quitamos estaban apuntando a v_i , y están en el corte (A_i, A_i^c) , pero entonces hay $D_{\mathcal{O}}(v_i) = \text{indeg}_{\mathcal{O}} v_i - 1$ bomberos e $\text{indeg}_{\mathcal{O}} v_i$ fuegos entrantes, así que v_i se quema y el incendio consume todo el grafo.

Parte 2: El corolario 2.11 se encarga de probar que el mapeo sea inyectivo, así que solamente tenemos que mostrar que es sobreyectivo. Sea $D \in \text{Div}(G)$ un divisor

que satisfaga $D(q) = -1$, $\deg D = g - 1$ y que está q -reducido, esto implica que $\deg_{G \setminus q} D = \sum_{v \in G \setminus q} D(v) = g$. Construimos una orientación acíclica \mathcal{O} con $D = D_{\mathcal{O}}$ al llevar registro de la dirección del fuego en el algoritmo de Dhar. Corremos el algoritmo de Dhar para obtener conjuntos B_0, B_1, \dots, B_i . Para todos los vértices $v \in G$ tenemos que $v \in B_j$ para algún j , así que orientamos hacia v todas las aristas que conectan v con algún vértice $u \in B_k$ y $k < j$.

Dado que D está q -reducido entonces el incendio consume D , así que cada vértice $v \in G \setminus q$ termina con al menos $D(v) + 1$ aristas apuntando hacia el vértice. Hemos orientado al menos $\sum_{v \in G \setminus q} (D(v) + 1) = g + (n - 1) = m$ aristas, así que en realidad hemos orientado todas las aristas. Sea \mathcal{O} la orientación resultante. Uno observa que $\text{indeg}_{\mathcal{O}} q = 0$.

Para $v \neq q$ consideremos $D(v) < \text{indeg}_{\mathcal{O}} v$ así que $D(v) \leq \text{indeg}_{\mathcal{O}} v - 1$. Sumamos sobre todos los vértices de $G \setminus q$ para obtener $g = \sum_{v \in G \setminus q} D(v) \leq \sum_{v \in G \setminus q} \text{indeg}_{\mathcal{O}} v - 1 = g$. La igualdad de la suma implica $D(v) = \text{indeg}_{\mathcal{O}} v - 1$ para todo $v \in G \setminus q$ y además $D(q) = -1 = \text{indeg}_{\mathcal{O}} v - 1$ como deseamos. Finalmente es fácil ver que \mathcal{O} es acíclica, porque e está orientada de u hacia v si y sólo si $u \in B_j$, $v \in B_k$ con $j < k$, por lo que es imposible tener ciclos dirigidos porque la existencia de un ciclo dirigido nos daría enteros j, k con $j < k$ y $k < j$. \square

Corolario 2.15. Sea $D \in \text{Div}(G)$ un divisor q -reducido, entonces $\deg_{G \setminus q} D \leq g$.

Demostración. Sea D un divisor con $\deg D > g$. Hacemos el inicio de la parte 2 de la prueba anterior con el divisor D , es decir, iniciamos un fuego en q y suponemos que consume a todo el divisor D , dejamos que una arista e apunte hacia v si v tenía un fuego entrante al momento de quemarse. Cuando contamos cuántas aristas orientamos, obtenemos al menos $\sum_{v \in G \setminus q} (D(v) + 1) > g + (n - 1) = m$, una contradicción. Así que iniciar un fuego en q no puede consumir todo el divisor D , así que por el algoritmo de Dhar el divisor D no está q -reducido. \square

Del resultado anterior obtenemos otra cota, si D está q -reducido entonces $D(q) \geq \deg D - g$. También obtenemos que el número de divisores de grado d fijo que están q -reducidos es finito. En particular esto implica un resultado que ya habíamos enunciado en la sección 1.4, que el número de juegos distintos que existen con d fichas es finito. Esta demostración es intrínseca al juego de disparo de fichas, en lugar de apelar a argumentos de la forma normal de Smith.

Corolario 2.16. Cada juego que empieza en $D \in \text{Div}(G)$ con $\deg D \geq g$ se puede ganar.

Demostración. Supongamos que hay un divisor D con $\deg D \geq g$ que no se puede ganar, entonces por el corolario 2.6 el divisor equivalente que está q -reducido $D_q \sim D$ tiene $D_q(q) \leq -1$, pero entonces $\deg_{G \setminus q} D > g$, contradicción. \square

El conjunto de divisores especiales es no vacío, lo que garantiza que existen juegos de $g - 1$ fichas que no se pueden ganar, y todo juego con por lo menos g fichas puede ser ganado. Queremos resaltar que estos resultados vienen directamente de una introspección cuidadosa del algoritmo de Dhar y de su funcionamiento interno. Hemos andado un largo camino para poder responder las dos preguntas planteadas a finales del capítulo previo y que motivaban la investigación, pero la teoría aún no se acaba. Por ahora cerramos la sección al resumir los resultados en un hermoso teorema que apunta a una dualidad entre los juegos que no pueden ser ganados y los juegos que sí pueden ser ganados. Esta dualidad es que todo juego que no puede ser ganado induce un juego que sí puede ser ganado, y todo juego que sí puede ser ganado induce un juego que no puede ser ganado.

Teorema 2.17. *Para cada divisor $D \in \text{Div}(G)$ existe un moderador $\nu \in \mathcal{N}$ tal que se cumple una de dos, ya sea $D \sim E$ o $\nu - D \sim E$ con $E \in \text{Div}(G)$ un divisor efectivo.*

Demostración. Supongamos que $D \sim E$ con E efectivo, entonces para cualquier moderador ν tenemos que $\nu - D \sim \nu - E$, y el divisor $\nu - E$ no es equivalente a un divisor efectivo, porque por el teorema 2.14 el divisor ν no es equivalente a un divisor efectivo y $\nu \geq \nu - E$.

Supongamos que D no tiene un divisor efectivo equivalente, entonces podemos suponer que D está q -reducido, así que $D(q) \leq -1$ y $\deg D < g$. Nuevamente repetimos la parte 2 de la prueba del teorema 2.14, pero ahora podemos terminar con aristas que no tienen orientación. Uno observa que si e es una arista que conecta u con v que no recibió una orientación entonces $u, v \in B_i$, es decir en el algoritmo de Dhar se queman en el mismo paso i .

Terminamos la demostración si podemos orientar las aristas restantes sin crear ciclos, pero esto es fácil porque solo necesitamos tomar una enumeración de los vértices v_1, v_2, \dots, v_n y como hemos hecho antes, e apunta de u hacia v si u aparece primero en la enumeración. Al hacer esto obtenemos que no hay ciclos contenidos en los conjuntos B_i , y como argumentamos antes no pueden aparecer ciclos cuyos vértices estén en distintos B_j, B_k , así que obtenemos una orientación acíclica \mathcal{O} con un moderador $D_{\mathcal{O}} \geq D$, por lo tanto $D_{\mathcal{O}} - D \geq 0$ como se deseaba. \square

3. Riemann-Roch para grafos finitos

3.1. El juego de Brill-Noether

Esta sección es un pequeño intermedio. Introducimos un nuevo juego con el propósito de ilustrar nuestros estudios posteriores que llevarán a la fórmula de Riemann-Roch. Podría parecer que tenemos ya todas las respuestas a las preguntas relacionadas al juego de disparo de fichas, esto porque el teorema 2.17 responde nuestras preguntas de motivación 1.6 y 1.7, pero vamos a elaborar más sobre la situación del ejemplo 2.8 (el grafo del hombre de jengibre), diremos un par de cosas sobre la situación en la cuál el divisor q -reducido siempre resulta con una ficha en q , esto nos llevará a formular una conjetura abierta.

El juego de *Brill-Noether* es una modificación para dos jugadores del juego de disparo de fichas. Dos jugadores, llamémosles Brill y Noether, eligen dos enteros r y d . Brill propone una configuración D con $\deg D = d$ fichas y luego Noether propone un reto $E \geq 0$ $\deg E = r$. Para vencer el reto, Brill tiene que usar jugadas de disparo para mover las fichas de D y obtener una configuración $D' \geq E$. Noether gana si puede encontrar un reto que Brill no pueda vencer, en caso contrario gana Brill.

Ejemplo 3.1. Sea G el grafo *luna creciente* como se muestra en la figura 3.1. Sea $D = 4v_1$ la configuración inicial y sea $E = v_3 + v_7$ el reto. Deseamos mostrar que $D - E$ es equivalente a algún divisor efectivo.

Una manera de mostrar que Brill puede vencer el reto es mediante el uso repetido del algoritmo iterado de Dhar. Reduciendo D respecto a v_3 , uno mira que $D \sim 4v_3$, así que $D - v_3 \sim 3v_3$. Ahora reduciendo $3v_3$ respecto a v_7 uno obtiene $3v_3 \sim v_3 + 2v_7$. Por lo tanto $D - E = 4v_1 - v_3 - v_7 \sim v_3 + v_7$.

Una forma un poco más ingeniosa es considerar $D' = 2v_1$. El algoritmo iterado de Dhar aplicado a v_7 y luego a v_3 muestra que $D' \sim v_5 + v_9 \sim v_6 + v_8 \sim 2v_7$ y también $D' \sim v_2 + v_4 \sim 2v_3$. Así que D' puede vencer cualquier reto con *una* ficha, por lo tanto $D = 2D'$ puede vencer cualquier reto con dos fichas. Con esto concluimos algo más fuerte, es decir que Brill gana este juego para $r = 2$, $d = 4$ usando la configuración D para jugar.

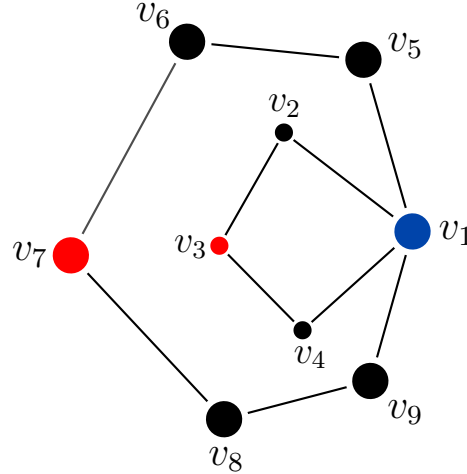


Figura 3.1. Grafo luna creciente. Elaboración propia.

Nos preguntamos si D también sería una configuración ganadora para Brill si $r = 3$. Esta pregunta se responde de una manera rápida, porque al comenzar un fuego en v_1 quema el divisor $v_3 + v_7$, por lo tanto D no puede vencer el reto $E = v_1 + v_3 + v_7$.

En el caso de curvas algebraicas, las curvas con género bajo son bastante simples en lo que respecta a su teoría de divisores. De manera análoga el grafo que mostramos es un grafo de género 2. Está hecho de dos ciclos independientes. Es natural pensar en una generalización, es decir cadenas de n ciclos independientes. Esta generalización la mencionamos en el ejemplo 3.12. Adelantamos que este caso tiene consecuencias sorprendentes. \square

Notamos que si $r = 0$ entonces Noether no puede formular retos, y el juego se reduce a determinar si D tiene un divisor equivalente que sea efectivo, por lo que el juego de disparo de fichas es un caso especial del juego de Brill-Noether. Fijemos un divisor $D \in \text{Div}(G)$, es natural preguntarse cuál es el entero más pequeño r para el cual D no es una configuración que Brill pueda usar para ganar el juego de Brill-Noether. Esta pregunta es análoga a determinar la dimensión del sistema lineal de un divisor en el contexto de curvas algebraicas. Para el divisor D del ejemplo anterior la respuesta a tal pregunta es 3.

Definición 3.2. El *rango* de D es el entero $r(D)$ para el cual Brill puede vencer cualquier reto E con $\deg E = r(D)$, y existe un reto E con $\deg E = r(D) + 1$ el cual Brill no puede vencer. Si D no es equivalente a un divisor efectivo entonces $r(D) = -1$.

Observación 3.3. Podemos escribir $r_G(D)$ si deseamos aclarar cuál grafo G esta-

mos considerando al calcular el rango de D . Esta notación es útil porque también es posible calcular el rango de D en subgrafos de G y en este caso el valor del rango puede variar.

En un sentido intuitivo $r(D)$ está relacionado con $\deg D$. Para $\deg D \geq 0$ tenemos $r(D) \leq \deg(D)$. Podemos estudiar esta situación en el grafo más simple en términos del género, un árbol. Más adelante demostraremos una relación más significativa, la fórmula de Riemann-Roch. De la fórmula de Riemann-Roch se desprende una cota no trivial para el rango, el teorema de Clifford.

Ejemplo 3.4. Recordemos que hemos definido un árbol como un grafo T de género 0. Hay varias definiciones equivalentes para un árbol, entre ellas mencionamos que T es un árbol si y sólo si no tiene ciclos; si entre cualesquiera dos vértices v, w existe exactamente un camino; si al borrar cualquier arista se produce un grafo desconectado. La figura 3.2 muestra un árbol.

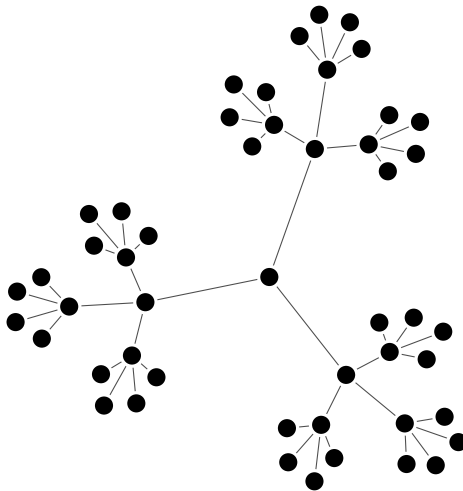


Figura 3.2. Un árbol. Elaboración propia

Si al borrar una arista de G entre dos vértices u_1 y u_2 produce un grafo con dos componentes conexas, entonces podemos disparar una de las componentes para ver que se cumple la equivalencia de divisores $(u_1) \sim (u_2)$. En el caso de un árbol T , borrar cualquier arista desconecta el árbol, y obtenemos que $D_1 \sim D_2$ para cualquiera $D_1, D_2 \in \text{Div}(T)$ si y sólo si $\deg D_1 = \deg D_2$. En particular, $r(D) = \deg D$ para todo $D \in \text{Div}(T)$ con $\deg D \geq 0$, y $r(D) = -1$ si D tiene grado negativo. \square

En el ejemplo anterior la relación entre $r(D)$ y $\deg D$ es bastante simple, y uno observa que la cota trivial que habíamos mencionado es exacta. También vemos que

el juego de disparo de fichas no es particularmente emocionante en un árbol. Dado que queremos alejarnos de situaciones aburridas, podemos hacer un requerimiento más fuerte a la conectividad de G .

La caracterización clave de los árboles es que borrar cualquier arista desconecta el grafo. Por lo tanto definimos un grafo 2-arista-conectado como un grafo G tal que borrar cualquier arista no desconecta G . En general, uno define un grafo k -arista-conectado como un grafo en donde uno puede borrar cualesquiera $k - 1$ aristas y el resultado está conectado.

Supongamos que G no es un grafo 2-arista-conectado y consideremos una arista e que al borrarla resulta en una desconexión. Construimos un nuevo grafo contrayendo la arista, es decir borramos la arista y luego pegamos los dos vértices que eran conectados por la arista, y obtenemos el grafo G_1 y un mapeo $\rho_1 : G \rightarrow G_1$ que envía a los dos extremos de e hacia el vértice que fue pegado en G_1 , y para todos los demás vértices es la identidad. Como G tiene un número finito de aristas, podemos repetir este proceso a lo sumo un número finito de veces, digamos k veces, hasta obtener $\bar{G} = G_k$, con \bar{G} un grafo 2-arista-conectado y un mapeo natural $\rho = \rho_k \circ \rho_{k-1} \circ \cdots \circ \rho_1$. Como nos indican las consideraciones del ejemplo 3.4, nos podemos restringir a estudiar el rango de divisores en grafos 2-arista-conectados.

Lema 3.5. *Sea G un grafo, y \bar{G} el grafo que resulta de contraer todas las aristas de G que no son parte de un ciclo, con $\rho : G \rightarrow \bar{G}$ el mapeo natural como en el párrafo anterior. Para un divisor $D = \sum_{v \in G} D(v)v$ consideramos el divisor $\rho_*(D) \in \text{Div}(\bar{G})$ definido por $\rho_*(D) = \sum_{v \in G} D(v)(\rho(v))$. Entonces $r_G(D) = r_{\bar{G}}(\rho_*(D))$.*

Observación 3.6. De hecho para vértices $u, v \in G$ y para divisores $u, v \in \text{Div}(G)$ tenemos que $u \sim v$ si y sólo si hay un camino único que va desde u hacia v . El lema anterior se sigue de esta observación. Ver [10, lema 4.6, corolario 4.7].

Hay otro ángulo desde el cual podemos fórmular preguntas. En lugar de fijar un divisor D y preguntarnos sobre su rango, podemos fijar un rango r y preguntarnos sobre el conjunto de divisores con ese rango fijo. Un divisor de rango- r es un divisor con rango igual a r . También diremos a veces *al menos rango- r* para significar que el rango del divisor es al menos r . En el contexto de la relación entre $r(D)$ y $\deg D$ preguntamos cuál es el mínimo número de fichas necesarias para obtener un divisor de rango- r .

Esta pregunta fue fórmulada para superficies de Riemann por los matemáticos alemanes Alexander von Brill y Max Noether. En 1874 ellos conjeturaron cuál es el número de fichas necesarias, y dieron un argumento heurístico apoyando su conjetura. Más tarde la misma conjetura fue fórmulada en un contexto más gene-

ral de curvas algebraicas. Se requirió el trabajo de varios matemáticos para lograr demostrar completamente la conjetura, siendo la última pieza demostrada por los matemáticos estadounidenses Phillip Griffiths y Joe Harris[20]. Se cree que una conjetura similar es verdad en este contexto de Riemann-Roch en grafos.

Conjetura 3.7 (Baker). Sea G un grafo de género g . Consideremos el juego de Brill-Noether con parámetros d, r . Sea $\rho(d, r, g) = g - (r + 1)(g - d + r)$. Si $\rho \geq 0$ entonces Brill tiene al menos una configuración ganadora. Si $\rho < 0$ entonces existe al menos un grafo donde Brill no tiene una configuración ganadora.

Para $r = 1$ la conjetura predice la existencia de al menos un divisor D con rango al menos 1 tal que $\deg D = \lceil g/2 \rceil + 1$. El siguiente capítulo contiene datos experimentales respecto a esta conjetura. Incluso en este caso especial el problema demostró ser sorprendentemente rico y algo impredecible. El caso especial permanece abierto, a pesar de que recabamos bastante evidencia numérica apoyando la conjetura.

Conjetura 3.8 (Caso especial). Sea G un grafo de género g . Existe un divisor $D \in \text{Div}(G)$ con grado $\lceil g/2 \rceil + 1$ y rango al menos 1.

3.2. La fórmula de Riemann-Roch

En esta sección culminamos buena parte de los esfuerzos de los capítulos precedentes, con una fórmula que resume la mayoría de los resultados, la fórmula de Riemann-Roch para grafos finitos. Ya hemos hecho todo el trabajo pesado y sólo queda realizar un par de trucos y manipulaciones. Quizás la única tarea difícil que nos queda es motivar la fórmula de Riemann-Roch. ¿Por qué esperamos que esta fórmula se cumpla en los grafos? Respecto a esto diremos simplemente que la analogía con geometría algebraica funciona tan bien, muchas cosas se transfieren de una manera tan natural, que la situación sorprendente sería tener que Riemann-Roch no funciona.

Teorema 3.9 (Riemann-Roch para grafos). *Para cada divisor $D \in \text{Div}(G)$ y para el divisor canónico $K \in \text{Div}(G)$, $K(v) = \deg v - 2$, se cumple la fórmula de Riemann-Roch:*

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

Usaremos el teorema 2.17 para demostrar una fórmula para $r(D)$, la cuál hará que la demostración de la fórmula de Riemann-Roch sea bastante directa. Definimos el *grado positivo* de D como la suma de los coeficientes positivos de D y lo denotamos

como $\deg^+ D = \sum_{v \in G, D(v) > 0} D(v)$.

Lema 3.10. *La siguiente fórmula se cumple, en donde el mínimo se toma sobre todos los divisores $D' \sim D$ y todos los moderadores $\nu = D_{\mathcal{O}}$ para una orientación acíclica \mathcal{O} .*

$$r(D) = \min \deg^+(D' - \nu) - 1.$$

Demostración. Sea $r' = \min \deg^+(D' - \nu) - 1$. Notamos que cada divisor D puede ser escrito de manera única como $D = E - F$, donde ambos divisores E, F son efectivos.

Sea E un divisor efectivo de grado $\deg E = r(D) + 1$ tal que $r(D - E) = -1$. Entonces por el teorema 2.17 existe un moderador ν y un divisor efectivo F tal que $F \sim \nu - D + E$, y entonces $D - \nu \sim E - F$, y para algún divisor $D' \sim D$ tenemos la igualdad $D' - \nu = E - F$, así que $r' < \deg^+(D' - \nu) \leq \deg E = r(D) + 1$.

Elegimos D' y ν tal que $\deg^+(D' - \nu)$ alcanza el mínimo. Entonces $D' - \nu = E - F$, con E y F divisores efectivos y $\deg E = r' + 1$. Escribimos $\nu \sim D - D' + \nu \sim D - E + F$. Como ν no es equivalente a un divisor efectivo, tampoco lo es $D - E + F$, y dado que F es efectivo, entonces $D - E$ no tiene un divisor equivalente que sea efectivo, entonces $r(D) < \deg E = r' + 1$. \square

Demostración de 3.9. Recordemos que para un moderador ν y su moderador dual $\bar{\nu}$ tenemos $K = \nu + \bar{\nu}$. También podemos escribir $\deg D = \deg^+ D - \deg^+(-D)$, entonces $\deg^+(D' - \nu) - \deg^+(\nu - D') = \deg D' - \deg \nu = \deg D' - g + 1$. Por el lema 3.10:

$$\begin{aligned} r(D) + 1 &= \min \deg^+(D' - \nu) \\ &= \min(\deg D - g + 1 + \deg^+(\nu - D')) \\ &= \deg D - g + 1 + \min(\deg^+(\nu - K + K - D')) \\ &= \deg D - g + 1 + \min(\deg^+(-\bar{\nu} + (K - D'))) \\ &= \deg D - g + 1 + r(K - D) + 1. \end{aligned}$$

Transponiendo términos obtenemos la fórmula de Riemann-Roch. \square

Ejemplo 3.11. Sea G un grafo 2-arista-conectado de género 1, es decir, G es un único ciclo de longitud n . En la figura 3.3 tenemos $n = 9$. Notamos que las curvas proyectivas no singulares sobre \mathbb{C} se clasifican, salvo isomorfismos, con una constante conocida como la *j-invariante*. En el caso de grafos que son 2-arista-conectados y de género 1 se pueden clasificar, salvo isomorfismos, mediante la longitud del ciclo.

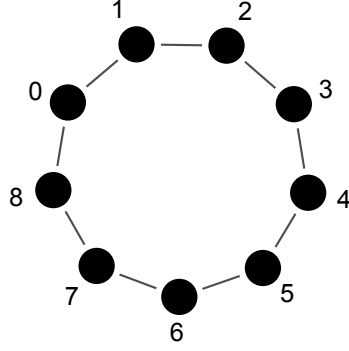


Figura 3.3. Un ciclo. Elaboración propia.

Para $g = 1$ tenemos como divisor canónico a $K = 0$, así que para cualquier divisor D con $\deg D > 0$ obtenemos $r(K - D) = -1$, y aplicando Riemann-Roch obtenemos $r(D) = \deg D - 1$, lo cuál se puede verificar de manera independiente utilizando el algoritmo de Dhar. \square

Ejemplo 3.12. Mencionamos una generalización del ejemplo anterior, esta generalización es estudiada en [17], de donde tomamos la figura 3.4. Numeramos $g + 1$ vértices como v_0, v_1, \dots, v_g y unimos v_i con v_{i+1} utilizando dos caminos, uno con l_i aristas y el otro con m_i aristas. El resultado es una *cadena de ciclos*, como se muestra conceptualmente en la figura. Cada ciclo $\gamma_1, \gamma_2, \dots, \gamma_g$ lo representamos con un círculo para no tener que representar cada uno de los vértices entre v_i y v_{i+1} .

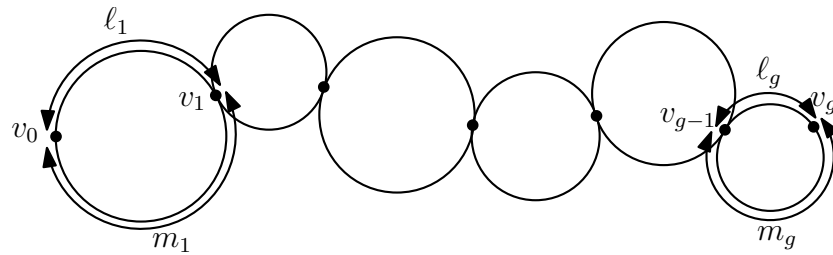


Figura 3.4. Cadena de ciclos. Tomada de [17].

Uno puede demostrar mediante el algoritmo de Dhar que un divisor es v_n -reducido si y sólo si hay a lo sumo una ficha en los conjuntos $\gamma_i \setminus v_i$ para $i \leq n$ y $\gamma_i \setminus v_{i-1}$ para $n < i$. Es decir, afuera de v_n un divisor que esté v_n -reducido tiene a lo sumo dos fichas en cada ciclo γ_i , y este número decrece a una ficha si excluimos los extremos del ciclo v_i o v_{i-1} . Esto sucede porque el fuego tiene que venir desde los dos lados para quemar una ficha en el interior de un ciclo.

Supongamos que $g = 2g'$. El divisor $D = (g' + 1)v_{g'}$ tiene rango al menos 1,

porque $v_{g'}$ está en el medio de la cadena así que tiene g' ciclos a su izquierda y g' ciclos a su derecha, así que al reducir respecto a v_i o v_{i+1} nos da un ciclo γ_i con al menos dos fichas, lo cuál es suficiente para vencer cualquier reto de una ficha. Un argumento similar funciona para $g = 2g' - 1$ and $D = (g' + 1)v_{g'}$. Por lo tanto hemos mostrado que para todo género existe un número finito de grafos para los cuales la conjetura 3.8 es verdadera, es decir que existe un divisor de rango al menos 1 y con $g' + 1$ fichas.

El divisor que usamos es un ejemplo particular. El artículo que hemos citado caracteriza completamente todos los divisores de rango 1, incluso cuenta cuántos hay. También bajo ciertas condiciones de generalidad que se le imponen a las longitudes l_i y m_i uno puede demostrar que ningún divisor de grado g' tiene rango 1. De esta cuenta, junto con otros resultados de la teoría, logran dar una nueva demostración para el teorema clásico de Brill-Noether para curvas algebraicas. \square

Ejemplo 3.13. Hay varios resultados de la teoría clásica que se trasladan a este contexto, que se siguen de una manipulación de la fórmula de Riemann-Roch.

- (a) $r(K) = g - 1$.
- (b) $\deg K = 2g - 2$.
- (c) El teorema de Clifford: Sea $D \in \text{Div}(G)$ con $|K - D| \neq \emptyset$, entonces $r(D) \leq \frac{1}{2} \deg D$.

Recordamos nuestras meditaciones anteriores respecto a la relación entre $r(D)$ y $\deg D$ de la sección previa. Habíamos dicho que trivialmente tenemos $r(D) \leq \deg D$ y más allá de eso solamente pudimos analizar ciertos casos. Ahora podemos decir más, si $|K - D| = \emptyset$ entonces obtenemos que precisamente $r(D) = \deg(D) - g$, y en caso contrario el teorema de Clifford nos da la cota no trivial $r(D) \leq \frac{1}{2} \deg D$. \square

3.3. Un criterio para Riemann-Roch

Cerramos este capítulo reproduciendo el principio de la sección 2 del artículo [10]. Baker y Norine caracterizan la fórmula de Riemann-Roch con dos propiedades combinatorias. El poder de esta formulación general ha motivado trabajo posterior que introduce la formula de Riemann-Roch en objetos llamados grafos métricos y curvas tropicales.

Repasamos la notación que hemos definido hasta ahora, y así abstraemos los conceptos y los llevamos a un contexto más general. Sea X un conjunto no vacío y

sea $\text{Div}(X)$ el grupo abeliano libre sobre X , este grupo consiste en las combinaciones lineales formales de elementos de X , con coeficientes enteros, y un número finito de términos. La operación de suma es una suma formal.

Los elementos de $\text{Div}(X)$, $D = \sum_{x \in X} D(x)x$, son llamados *divisores*. Definamos $\deg D := \sum_{x \in X} D(x)$. Para $D, E \in \text{Div}(X)$ decimos que E domina a D , escribimos $E \geq D$, si $E(x) \geq D(x)$ para todo $x \in X$. Un divisor E es *efectivo* si $E \geq 0$. El soporte de un divisor, denotado por $\text{supp } D$, es el conjunto $\{x \in X \mid D(x) \neq 0\}$.

Sea \sim una relación de equivalencia en $\text{Div}(X)$ que satisface las siguientes condiciones de linealidad:

(E1) Si $D \sim D'$ entonces $\deg D = \deg D'$.

(E2) Si $D_1 \sim D'_1$ y $D_2 \sim D'_2$ entonces $D_1 + D_2 \sim D'_1 + D'_2$.

Definimos el sistema lineal de un divisor D como $|D| := \{E \in \text{Div}(X) : E \geq 0, E \sim D\}$. Los elementos del sistema lineal $|D|$ son todos los divisores efectivos que son equivalentes a D . Definimos el rango de un divisor como una función $r : \text{Div}(X) \rightarrow \{-1, 0, 1, 2, \dots\}$. Para un divisor $D \in \text{Div}(X)$, el rango $r(D)$ es el entero más pequeño tal que existe un divisor efectivo E de grado $r(D) + 1$ tal que $|D - E| = \emptyset$.

Para un divisor $D \in \text{Div}(X)$ uno observa $r(D) = -1$ si y sólo si $|D| = \emptyset$. Un lema fácil de probar, que usamos implícitamente en el ejemplo 3.1, relaciona el rango de $D_1 + D_2$ con los rangos de D_1 y D_2 :

Lema 3.14 (Lema 2.1 en [10]). *Para todo $D, D' \in \text{Div}(X)$ con $r(D), r(D') \geq 0$, tenemos que $r(D + D') \geq r(D) + r(D')$.*

Como definimos en una sección previa, $\text{Div}^d(X)$ son los divisores de grado d . Ahora sea g un entero no negativo, y definamos el conjunto de *divisores no especiales* de grado $g - 1$ como:

$$\mathcal{N} = \{D \in \text{Div}^{g-1}(X) : |D| = \emptyset\}.$$

Sea $K \in \text{Div}(X)$ un elemento de grado $2g - 2$. El siguiente teorema caracteriza la fórmula de Riemann-Roch para elementos del conjunto X/\sim .

Teorema 3.15 (Teorema 2.2 de [10]). *La fórmula de Riemann-Roch:*

$$r(D) - r(K - D) = \deg(D) + 1 - g \tag{3.1}$$

se cumple para todo $D \in \text{Div}(X)$ si y sólo si para todo $D \in \text{Div}(X)$ las siguientes dos propiedades son satisfechas:

(RR1) Existe $\nu \in \mathcal{N}$ tal que exactamente una de las proposiciones es verdad: $r(D) \neq -1$ y $r(\nu - D) = -1$, o si no $r(D) = -1$ y $r(\nu - D) \neq -1$.

(RR2) Si $\deg D = g-1$ entonces exactamente una de las siguientes proposiciones es verdad: $r(D) \neq -1$ y $r(K - D) \neq -1$, o si no $r(D) = -1$ y $r(K - D) = -1$.

Observación 3.16. (i) Si $\deg D \geq g$ en (RR1), entonces $\deg(\nu - D) < 0$, así que $r(\nu - D) = -1$. Uno debe probar que $r(D) \geq 0$, lo cuál corresponde a la pregunta 1.6.

(ii) Si hacemos que $D = 0$ en (RR1) entonces debemos probar que $r(\nu) = -1$, lo cuál corresponde a la pregunta 1.7.

(iii) En el caso donde $X = G$ y \sim siendo la equivalencia de disparo de fichas nos queda que (RR1) es el teorema 2.17. Además (RR2) puede ser demostrado del teorema 2.17 al hacer que $K = D_{\mathcal{O}} + D_{\bar{\mathcal{O}}} = \sum_{v \in G} (\deg v - 2)v$.

(iv) La propiedad (RR2) es equivalente a $r(K) \geq g - 1$.

La prueba de la equivalencia entre la fórmula de Riemann-Roch y las condiciones (RR1) y (RR2) sigue argumentos análogos a los que hicimos de manera específica para grafos. Mencionamos una versión general del lema 3.10, que ahora se vuelve una consecuencia de RR1.

Lema 3.17 (Lema 2.7 en [10]). *Si (RR1) se cumple, para cada $D \in \text{Div}(X)$ tenemos que $r(D) = \min \deg^+(D' - \nu) - 1$, donde el mínimo se toma sobre todos los divisores $D' \sim D$ y los divisores no especiales ν .*

Notamos que la fórmula de Riemann-Roch responde la pregunta de cómo se comportan los divisores con más de $2g - 2$ fichas. Desde este punto en adelante nos interesarán los divisores con un grado relativamente bajo, es decir aquellos cuyo comportamiento no sea explicado completamente por la fórmula de Riemann-Roch, aquellos con $|K - D| \neq \emptyset$. Específicamente, la conjetura 3.8 se vuelve ahora nuestro objeto de estudio.

4. Ejemplos Numéricos

Este capítulo contiene resultados numéricos obtenidos como evidencia de apoyo a la conjetura 3.8, es decir calculamos muchos ejemplos donde encontramos para un grafo G de género g todas las clases de divisores de grado $\lceil g/2 \rceil + 1$ con rango al menos 1. Los ejemplos son calculados usando un programa, cuyo algoritmo es descrito y el código fuente se puede encontrar en el apéndice B de este trabajo.

Con el programa de computadora se verificó la conjetura para alrededor de un millón de grafos aleatorios con un género que varía entre 6 y 12, y un número de vértices que varía entre 10 y 120. No se encontraron contraejemplos a la conjetura. Algunos grafos llegaron a estar cerca a ser un contraejemplo, ya que sólo poseen una única clase de divisores con rango 1.

También exploramos un par de cotas. Como en las secciones anteriores, sean n y m el número de vértices y de aristas de G respectivamente. Trivialmente, si tenemos n fichas entonces uno puede obtener un divisor de rango al menos 1 al colocar una ficha en cada vértice. Por lo tanto la conjetura es verdadera para el caso $n \leq \lceil g/2 \rceil + 1$, es decir cuando $n \leq g/2 + 1 = (m - n + 1)/2 + 1 \leftrightarrow 3n - 3 \leq m$.

Asumamos $m < 3n - 3$. Uno puede continuar este juego de las construcciones utilizando ideas más ingeniosas. Por ejemplo, supongamos que G es un grafo simple (uno que no tiene aristas múltiples), y sea S un conjunto independiente de G , es decir, para cada pareja de vértices $u, v \in S$ se tiene que no hay arista que los conecte. Coloquemos una ficha en cada vértice afuera de S , obtenemos un divisor de rango al menos 1. Tomemos un conjunto independiente máximo, es decir, un conjunto independiente con el máximo número posible de vértices. La cardinalidad de tal conjunto se denota $\alpha(G)$, se conoce como el número de independencia de G , y algunas cotas son conocidas [21]:

$$\alpha(G) \geq \sum_{i=1}^n \frac{1}{1 + \deg v_i} \geq \frac{n^2}{n + 2m} > n/7.$$

La primera desigualdad fue descubierta de manera independiente por los mate-

máticos Caro y Wei. La segunda desigualdad es una aplicación de la desigualdad de media aritmética-media armónica que nos da $\sum 1/h_i \geq n^2/(\sum h_i)$. El resultado es que si tenemos alrededor de $6n/7$ fichas entonces podemos construir un divisor con rango al menos 1 al encontrar un conjunto independiente con al menos $n/7$ vértices, y luego colocando una ficha en cada vértice del conjunto complemento. Esto es una pequeña mejora de la cota trivial de tener n fichas.

El autor considera que debería ser posible encontrar una construcción explícita que cubra el caso cuando $n \leq g < 2n$, es decir, si tenemos alrededor de $n/2$ fichas nos preguntamos cómo construir explícitamente un divisor de rango al menos 1. En un sentido intuitivo, tenemos una ficha para cada dos vértices, así que vencer desafíos de una ficha debería ser sencillo. Quizás métodos probabilísticos puedan ser usados.

Concluimos entonces que la verdadera naturaleza del problema reside en ejemplos donde el número de vértices sea grande comparado con el género. Nos interesa ganar intuición en este caso. Para eso desarrollamos un programa que dado un grafo G encuentra todas las clases de divisores en $\text{Jac}(G)$ de rango 1. Sabemos que la cardinalidad de $\text{Jac}(G)$ es igual a κ , el número de árboles generadores de G , y que por el teorema de Kirchhoff este número se puede calcular fácilmente del laplaciano combinatorio L . Así que conociendo el número de clases de equivalencia de rango 1, lo podemos comparar fácilmente al número total de clases de equivalencia, y saber qué tan probable es que un divisor cualquiera resulte tener rango 1.

Describimos el algoritmo que el programa usa, el algoritmo 3. Fijamos un vértice v_0 , y expresamos las clases de equivalencia con su representante v_0 -reducido. La implementación se realizó en el lenguaje Perl (ver apéndice B). La implementación también incluye una función para generar grafos aleatorios que cumplan con ser 2-arista-conectados. Con esta implementación exploramos la conjetura 3.8.

Algoritmo 3: Encontrar todas las clases de divisores con rango 1

Datos: Un v3rtice $q \in G$, una enumeraci3n E_1, \dots, E_N de todos los divisores efectivos de grado $\lceil g/2 \rceil$

Resultado: El conjunto R_1 de representantes q -reducidos de las clases de divisores de rango 1.

Funci3n *EsRangoUno*(D : un divisor efectivo) es

$B = \text{supp } D$;

repetir

 Escoger $v \in G, v \notin B$;

 Ejecutar el algoritmo iterado de Dhar, con argumentos v y D , hacemos que D_1, D_2, \dots, D_M sean los pasos intermedios del algoritmo de Dhar;

si $D_M(v) = 0$ **entonces**

 | regresar FALSO;

fin

$B \leftarrow B \cup \text{supp } D_1 \cup \text{supp } D_2 \cup \dots \cup \text{supp } D_M$;

hasta $|B| = n$;

regresar VERDADERO;

fin

$R_1 = \{\}$;

para $D \leftarrow (q) + E_1$ **hasta** $(q) + E_N$ **hacer**

si $D(v) \geq \deg v$ para alg3n $v \in G$ **entonces**

 | saltarse a la siguiente iteraci3n

fin

 Ejecutar el algoritmo de Dhar con argumentos q y D , poner el resultado en la variable SEQUEMA;

si no SEQUEMA **entonces**

 | saltar a la pr3xima iteraci3n

fin

si *EsRangoUno*(D) **entonces**

 | $R_1 \leftarrow R_1 \cup \{D\}$

fin

fin

Regresar R_1 .

La función `EsRangoUno` funciona para detectar si un divisor es de rango al menos 1 por el hecho que D es de rango al menos 1 si y sólo si $D_v(v) \geq 1$ para todo $v \in G$, donde $D_v \sim D$ y D_v está v -reducido. Mantenemos registro de los pasos intermedios del algoritmo iterado de Dhar, específicamente por cuáles vértices ha pasado una ficha, para así disminuir el número de veces que necesitamos ejecutar el algoritmo iterado de Dhar. Cuando vemos que un vértice v_j tiene una ficha en un paso intermedio lo incluimos en el conjunto B . Únicamente reducimos respecto a vértices afuera de B , es decir vértices para los cuáles aún no hemos visto un divisor que pueda vencer un reto de una ficha en v_j .

Necesitamos probar esto con todas las clases de divisores, así que hacemos una enumeración exhaustiva de divisores. Lo ideal sería escoger un representante de cada clase de divisores y correr la función `EsRangoUno` en cada uno de estos representantes. Para lograr esto usamos divisores que estén q -reducidos. Podemos asumir que el divisor q -reducido tiene una ficha en q , porque en caso contrario no sería de rango 1. Por lo tanto escogemos divisores de la forma $D = (q) + E_i$, con E_i efectivo de grado $\lceil g/2 \rceil$. Tenemos en total $\binom{n-1+\lceil g/2 \rceil}{n-1}$ divisores que cumplen tales condiciones. Antes de correr el algoritmo de Dhar en D , revisamos si D satisface $D(v) < \deg v$ para todo $v \in G$, una condición necesaria para ser q -reducido y que es más rápido de verificar que correr el algoritmo de Dhar. Si el algoritmo de Dhar nos dice que no es reducido, nos saltamos al siguiente divisor. Si es reducido, entonces ejecutamos la función `EsRangoUno` y agregamos el divisor a la lista de divisores de rango 1 si el resultado sale positivo.

Hay dos posibles optimizaciones al implementar la función `EsRangoUno`. La primera es cómo escogemos $v \in B^c$. En nuestra implementación tomamos $v \in B^c$ maximizando $\deg v$. En las pruebas del programa este cambio dió un decremento del 20% del tiempo necesario para computar la respuesta. La justificación heurística de esto es que estamos probando vértices con una probabilidad mayor de hacer que D falle ser de rango al menos 1.

La segunda optimización posible es regresar VERDADERO tan pronto como alguno de los casos intermedios D_j satisfaga que $D_j(v) > 0$, sin llegar a computar el divisor reducido D_M . Esta optimización hace una gran diferencia cuando el número de fichas es relativamente grande en comparación con el número de vértices, por ejemplo al tener cerca de $n/2$ fichas disponibles. En ese caso se vuelve relativamente rápido darse cuenta si un reto se puede vencer, pero computar el divisor q -reducido se vuelve algo lento, porque el número de pasos intermedios depende del número de fichas afuera de q .

El algoritmo está implementado en la función `_all_divisors_iterator1`, y puede ser llamado utilizando el método `all_rank1` que opera en un objeto de `ChipFiring`.

Para obtener ejemplos empezaremos fijando valores para g y para n . Idealmente los grafos que estudiemos deberán ser 2-arista-conectados, porque en caso contrario por el lema 3.5 existen relaciones de equivalencia triviales entre los vértices, y deseamos utilizar el parámetro n para una descripción estadística de los grafos. El problema de muestrear aleatoriamente el espacio de grafos que estén 2-arista-conectados y que tengan un número de vértices dado es muy interesante en sí mismo, y es no trivial. Dado que nuestro fin no es resolver este problema, nos contentamos con un procedimiento que genere grafos conectados tales que el grado mínimo de los vértices es 2. Todo grafo que sea 2-arista-conectado satisface estas condiciones.

Seguimos un procedimiento sencillo para generar los grafos. Comenzamos con n vértices. Aleatoriamente seleccionamos parejas para unir las con una arista. Cuando un vértice alcanza grado 2, ya no lo consideramos como candidato para hacer una pareja. Este proceso continúa hasta que todos los vértices tengan grado al menos 2. Entonces consideramos nuevamente todos los vértices y seleccionamos parejas al azar para unir con aristas hasta alcanzar el número de aristas necesario para tener el género deseado. Al final revisamos si el resultado está conectado. Si uno desea grafos que sean 2-arista-conectados, entonces en este punto se podría agregar una función que revise dicha condición. Este código está implementado en la función `randomgraph12`. Para poder generar grafos aleatorios se usa el método `random_graph`, el cuál selecciona entre varios métodos de grafos aleatorios, aunque actualmente sólo tenemos uno implementado, el que hemos descrito.

Ya con esto podemos generar muchos grafos y procesarlos con nuestro algoritmo. Por motivos de tiempo computacional las muestras tienen un género relativamente bajo. Específicamente trabajamos con género 6 y con género 8 de manera extensa. Los conjuntos de datos que fueron generados se encuentran en el repositorio de github <https://github.com/AV-2/chipfiring>.

Como primer experimento usamos el generador de grafos aleatorios para tomar 90 muestras. Cada muestra contiene por lo menos mil grafos de género 6, todos con el mismo número de vértices. Hay una muestra de grafos con diez vértices, una de grafos con once vértices, y así hasta 100 vértices. Entonces usando el algoritmo 3 contamos para cada grafo el número de clases de divisores con rango al menos 1. La siguiente gráfica resume la información al calcular, para cada valor n de vértices,

el promedio del número de clases de divisores de rango al menos 1 para la muestra tomada.

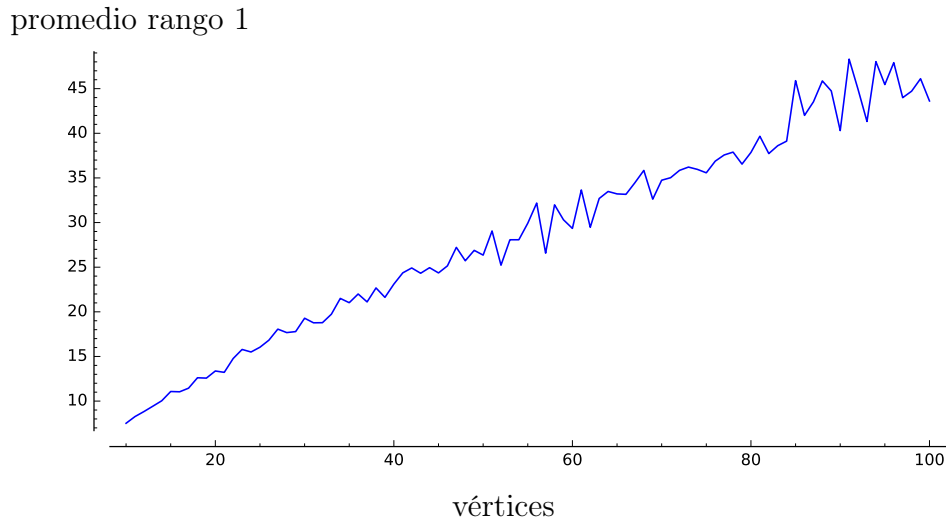


Figura 4.1. Género 6, promedio de número de divisores de rango 1 en función del número de vértices. Elaboración propia.

Observamos en la figura 4.1 que a medida que el número de vértices crece, también crece el número de clases de divisores de rango 1, en promedio. Dado que este es un comportamiento promedio, vemos un caso específico. Tomamos una muestra de 96000 grafos con género 6 y 25 vértices. El histograma de la figura 4.2 muestra la distribución de cuántos grafos tienen un número determinado de clases de divisores con rango al menos 1.

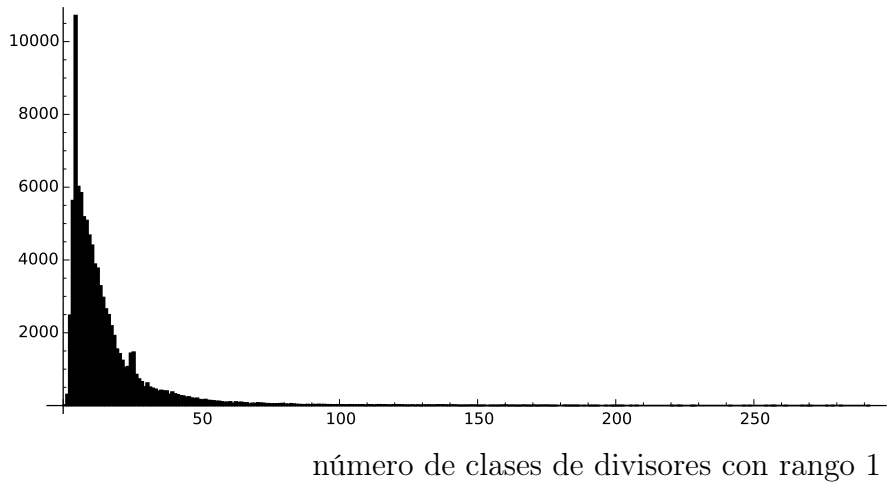


Figura 4.2. Género 6, histograma de divisores de rango 1. Elaboración propia.

Observamos que hay un pico en cinco. La muestra que realizamos tiene 10716 grafos con exactamente 5 clases de divisores de rango al menos 1. Esto es algo interesante, así que nos preguntamos si en las otras muestras también cinco es el valor más observado. Revisamos y en efecto, en todas las muestras sucede este fenómeno. En la gráfica de la figura 4.3 mostramos el porcentaje de grafos de las muestras que tienen exactamente 5 clases de divisores de rango 1.

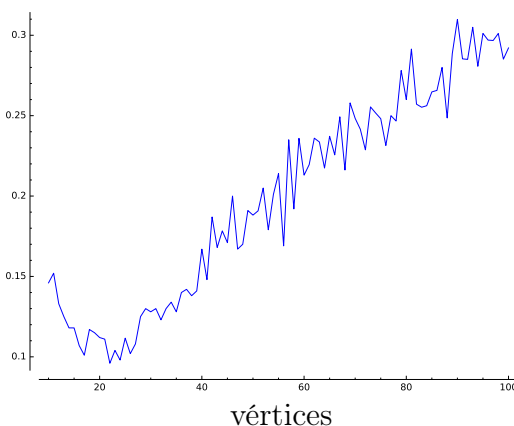


Figura 4.3. Género 6, porcentaje de grafos con 5 divisores de rango 1. Elaboración propia.

Estos resultados coinciden con una fórmula de conteo dada en el artículo [17]. En el artículo se trata solamente el caso de cadenas de ciclos, y hay unas condiciones de generalidad para estos grafos (definición 4.1). Los resultados de arriba nos llevan a creer que también en el caso general habrán condiciones que se le pueden imponer al grafo para asegurar un conteo de clases de equivalencia de rango al menos 1. Para $g = 6$ uno necesita al menos 61 vértices, según la definición 4.1 del artículo citado, para tener una cadena *genérica* de ciclos. Es natural entonces que a medida que el número de vértices se incrementa, en nuestra situación, veamos más grafos con exactamente cinco clases de equivalencia de rango al menos 1.

Finalmente, graficamos el tiempo promedio que le toma al programa realizar los cálculos en un procesador de 2.5 GHz, en la figura 4.4. De nuestro algoritmo no hacemos un análisis del tiempo de ejecución, pero dado que el número de divisores efectivos de grado 4 para un grafo de género 6 tiene crecimiento polinomial a medida que n se incrementa, y que las otras partes son polinomiales también, esperaríamos en general un comportamiento polinomial.

Repetimos el experimento con grafos con un género igual a 8. Con género igual a 8 tenemos 5 fichas para formar los divisores, así que el programa necesita más tiempo.

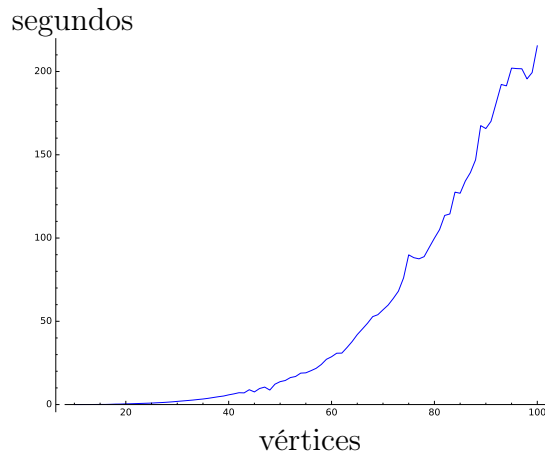


Figura 4.4. Género 6, tiempo de procesamiento en función del número de vértices. Elaboración propia.

Por lo tanto sólo pudimos tomar muestras de grafos con un número de vértices entre 10 y 50. Otra vez cada muestra tiene al menos mil grafos, y en cada muestra el número de vértices es constante. La gráfica de la figura 4.5 muestra el promedio del número de clases de divisores de rango al menos 1. Aquí encontramos un número más alto, y nuevamente se da un crecimiento aparentemente lineal respecto a n , el número de vértices.

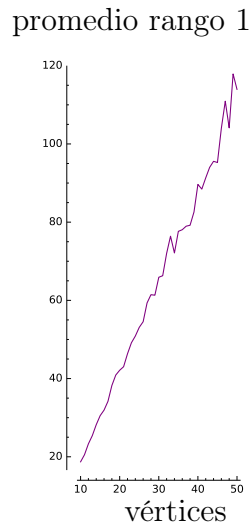


Figura 4.5. Género 8, promedio de número de divisores de rango 1 en función del número de vértices. Elaboración propia.

Ahora revisamos nuevamente cuál es el número más frecuente, y a diferencia del caso anterior, nos encontramos con respuestas distintas para cada muestra. Parece que necesitamos más grafos para poder hacer conjeturas, pero para esto no poseemos

suficiente poder computacional, o tiempo. El histograma de la figura 4.6 resume los datos de una muestra que cumple $g = 8$ y $n = 25$, en donde 25000 grafos fueron muestreados. El histograma está recortado, dado que tenemos un grafo con 1022 clases de divisores de rango al menos 1, pero a partir de 300 clases de divisores la frecuencia cae bastante.

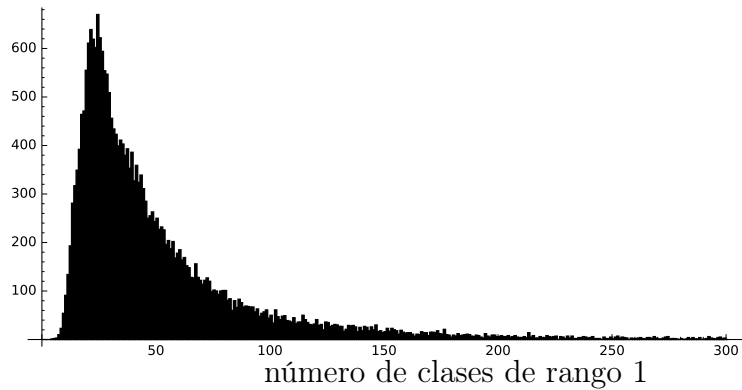


Figura 4.6. Género 8, histograma. Elaboración propia.

Hay un pico en 24, para el cuál hay 670 grafos que tienen 24 clases de divisores de rango al menos 1. Ahora mostramos en una gráfica los números más frecuentes para el conteo de clases de divisores de rango al menos 1. Como hemos indicado ya, en este caso varía para cada muestra. Además mostramos una gráfica del tiempo en la figura 4.7.

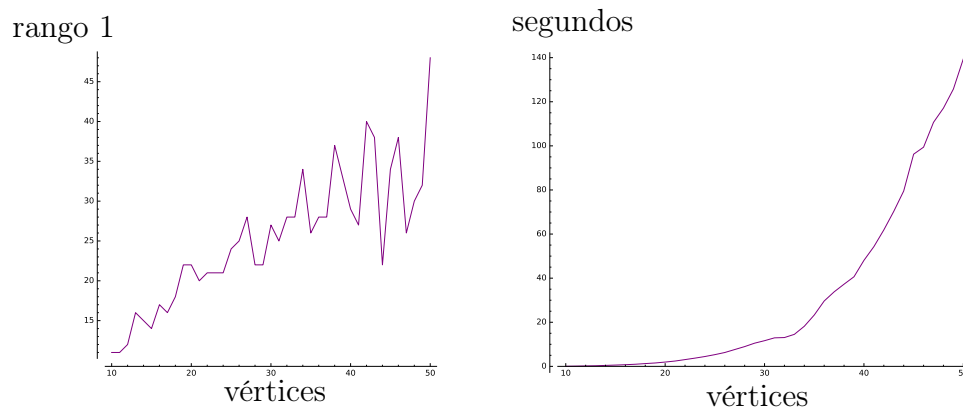


Figura 4.7. Género 8, rango y tiempo. Elaboración propia.

Mostramos ahora un ejemplo particular de todos los grafos que hemos muestreado. Con este grafo particular tuvimos la idea de estudiar cómo varía el número

de clases de divisores de rango al menos 1 cuando se le aplican pequeños cambios a un grafo. Usamos el grafo de la figura 4.8.

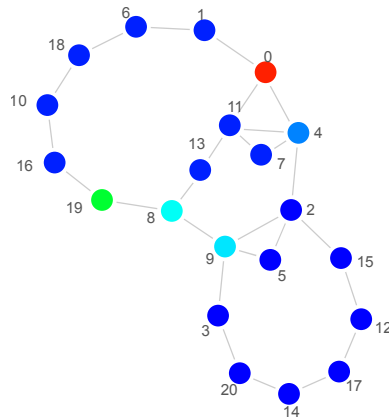


Figura 4.8. Grafo inicial para experimento. Elaboración propia.

Este grafo tiene género 6 y 21 vértices. El número de clases de divisores con rango al menos 1 es 32. Iniciando en este grafo, escogimos una arista y la subdividimos en dos, esto se logra insertando un vértice en medio de la arista. Iteramos el procedimiento y en cada paso calculamos el número de clases de divisores de rango al menos 1. Escogimos un grafo que tuviera un número de clases de divisores de rango al menos 1 diferente de cinco de manera deliberada, para poder estudiar la pregunta de si al agregar más vértices al grafo entonces en algún punto el número de clases de divisores de rango 1 se incrementa, o si decae hasta llegar a 5, el valor común que habíamos observado antes. En la gráfica de la figura 4.9 mostramos la evolución del grafo, específicamente la evolución del número de clases de divisores de rango al menos 1, hasta llegar a 93 vértices. Inicialmente vemos una caída en el número de clases de divisores de rango al menos 1, pero dura poco y luego comienza a crecer a medida que subdividimos más aristas. La relación pareciera ser lineal. En varios puntos vemos que el número de clases de divisores de rango al menos 1 permanece constante a pesar de las subdivisiones. Esto sucede cuando el grafo tiene 45 vértices por ejemplo. Cuando tiene 69 vértices hay 83 clases de divisores de rango al menos 1, y este número salta rápidamente a 95 cuando alcanzamos 70 vértices. Luego continúa con este crecimiento lineal hasta tener 73 vértices, en este punto alcanza 101. En la siguiente iteración, parece que se corrige el salto que ocurrió en setenta, porque el número 101 cae a 93.

Estos saltos son interesantes porque sugieren que ciertas longitudes métricas específicas del grafo pueden crear o imposibilitar un número extra de divisores.

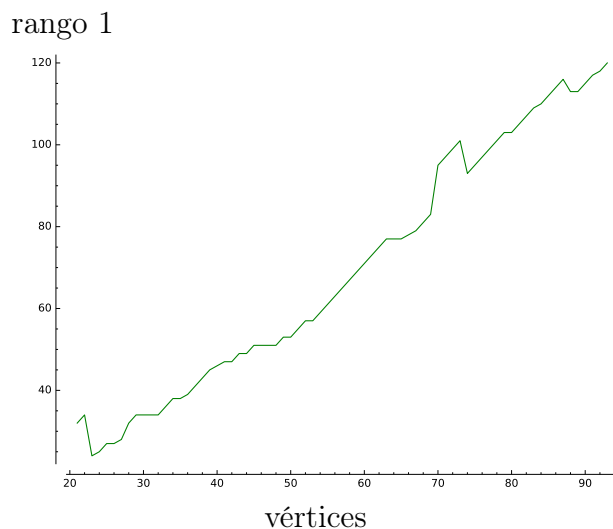


Figura 4.9. Evolución rango 1. Elaboración propia.

Mostramos el grafo cuando tiene 73 vértices y 101 clases de divisores de rango al menos 1, en la figura 4.10. Lo llamamos el *grafo pez tropical*. La manera como han sido numerados los vértices refleja cómo se hizo el proceso de subdivisión. Respecto a cómo hemos coloreado los grafos, hemos tomado los divisores de rango al menos 1 y considerado sus representantes que están v_0 -reducidos. Inicialmente los vértices son grises. Un vértice está coloreado si está contenido en al menos uno de los soportes de los representantes v_0 -reducidos. Si está contenido en más de un soporte, entonces el color va cambiando de azul a verde, luego amarillo, naranja y finalmente rojo en función del número de soportes que incluyen el vértice dado.

Con esta coloración los vértices grises no pertenecen a ninguno de los soportes de los representantes v_0 -reducidos. Vemos cómo hay tres regiones entre v_9 y v_2 que aparecen totalmente grises. Este es un fenómeno que se repite en otros grafos, el de tener regiones grises.

Este experimento nos deja con dos preguntas. Primero habría que responder si es posible hacer crecer el número de clases de divisores de rango al menos 1 de manera indefinida al ir agregando más y más vértices. Segundo, nos podemos plantear encontrar una explicación a por qué existen dichas regiones de color gris.

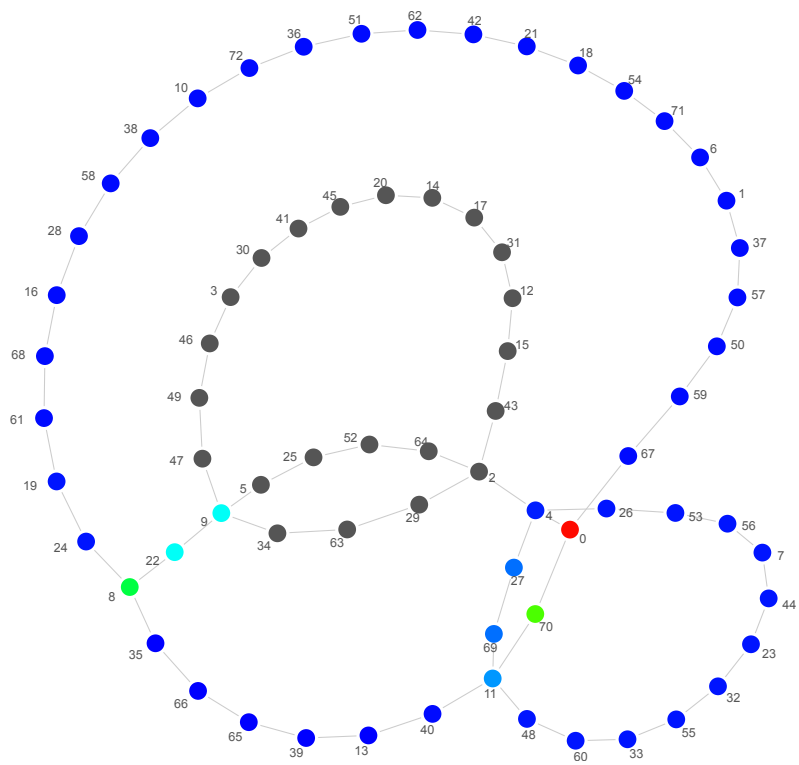


Figura 4.10. Grafo pez tropical. Elaboración propia.

CONCLUSIONES

1. El número total de juegos distintos en un grafo G conexo, con un número de fichas fijo, es igual al número de árboles generadores que tiene G .
2. En un grafo G conexo y de género g todo juego de disparo de fichas con g fichas, o más, se puede ganar.
3. En un grafo G conexo y de género g existen juegos de disparo de fichas con $g - 1$ fichas que no se pueden ganar. Estos juegos están inducidos por una orientación acíclica con una sola fuente.
4. Dado un juego en un grafo G conexo, se puede usar el algoritmo iterado de Dhar para determinar si tiene estrategia ganadora o no. En caso afirmativo, el algoritmo encuentra una estrategia en particular.
5. En un grafo G de género g la fórmula de Riemann-Roch expresa el rango de un juego con más de $2g - 2$ fichas en función de su número de fichas.
6. Un grafo G conexo y 2-arista-conectado se puede ver como un análogo discreto de una superficie de Riemann compacta.
7. Hay una cantidad fuerte de evidencia para creer que existe un análogo de la teoría de Brill-Noether en el contexto de grafos.

RECOMENDACIONES

1. Los primeros tres capítulos de esta tesis se pueden usar para un curso corto sobre juegos de disparo de fichas.
2. El estudio del grupo jacobiano de un grafo se puede asignar como tema de trabajo de graduación, hay muchos resultados del grupo jacobiano a los que este trabajo de graduación aludió sin entrar a estudiarlos a fondo.
3. Se recomienda agregar el estudio de teoría de grafos y sus aplicaciones a los temas que se imparten en la licenciatura en matemática aplicada de la Universidad de San Carlos, por ser un tema con muchas aplicaciones y un amplio contenido de resultados e investigación actual.

A. Geometría algebraica

Este apéndice es un pequeño recorrido a través de ciertos aspectos del mundo de la geometría algebraica. Nuestro resumen tiene un sabor clásico. El propósito principal es esbozar la motivación detrás de las diversas definiciones y de la teoría que recopilamos en este trabajo. Esta motivación se logrará al enunciar la fórmula de Riemann-Roch para curvas algebraicas. La geometría algebraica moderna persigue alcanzar generalidad absoluta, y al hacerlo la maquinaria técnica se torna pesada rápidamente. En este apéndice lidiaremos únicamente con variedades algebraicas de dimensión uno, es decir, curvas algebraicas. En general las variedades algebraicas se construyen mediante un proceso de pegado de variedades más simples, variedades afines. Cuando decimos que estas páginas tienen un sabor clásico, nos referimos en concreto a que únicamente consideraremos curvas afines y curvas proyectivas.

La primera sección define los espacios afines, los espacios proyectivos, y las curvas inmersas en esos espacios. La segunda sección define el campo de funciones de una curva algebraica y el orden con el cuál una función se desvanece en un punto. La tercera sección estudia sistemas lineales y enuncia la fórmula de Riemann-Roch. Lectores con conocimientos previos se pueden saltar alguna de las secciones y encontrarán al principio de cada sección la notación relevante que se ha definido con anterioridad. En este apéndice sea K un campo algebraicamente cerrado.

Para referencias la literatura sobre geometría algebraica es vasta. El libro usual de referencia, con un poco de geometría algebraica clásica, y gran parte de la geometría algebraica moderna es el libro de Hartshorne [22]. Una buena introducción a la geometría algebraica para principiantes es el libro de Fulton [19]. Su capítulo 8 contiene una exposición sólida y elemental sobre el teorema de Riemann-Roch para curvas algebraicas. Las ideas de la demostración que se da ahí sigue una demostración clásica dada por Alexander von Brill y Max Noether.

A.1. El espacio afín y el espacio proyectivo

En la teoría clásica una *variedad* es el conjunto de ceros de un sistema de polinomios sobre un campo K . Si uno pregunta en cuál espacio vive este conjunto de ceros, entonces la respuesta más simple es en un *espacio afín*. La construcción del espacio afín es sencilla, pero si uno intenta responder preguntas sobre intersecciones de variedades afines entonces las respuestas sugerirán que al espacio afín le faltan puntos, esto debido a que algunas intersecciones no suceden. Al agregar estos puntos se obtiene el espacio proyectivo.

Definimos el n -espacio afín $\mathbb{A}^n(K)$ como el producto cartesiano de K consigo mismo n veces. Escogemos esta notación para subrayar el hecho de que no vemos $\mathbb{A}^n(K)$ como un espacio vectorial. Escribimos \mathbb{A}^n por brevedad si se entiende cuál es el campo base. Un *conjunto algebraico* es un subconjunto de \mathbb{A}^n de la forma:

$$V(S) := \{P \in \mathbb{A}^n \mid F(P) = 0 \text{ para todo } F \in S\}$$

para S un subconjunto del anillo de polinomios $K[x_1, \dots, x_n]$.

Podemos definir una topología si tomamos como los cerrados a todos los conjuntos algebraicos. Este espacio topológico definido en \mathbb{A}^n es llamado la *topología de Zariski*. Recordemos que un conjunto cerrado es *irreducible* si no es igual a una unión finita de subconjuntos propios cerrados. Una *variedad afín* es un conjunto algebraico irreducible de \mathbb{A}^n . Algunas fuentes no incluyen el requerimiento de ser irreducible, y de esa manera borran la distinción entre conjuntos algebraicos y variedades afines.

Un tema de interés es la intersección de dos variedades. Preguntas de esta índole se conocen como *teoría de intersecciones*. Uno de los primeros resultados en esta dirección es el *teorema de Bézout*, el cuál da una formula para contar el número de intersecciones entre dos curvas bajo ciertas condiciones. Para poner un ejemplo específico, consideremos dos líneas distintas en el plano afín \mathbb{A}^2 , dadas por ecuaciones de la forma $y = ax + b$. Estas líneas se intersectan si los coeficientes de x son diferentes, caso contrario las líneas son paralelas y no tienen un punto común.

Por lo tanto las líneas paralelas forman un caso especial a considerar en cualquier teorema que hable sobre la intersección de dos rectas. Nos gustaría dar un tratamiento unificado, que los resultados no tengan que considerar casos especiales tales como el de las rectas paralelas. La manera como resolvemos este problema es pensando que estas intersecciones sí suceden pero están afuera del espacio afín. Es decir, al espacio afín le faltan puntos. Estos puntos son llamados *puntos al infinito*. En el caso del plano hay un punto por cada posible dirección que pueda tener una

familia de rectas paralelas, y entonces decimos que esta familia de paralelas se interseca en dicho punto al infinito. Agregar los puntos al infinito al espacio \mathbb{A}^n resulta en una construcción llamada el *espacio proyectivo* $\mathbb{P}^n(K)$. Definimos al espacio proyectivo n -dimensional $\mathbb{P}^n(K)$ como el conjunto de líneas que pasan por el origen del espacio vectorial $n + 1$ -dimensional K^{n+1} .

Ilustramos la definición para el caso $n = 2$ para mostrar que el plano proyectivo se puede ver como el plano afín con los puntos al infinito agregados, y que los puntos al infinito se encuentran en una recta llamada la *recta al infinito*. Sea $a : x_3 = 1$ un plano en K^3 que identificamos con \mathbb{A}^2 . Cada punto en a determina una línea única que pasa a través del origen de K^3 . Conversamente, una línea a través del origen tiene dos opciones. Están las rectas que intersectan al plano a , y entonces como esta intersección sucede en un punto único tenemos que el conjunto de estas líneas está en biyección con el plano afín \mathbb{A}^2 . En el otro caso están las rectas que no intersectan al plano a , lo cuál significa que están contenidas en el plano $x_3 = 0$. En el segundo caso vemos que son todas las líneas que atraviesan el origen de un plano que puede ser identificado con el espacio vectorial K^2 , es decir por definición este conjunto es la recta proyectiva $\mathbb{P}^1(K)$. Entonces conceptualizamos el plano proyectivo como la unión del plano afín con una copia de la recta proyectiva $\mathbb{P}^1(K)$, y a esta recta le llamamos la *recta al infinito*.

La recta proyectiva $\mathbb{P}^1(K)$ es la línea afín con un punto extra agregado, el punto al infinito ∞ . El lector puede intentar generalizar estos argumentos al espacio proyectivo n -dimensional $\mathbb{P}^n(K)$, es decir mostrar que se puede considerar $\mathbb{P}^n(K)$ como el espacio $\mathbb{A}^n(K)$ al que le agregamos el espacio $\mathbb{P}^{n-1}(K)$ en el infinito. Una línea a través del origen de K^{n+1} se puede representar de manera única con un elemento del cociente $(K^{n+1} - \vec{0}) / \sim$, donde $v_1 \sim v_2$ si existe un $\lambda \in K$ distinto de cero tal que $v_1 = \lambda v_2$. Con esto podemos construir *coordenadas homogéneas* $[z_0 : \dots : z_n]$ para $\mathbb{P}^n(K)$. Les decimos homogéneas porque cumplen que $[z_0 : \dots : z_n] = [\lambda z_0 : \dots : \lambda z_n]$ para cualquier $\lambda \in K$ no nulo. De esta forma cada punto del espacio proyectivo obtiene coordenadas que son únicas, salvo multiplicación de las coordenadas por un escalar no nulo. Los índices de las coordenadas empiezan en 0, así que en total tenemos $n + 1$ números para especificar las coordenadas.

Dado que el espacio proyectivo n -dimensional puede ser cubierto por un espacio afín junto con un espacio proyectivo de una dimensión menor, nos preguntamos si es posible cubrirlo únicamente con espacios afines. Sea U_i el complemento del conjunto $\{z_i = 0\}$. Entonces $\mathbb{P}^n(K) = U_0 \cup U_1 \cup \dots \cup U_n$. Los conjuntos U_i son conocidos como las *cartas afines* porque existen mapeos canónicos de U_i hacia el

espacio afín \mathbb{A}^n (de hecho estos mapeos son isomorfismos, pero nos saltaremos definir precisamente qué significa un isomorfismo en este contexto). El mapeo de U_i a \mathbb{A}^n envía $[z_0 : \dots : z_i : \dots : z_n] \in U_i$ al punto $(z_0/z_i, \dots, z_{i-1}/z_i, z_{i+1}/z_i, \dots, z_n/z_i) \in \mathbb{A}^n$. Esto está bien definido porque $z_i \neq 0$.

Debemos modificar ligeramente la definición de una variedad afín para poder obtener las variedades proyectivas, debido a que en general para un polinomio $f \in K[z_0, \dots, z_n]$ los valores de $f(z_0, z_1, \dots, z_n)$ y los valores de $f(\lambda z_0, \lambda z_1, \dots, \lambda z_n)$ son diferentes, así que estos polinomios no se pueden interpretar como funciones que van de $\mathbb{P}^n(K)$ a K .

El concepto de raíz, es decir que un punto P de un espacio proyectivo sea raíz de un polinomio, se puede recuperar si f es un *polinomio homogéneo*. Recordemos que el grado de un término $z_0^{d_0} \dots z_n^{d_n}$ es la suma de los exponentes, $\sum d_i$. Un polinomio homogéneo es una combinación lineal de términos del mismo grado. Si f es un polinomio homogéneo de grado d entonces $f(\lambda z_0, \lambda z_1, \dots, \lambda z_n) = \lambda^d f(z_0, z_1, \dots, z_n)$. Por lo tanto concluimos $f(\lambda z_0, \lambda z_1, \dots, \lambda z_n) = 0$, y vemos que las raíces de un polinomio están bien definidas para puntos proyectivos.

Un *conjunto algebraico proyectivo* es un subconjunto de $\mathbb{P}^n(K)$ de la forma $V(S)$, donde S es un conjunto de polinomios homogéneos. Nuevamente los subconjuntos algebraicos proyectivos definen una topología de Zariski, y definimos una *variedad proyectiva* como un conjunto algebraico proyectivo que cumpla con ser irreducible.

Ahora sabemos en dónde deben vivir nuestras curvas, y también sabemos que dichas curvas deben ser variedades. Para terminar de definir qué es una curva debemos formular el concepto de dimensión. Dada una variedad proyectiva X y un subconjunto propio Y tal que Y es una variedad proyectiva también, a Y le llamamos una *subvariedad* de X . La *dimensión* de X es la longitud máxima posible n de una secuencia de contenciones estrictas de subvariedades $Y_1 \subset Y_2 \subset \dots \subset Y_n \subset X$. Notamos que un conjunto con un punto es una variedad de dimensión cero. Las variedades proyectivas de dimensión 1 son llamadas *curvas proyectivas*. Así que las curvas proyectivas son variedades distintas a los puntos cuyas únicas subvariedades son puntos.

Como mencionamos antes, es posible construir curvas algebraicas de una manera más general, es decir curvas algebraicas que no son isomorfas a alguna curva proyectiva. En el teorema de Riemann-Roch vamos a imponer dos condiciones en las curvas, una que sea una curva no singular y la otra que sea una curva completa. Que no sea singular y que sea completa son propiedades análogas a la no

singularidad y a la compacidad en el análisis clásico. Bajo estas condiciones uno puede demostrar que toda curva algebraica (en el sentido moderno) es completa si y solo si es isomorfa a alguna curva proyectiva [22, capítulo II, proposición 6.7]. Los morfismos son mapeos entre curvas que preservan estructuras adicionales (inducen homomorfismos de su campo de funciones, a ser definido en la próxima sección) y los isomorfismos son mapeos uno a uno que preserva completamente la estructura. Por lo tanto considerar curvas inmersas en el espacio proyectivo es suficiente para nuestros fines.

A.2. Funciones racionales

Notamos con $\mathbb{A}^n(K)$ y $\mathbb{P}^n(K)$ el espacio n -dimensionales afín y proyectivo, respectivamente, sobre el campo K .

En el contexto clásico la fórmula de Riemann-Roch es un resultado relacionado con los espacios de funciones racionales definidos en una superficie de Riemann (que en geometría algebraica se piensa como una curva algebraica sobre \mathbb{C} , con las condiciones extra de no tener singularidades y ser completo). La fórmula de Riemann-Roch cuenta la dimensión del espacio vectorial de funciones racionales cuando uno prescribe en dónde se encuentran los polos y el orden de los mismos. Definir una función racional en una variedad afín es fácil, hacerlo para una variedad proyectiva es más difícil. Presentamos dos formas con las que uno puede conceptualizar las funciones racionales en una variedad proyectiva. El lector que desee tener más intuición respecto a este tema puede trabajar los detalles de cómo se hace para variedades afines [19, capítulo 2].

Sea $X \subset \mathbb{P}^n(K)$ una variedad proyectiva. La manera abstracta de definir una función racional en una variedad es emular la situación clásica del análisis complejo. Una función racional en una superficie de Riemann se define localmente como una función meromorfa en cada vecindad abierta, es decir que podemos expresar a la función como un cociente de polinomios dados en alguna carta de coordenadas, y luego el objeto global se obtiene pegando la información local.

Una función racional f en una variedad proyectiva X está definida en un conjunto abierto $U \subset X$ si puede ser expresada como un cociente g/h con g, h polinomios homogéneos del mismo grado, y que h no se desvanezca en ningún punto de U . Una función racional global pega una familia de funciones racionales definidas sobre abiertos de tal forma que las funciones coinciden en las intersecciones de sus dominios abiertos. Queremos remarcar que todo conjunto abierto es denso en la to-

pología de Zariski, así que estar definido sobre un conjunto abierto nos da bastante información sobre la función.

Ahora para poder conceptualizar las funciones de una manera más concreta, que no requiera un proceso de pegado, necesitamos el anillo homogéneo de coordenadas de X . Sea $I(X)$ el ideal de polinomios homogéneos de $K[z_0, \dots, z_n]$ que se desvanece en X . El anillo de coordenadas homogéneo de X se define como $S(X) := K[z_0, \dots, z_n]/I(X)$. Lo que hace este cociente es identificar los polinomios cuya diferencia se desvanece en los puntos de X . Ahora el campo de funciones racionales $K(X)$ de X es el conjunto de elementos de la forma f/g con $f, g \in S(X)$, g no es idénticamente cero, y los grados de f y de g son los mismos. Las operaciones del campo son la adición y multiplicación usual de una estructura de campo de fracciones. Las dos descripciones que hemos dado para el campo de funciones resultan en objetos distintos, pero las estructuras son isomorfas [22, capítulo I, teorema 3.4].

Sea C una curva proyectiva y $f \in K(C)^*$, una función racional diferente de cero. En análisis complejo uno puede definir el orden con el cuál se desvanece una función racional en un punto. Para hacer lo mismo en geometría algebraica uno utiliza los anillos locales que resultan de localizar en un punto $P \in C$. Bajo ciertas condiciones estos anillos vienen con una valuación discreta natural, y tomamos el valor de esta valuación discreta como el orden con el cuál la función f se desvanece en P . Nos saltamos la construcción debido a que requiere una cantidad considerable de álgebra conmutativa. En su lugar listamos varias de las propiedades deseadas que tal construcción tiene. Sea $f, g \in K(C)$, y $P \in C$. Queremos una función $v_P : K(C) \rightarrow \mathbb{Z} \cup \{\infty\}$ que no sea idénticamente cero y que satisfaga:

- (a) $v_P(f) = \infty$ si y sólo si $f = 0$.
- (b) $v_P(f + g) \geq \min(v_P(f), v_P(g))$.
- (c) $v_P(fg) = v_P(f) + v_P(g)$.
- (d) Para una función fija $f \in K(C)$, $v_P(f)$ es diferente de cero para una cantidad finita de puntos $P \in C$.
- (e) Para una función fija $f \in K(C)$, $\sum_{P \in C} v_P(f) = 0$.

Una función con imagen $\mathbb{Z} \cup \{\infty\}$ y que satisfaga las propiedades (a) a la (c) se llama una *valuación discreta*. Ser capaz de construir la valuación v_P es equivalente a que el punto P sea *no singular*, también llamado *regular* [22, Capítulo I, Teorema

6.2A]. Así que necesitamos que todos los puntos de C sean no singulares, es decir que C sea una curva *no singular*, también conocido como *regular* o *suave*.

La propiedad (e) tiene su análogo en análisis con el hecho de que la suma de los polos y las raíces en una superficie de Riemann compacta es igual a cero. El análogo de compacidad es que la curva sea *completa*. Formalmente esta condición se enuncia como una propiedad que ciertos morfismos de la curva deben tener. En estas notas sólo necesitamos saber que si una curva C es suave y compacta entonces hay una valuación discreta v_P en todos los puntos de C que satisface las propiedades (a)-(e). Como hemos dicho ya, estas dos condiciones equivalen a pedir que C sea una curva proyectiva no singular.

A.3. Sistemas lineales de divisores

Sea C una curva proyectiva no singular, y sea $v_P : K(C) \rightarrow \mathbb{Z}$ la valuación en el punto $P \in C$, que satisfaga las propiedades (a)-(e) de la sección anterior.

Los divisores lineales son objetos que se utilizan para codificar información acerca de la curva y sus funciones racionales. En particular podemos usar los divisores para estudiar los ceros y los polos de las funciones racionales, para estudiar la intersección entre dos curvas, e incluso hay generalizaciones a variedades con dimensiones más altas.

Definimos el grupo de divisores $\text{Div}(C)$ de C como el grupo abeliano libre sobre los puntos de C , es decir, los elementos de $\text{Div}(C)$ son combinaciones lineales enteras de la forma $\sum_{P \in C} D(P)P$, donde $D(P)$ son enteros y sólo una cantidad finita de coeficientes son distintos de cero. Para los divisores $E, D \in \text{Div}(C)$, decimos que E domina a D , $E \geq D$, si $E(P) \geq D(P)$ para todo punto $P \in C$. Un divisor $E \in \text{Div}(C)$ es efectivo si $E \geq 0$. El grado de un divisor es la suma de los coeficientes, $\text{deg } D = \sum_{P \in C} D(P)$. Denotamos con $\text{Div}^k(C)$ al conjunto de todos los divisores de grado k .

A una función $f \in K(C)$ le asociamos un divisor de la siguiente forma:

$$\text{div}(f) = \sum_{P \in C} v_P(f)P.$$

De la discusión previa sobre las propiedades de v_P uno tiene que por (d) el divisor asociado $\text{div}(f)$ es un elemento de $\text{Div}(C)$. También el lector puede demostrar que por (c), $\text{div}(fg) = \text{div}(f) + \text{div}(g)$, y esto implica que el conjunto de divisores $\text{div}(f)$ es un grupo abeliano denotado por $\text{Prin}(C)$ y llamado el grupo de

divisores principales de C . Notamos que $\deg \operatorname{div}(f) = 0$ por (e), así que $\operatorname{Prin}(C)$ es un subgrupo de $\operatorname{Div}^0(C)$, del grupo de divisores de grado 0. Al grupo cociente $\operatorname{Cl}(C) := \operatorname{Div}(C)/\operatorname{Prin}(C)$ le llamamos el grupo de clases de C . Dado que los elementos de $\operatorname{Prin}(C)$ son de grado 0 entonces cada elemento de $\operatorname{Cl}(C)$ tiene un grado bien definido.

Ahora estamos interesados en controlar los polos de las funciones racionales. Escogemos un conjunto finito de puntos, y para cada punto un entero que representa el orden del polo en ese punto. Esta elección puede ser representada con un divisor D . Por lo tanto uno busca funciones racionales cuyos polos en los puntos especificados tengan un orden que no sea mayor a los enteros que hemos elegido, es decir se considera el conjunto:

$$L(D) = \{f \in K(C) \mid \operatorname{div}(f) \geq -D\} \cup \{0\}$$

Este conjunto se conoce como el *espacio de Riemann*. Notamos que $L(D)$ tiene una estructura natural de espacio vectorial sobre el campo K . De hecho es un subespacio finito dimensional del espacio vectorial $K(C)$. El valor $\ell(D)$ de esta dimensión es una medida de *cuán grande* o *cuán pequeños* estos espacios son. En estas líneas la fórmula de Riemann-Roch es uno de los resultados más importantes de la teoría clásica. La contribución de Bernhard Riemann es en la forma de una desigualdad, la contribución Gustav Roch fue agregar un *término de corrección* para hacerla una igualdad. Vamos a enunciar ambos resultados.

Teorema A.1 (El Teorema de Riemann para curvas algebraicas). *Dada C una curva proyectiva no singular, existe un entero g tal que para todo $D \in \operatorname{Div}(C)$ se tiene que $\ell(D) \geq \deg(D) + 1 - g$. Al entero más pequeño que cumple esa propiedad se le llama el género de C .*

Para poder agregar el término de corrección a la desigualdad de Riemann requerimos construir una clase de divisores conocida como la *clase de divisores canónicos* $K_C \in \operatorname{Div}(C)$. Nuevamente nos saltamos los detalles y esbozamos la idea. Uno tiene que trasladar el concepto de formas diferenciales del análisis hacia la geometría algebraica. Para poder hacer esto, uno trata a las formas diferenciales como símbolos formales df , con $f \in K(C)$ cualquier función racional. Requerimos que estos símbolos satisfagan varias reglas (las reglas usuales que los diferenciales deben satisfacer en análisis, la más importante siendo la regla de Leibniz).

Luego uno tiene que definir el orden de desvanecimiento de un diferencial en un punto P mediante una valuación, y asociarle un divisor a df de una manera similar

a como uno asocia un divisor a f . Esto se puede hacer de una manera consistente porque los diferenciales en curvas algebraicas forman un espacio vectorial sobre $K(C)$ de dimensión 1. Una consecuencia de esto es que cualquiera dos divisores que vengan de diferenciales van a ser equivalentes. Entonces hay una única clase de equivalencia de divisores canónicos, la clase de divisores canónicos. Usamos K_C para denotar la clase o para denotar un representante, dependiendo del contexto.

Teorema A.2 (Riemann-Roch para curvas algebraicas). *Sea K un campo algebraicamente cerrado. Sea C una curva proyectiva no singular sobre K . Sea g el género de C determinado como en el teorema de Riemann. Sea K_C un divisor canónico para C . Para cada divisor $D \in \text{Div}(C)$ se cumple la fórmula de Riemann-Roch:*

$$\ell(D) - \ell(K_C - D) = \deg(D) + 1 - g.$$

No damos una demostración para Riemann-Roch, pero sí vamos a enunciar varias observaciones e identidades que el lector podría usar para demostrar la fórmula.

- (I) (Ejercicio 8.13 de [19]) Para un divisor $D \in \text{Div}(C)$, se cumple la siguiente desigualdad

$$\ell(D - P) \geq \ell(D) - 1.$$

La igualdad se da para la mayoría de puntos $P \in C$, sólo hay un número finito de excepciones.

- (II) (El lema de reducción de Noether, sección 8.6 de [19]) Si $\ell(D) > 0$ y $\ell(K_C - D - P) \neq \ell(K_C - D)$, entonces $\ell(D + P) = \ell(D)$.

Por último en el transcurso de este proyecto trabajamos con grafos, ahí no podemos construir un objeto totalmente análogo a $L(D)$ porque no tenemos la estructura de espacio vectorial sobre K . Necesitamos una forma indirecta de medir $L(D)$, una forma más combinatoria. También tenemos que buscar un procedimiento que no le asigne tanta importancia a las funciones $f \in K(C)$. Notamos que en la definición de $L(D)$, buscamos funciones f que satisfagan $D \geq -\text{div}(f)$, es decir $D + \text{div}(f) \geq 0$. El divisor efectivo $E = D + \text{div}(f)$ es equivalente a D en las clases de equivalencia de $\text{Cl}(C)$. Entonces una alternativa a considerar el conjunto $L(D)$, el cuál es un espacio de funciones, es considerar el *sistema lineal* de D , dado por $|D| = \{E \in \text{Div}(G) \mid E \geq 0, E \sim D\}$, es decir que consideramos todos los divisores efectivos equivalentes a D .

Medimos entonces el tamaño de $|D|$. Las identidades utilizadas para demostrar la fórmula de Riemann-Roch sugieren una caracterización en términos de cuántos

elementos de X se le pueden sustraer a D y aún así obtener un espacio de Riemann no trivial, es decir, cuál es el grado k más alto tal que para cualquier divisor efectivo $E \in \text{Div}^k(C)$ el conjunto $|D - E|$ es no vacío.

Definición A.3. Definimos el rango de un divisor $D \in \text{Div}(C)$ como una función $r : \text{Div}(C) \rightarrow \{-1, 0, 1, 2, \dots\}$. Para un divisor $D \in \text{Div}(C)$, el rango $r(D)$ es el entero más pequeño tal que existe un divisor efectivo E de grado $r(D) + 1$ tal que $|D - E| = \emptyset$.

Esta caracterización combinatoria es el corazón de la fórmula de Riemann-Roch para grafos. Para enlazarlo con el valor de $\ell(D)$, se puede demostrar:

Teorema A.4 (Lema 2.4 en [4]). *Para cada divisor $D \in \text{Div}(C)$ tenemos $r(D) = \ell(D) - 1$.*

Entonces en curvas algebraicas uno puede calcular, ya sea $\ell(D)$ o $r(D)$, y luego relacionarlos. En el caso de grafos finitos no podemos definir un análogo de $\ell(D)$, pero sí uno de $r(D)$ por lo que lo tomamos como nuestro concepto a estudiar al momento de emular toda la teoría expuesta en este apéndice.

B. Código fuente

Para este proyecto utilicé un modulo en Perl llamado `ChipFiring.pm` escrito con el propósito de implementar varios algoritmos de la manera más eficiente posible. El núcleo principal de este modulo es una función para encontrar todos los divisores de rango al menos 1 de un grafo G . El modulo también incluye una función para generar grafos aleatorios, y demás código usado para intentar encontrar contraejemplos, o evidencia que apoyara, conjeturas y resultados que se fueron dando. Documentación, varios programas de ejemplo, y un programa interactivo `chipfiring.pl` se pueden encontrar en <https://github.com/AV-2/chipfiring>.

A continuación se reproduce una versión del código fuente del módulo sin comentarios.

```
package ChipFiring;
use v5.20;
use strict;
use warnings;

use Math::MatrixReal;
use Data::Dumper;
use Math::Prime::Util qw/factor_exp gcd/;
use Carp;
use Math::Round qw/round/;
use POSIX qw/ceil floor/;
use DBI;
use Time::HiRes qw(gettimeofday tv_interval);

use Class::Tiny qw/
  n m g c
  matrix
  matrix_string
  edges
  neighbors
  links_string
  degrees
  lap
  support
  support_lap
  vertices
  vertices_sorted
  det
  Li
  q
  ginverse
  rankl
  rankl_string
  signature
  rankl_db_string
  prime_string
  rankl_divisors
```

```

empty_divisor
reduced_divisors
/, { signature => '' };

our $MAX_ATTEMPTS = 10;
our $n;
our @mat;
our @lap;
our @support;
our @support_lap;
our @vertices;
our @vertices_sorted;
our @empty_divisor;
our @degrees;
our $c;
our $dbh;
our $table_name;
our @Li;
our $det;

sub _process_graph {

    my $self = shift();

    @mat = @({ $self->{matrix} });
    $self->{n} = scalar @mat if ( !$self->{n} );
    $self->{vertices} = [ ( 0 .. $self->{n} - 1 ) ];
    $self->{empty_divisor} = [ (0) x $self->{n} ];
    $self->{support} = [
        map {
            my @row = @{$$_};
            [ grep { $row[$_] } @{$ $self->{vertices} } ]
        } @mat
    ];
    if ( !$self->{degrees} ) {
        $self->{degrees} = [];
        foreach my $i ( @{$ $self->{vertices} } ) {
            $self->{degrees}[$i] += $mat[$i][$_]
                foreach ( @{$ $self->{support}[$i] } );
        }
    }

    $self->{m} = 0;
    $self->{m} += $_ foreach ( @{$ $self->{degrees} } );
    $self->{m} /= 2;
    $self->{g} = $self->{m} - $self->{n} + 1;
    $self->{c} = $self->{g} / 2 + 1;

    $self->{vertices_sorted} =
        [ sort { -1 * ( $self->{degrees}[$a] <=> $self->{degrees}[$b] ) }
            @{$ $self->{vertices} } ];

    for my $i ( @{$ $self->{vertices} } ) {
        $self->{lap}[$i] = [ map { -$_ } @{$ $mat[$i] } ];
        $self->{lap}[$i][$i] = $self->{degrees}[$i];
    }
    $self->{support_lap} = [
        map {
            my @row = @{$$_};
            [ grep { $row[$_] } @{$ $self->{vertices} } ]
        } @{$ $self->{lap} }
    ];

    $self->{matrix_string} = _matrix2string( @{$ $self->{'matrix'} } );

    $self->_matrix2links();
    $self->{links_string} = _links2string( @{$ $self->{edges} } );
}

sub _matrix2links {

    my $self = shift();
    $self->_context();
}

```

```

my @edges;
my @neighbors;
push @neighbors, [] for ( 1 .. $n );

my $i = 0;
for my $vertex ( @vertices ) {
    for my $j ( $vertex + 1 .. $n - 1 ) {
        if ( $mat[$vertex][$j] > 0 ) {
            my @links;
            push @links, [ $i++, $vertex, $j, 0, 0 ]
                for ( 1 .. $mat[$vertex][$j] );

            push @edges, @links;
            push @{$neighbors[$vertex]}, @links;
            push @{$neighbors[$j]}, @links;
        }
    }
}

$self->{edges} = \@edges;
$self->{neighbors} = \@neighbors;

_clean_context();
}

sub from_matrix {
my $self = shift;
$self->{'matrix'} = shift;

$self->_process_graph();
}

sub from_links {
my $self = shift;
my @links = @_;

$links[$_] =~ s/\n/ /g foreach ( 0 .. $#links );

$self->from_matrix( _links2matrix(@links) );

return $self;
}

sub from_links_file {
my $self = shift;
my $file = shift;

open my $fh, '<', $file or croak("The_provided_file_is_inexistent");

my $link_string;
while (<$fh>) {
    chomp($link_string);
    $link_string .= $_;
}
$self->from_links($link_string);

return $self;
}

sub from_matrix_file {
my $self = shift;
my $file = shift;

open my $fh, '<', $file or croak("The_provided_file_is_inexistent");

my $matrix_string;
while (<$fh>) {
    chomp($matrix_string);
    $matrix_string .= $_ . "\n";
}
$self->from_matrix_string($matrix_string);

return $self;
}

```

```

sub from_matrix_string {
    my $self = shift;

    $self->from_matrix( _string2matrix( $_[0] ) );
}

sub random_graph {
    my $self = shift;
    my $n     = $self->{n} = shift;
    my $g     = $self->{g} = shift;
    my $method = shift || 1;

    my $i = 0;
    while (1) {
        randomgraph12( $self, $n, $g, 1 ) if ( $method == 1 );

        last if ( $self->is_connected() );
        croak(
            "The requested parameters made us reach the number of Max Attempts before finding a connected
            graph. Probably v is very high in relation to g, consider raising the variable ChipFiring::
            MAX_ATTEMPTS if you want me to try harder to find a graph, or modifying the random_graph
            generation method. Default value ten attempts. Attempted $i times."
        ) if ( ++$i >= $MAX_ATTEMPTS );
    }

    return $self;
}

sub randomgraph12 {
    my $self = shift();
    my $n     = shift;
    my $g     = shift;
    my $method = shift;
    my $m     = $self->{m} = $g + $n - 1;
    my ( @mat, $r1, $r2, $i1, $i2 );

    my $min_connectivity = $method;

    my @to_connect = ( 0 .. $n - 1 );
    my @degrees     = (0) x $n;
    push @mat, [ (0) x $n ] for ( 1 .. $n );

    for ( 1 .. $m ) {
        if ( @to_connect > 1 ) {
            my $n = @to_connect;

            $r1 = int( rand($n) );
            $r2 = $r1;
            $r2 = int( rand($n) ) while ( $r1 == $r2 );

            ( $r1, $r2 ) = ( $r2, $r1 ) if ( $r1 < $r2 );

            $i1 = $to_connect[$r1];
            $i2 = $to_connect[$r2];

            $mat[$i1][$i2] += 1;
            $mat[$i2][$i1] += 1;
            $degrees[$i1]++;
            $degrees[$i2]++;

            splice @to_connect, $r1, 1 if ( $degrees[$i1] > $method );
            splice @to_connect, $r2, 1 if ( $degrees[$i2] > $method );
        }
        elsif ( @to_connect == 1 ) {
            $r1 = $to_connect[0];
            $r2 = $r1;
            $r2 = int( rand($n) ) while ( $r1 == $r2 );

            $mat[$r1][$r2] += 1;
            $mat[$r2][$r1] += 1;
        }
    }
}

```



```

        $degrees[$r1]++;
        $degrees[$r2]++;

        @to_connect = () if ( $degrees[$r1] > $method );
    }

    else {
        $r1 = int( rand($n) );
        $r2 = $r1;
        $r2 = int( rand($n) ) while ( $r1 == $r2 );

        $mat[$r1][$r2] += 1;
        $mat[$r2][$r1] += 1;

        $degrees[$r1]++;
        $degrees[$r2]++;
    }
}

$self->{'matrix'} = \@mat;
$self->{'degrees'} = \@degrees;

$self->_process_graph();

return $self;
}

sub subdivide {

    my $self = shift;
    my $a     = shift;

    my $string;
    my $i = $self->{n};

    foreach my $edge ( @{$self->{edges}} ) {
        $string .= "$edge->[1],$i\n";

        if ( $a > 1 ) {
            foreach ( 2 .. $a ) {
                $i++;
                $string .= $i - 1 . ",$i\n";
            }
        }

        $string .= "$i,$edge->[2]\n";
        $i++;
    }

    return $string;
}

sub subdivide_edge {

}

sub all_rank1 {
    my $self = shift;
    my ( $div_str, $rank1, $rank2 ) = ( '', 0, 0 );
    $self->_context();

    my $divisors = _all_divisors_iterator1( $c - 1 );
    $self->{rank1} = $divisors->{rank1};
    $self->{rank1_divisors} = [ map { $_->[0] } @{$divisors->{rank1}} ];
    $self->{rank1_string} = join "\n", map {
        $rank1++ if ( $_->[1] == 1 );
        $rank2++ if ( $_->[1] > 1 );
        $_->[0]
    } @{$divisors->{rank1}};
    $self->{time_interval} = $divisors->{time_interval};
    $self->{reduced_divisors} = $divisors->{reduced_divisors};

    my $link_string = $self->{links_string};

```

```

$link_string =~ s/\n//g;
$self->{det} = round(
    abs( Math::MatrixReal->new_from_cols( \@lap )->minor( 1, 1 )->det() ) )
    unless ( $self->{det} );
$self->{prime_string} = join '*',
    map { join '^', @{$_} } factor_exp( $self->{det} );

$self->{rank1_db_string} = sprintf(
    "( '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s' )",
    $link_string,           $self->{rank1_string},
    $rank1,                 $rank2,
    $rank1 + $rank2,       $self->{reduced_divisors},
    $self->{g},              $self->{n},
    $self->signature(),     $self->{det},
    $self->{prime_string},  $self->{time_interval}
);

_clean_context();

return $divisors;
}

sub all_rank1_energy {
my $self = shift;
my $div_str = shift;
$self->_context();

my $c = $c - 1;

my $s = $n + $c - 1;
my $site = [];
push( @$site, $_ ) for ( 1 .. $c );

push( @$site, (0) x ( $s - $c ) );
my $last = $c - 1;
my $previous_failure = 0;
my $div_last = 0;
my $is_finished;
my @div = _convert_div($site);
my $burns;

my @divisors;
my %R1;
my %R2;
my @reps;
my $reduced_divisors;

my $t0 = [gettimeofday];

while (1) {
my @result;

for my $i ( 0 .. $n - 1 ) {
if ( $div[$i] > 0 ) {
$result[$_] += ( $div[$i] * $Li[$i][$_] )
foreach ( 0 .. $n - 1 );
}
}
$result[$_] = $result[$_] % $det foreach ( 0 .. $n - 1 );

my $result = join " ", @result;

push @reps, \@result if ( !defined( $R1{$result} ) );

$R1{$result} = \@div;

last if ( $is_finished );

if ( $last < ( $s - 1 ) ) {
$site->[$last] = 0;
$last++;
$site->[$last] = $last + 1;
}
else {

```

```

my $sum = 0;
for ( 1 .. $c ) {
    last if !$site->[ $s - $_ ];
    $sum++;
}

my $next;
for ( 0 .. ( $last - $sum ) ) {
    $next = $_ if $site->[$_];
}
$site->[$next] = 0;
$site->[ $next + 1 ] = $next + 2;

$site->[ $last + 1 - $_ ] = 0 for ( 1 .. $sum );
$site->[ $next + 1 + $_ ] = $next + 2 + $_ for ( 1 .. $sum );

$last = $next + 1 + $sum;
}
@div = _convert_div($site);
$is_finished = $div[-1] == $c;
}

for my $i ( 1 .. $n - 1 ) {
my %intersection;

foreach my $class (@reps) {
my @class = @$class;
my @result;
$result[$_] = ( $class[$_] + $Li[$i][$_] ) % $det
foreach ( 0 .. $n - 1 );
my $string = join " ", @result;
$intersection{$string} = $R1{$string}
if ( defined( $R1{$string} ) );
}
%R1 = %intersection;
}

say scalar keys %R1, "\t", tv_interval($t0);

_clean_context();
}

sub divisor_energy {
}

sub all_equivalent {
my $self = shift;
my $div_str = shift;
$self->_context();

$div_str = array2div( _reduce( 0, [ _div2array($div_str) ] ) );
return _all_divisors_iterator2( $c, \&ChipFiring::_equivalent_to_reduced,
    $div_str, 0 );

_clean_context();
}

sub _equivalent_to_reduced {
my @div = @{ $_[0] };
my $div = $_[1];
my $q = $_[2];

my $div_string = array2div(@div);
my $candidate_div = array2div( _reduce( $q, \@div ) );
my $rank;

return $div_string if ( $div eq $candidate_div );
return 0;
}

```

```

sub rank {
  my $self = shift;
  $self->_context();

  my $rank = _rank( [ _div2array( shift() ) ] )->[0];

  _clean_context();

  return $rank;
}

sub reduce {
  my $self = shift;
  my $q = shift;
  $self->_context();

  my $reduce = array2div( _reduce( $q, [ _div2array( shift() ) ] ) );

  _clean_context();

  return $reduce;
}

sub reduce_trace {
  my $self = shift;
  my $q = shift;
  $self->_context();

  my $steps = ( _reduce_footprint( $q, [ _div2array( shift() ) ], 1 ) ) [2];

  _clean_context();

  return $steps;
}

sub _rank {
  my @div = @ { shift() };
  my @footprint = map { $_ > 0 } @div;
  my $rank = 2;
  my $q = 0;

  foreach my $i (@vertices_sorted) {
    next if ( $footprint[$i] );
    @_ = _reduce_footprint_stop( $i, \@div );
    @div = @ { $_[0] };
    my @newfootprint = @ { $_[1] };
    @footprint =
      map { $footprint[$_] > 0 || $newfootprint[$_] } ( 0 .. $#div );
    $rank = ( $rank > $div[$i] ) ? $div[$i] : $rank;
    unless ($rank) {
      $q = $i;
      last;
    }
  }
  return [ $rank, $q ];
}

sub _reduce {
  my $q = shift;
  my @div = @ { shift() };
  my $keep_intermediate_steps = shift;

  my @intermediate_steps;

  while ( my $burnt = _burn( $q, \@div ) ) {
    push @intermediate_steps, [ @div ] if ($keep_intermediate_steps);
  }

  return \@intermediate_steps if ($keep_intermediate_steps);
  return @div;
}

```

```

sub _reduce_footprint {

    my $q          = shift;
    my @div        = @{ shift() };
    my @footprint  = map { $_ > 0 } @div;
    my $keep_track_intermediate = shift;
    my @intermediate;

    while ( my $burnt = _burn( $q, \@div ) ) {
        @footprint = map { $footprint[$_] > 0 || $div[$_] } ( 0 .. $#div );
        push @intermediate, array2div(@div) if ( $keep_track_intermediate );
    }

    return \@div, \@footprint, \@intermediate;
}

sub _reduce_footprint_stop {

    my $q          = shift;
    my @div        = @{ shift() };
    my @footprint  = map { $_ > 0 } @div;

    while ( my $burnt = _burn( $q, \@div ) ) {
        last if ( $div[$q] );
        @footprint = map { $footprint[$_] > 0 || $div[$_] } ( 0 .. $#div );
    }

    return \@div, \@footprint;
}

sub is_connected() {

    my $self = shift;

    $self->_context();
    my $burns = !_burn( 0, $self->{empty_divisor} );
    _clean_context();

    return $burns;
}

sub _burn {

    no warnings 'uninitialized';

    my $q = shift;
    my $div = shift();
    my ( @burnt, @is_burnt, @burning, @fires );
    @burnt      = ($q);
    @burning    = ($q);
    @fires      = @{$div};
    $fires[$q]  = -1;
    $is_burnt[$q] = 1;

    while ( scalar @burning > 0 ) {
        foreach my $i ( @burning ) {
            $fires[$_] = $fires[$_] + $lap[$i][$_]
                foreach ( @{ $support_lap[$i] } );
        }
        @burning = grep {
            push @burnt, $_
                if ( !$is_burnt[$_] && $fires[$_] < 0 && !$is_burnt[$_]++ )
            ;
        } map { @{ $support_lap[$_] } } @burning;
    }

    if ( scalar @burnt == $n ) {
        return 0;
    }
    else {
        $fires[$q] += 1 + $div->[$q];
        @{$div} = @fires;
        return ( \@burnt );
    }
}

```

```

}

sub neighborhood {

    return @{ $_[0]->{support}->[ $_[1] ] };
}

sub div_approximate {

    my $self = shift;
    $self->_context();
    my $div = shift;
    my $a = shift;

    my @div_array = _div2array($div);

    $div =~ s/\d*v//;
    my @div = split / \d+v/, $div;
    @div = grep { $_ >= $a } @div;
    my @approx = ( \@div );

    foreach my $i ( 0 .. $#div ) {
        next if ( $div[$i] < $a );

        $div_array[ $div[$i] ] = undef;
        @approx = map {
            my @copy = @{$_};
            map { $copy[$i] = $_; [ @copy ] } $self->neighborhood( $copy[$i] )
        } @approx;
    }

    @approx = map {
        my @copy_div_array = @div_array;
        $copy_div_array[$_]++ foreach @{$_};
        array2div(@copy_div_array)
    } @approx;

    return @approx;
}

sub some_equivalent {

    my $self = shift;
    $self->_context();
    my $div = shift;
    my @reduce = ( $_[0] ) ? @{$ $_[0] } : @vertices;
    my @equiv;

}

sub calculate_Li {
    my $self = shift;
    $self->_context();
    my $q = shift;
    $q++;

    my $mat = Math::MatrixReal->new_from_rows( $self->{lap} )->minor( $q, $q );

    my $det = round( $mat->det() );
    $mat = $mat / $det;
    $mat = $mat->inverse();
    my @Li;
    $Li[0][$_] = 0 foreach ( 0 .. $self->{n} - 1 );

    foreach my $i ( 1 .. $self->{n} - 1 ) {
        $Li[$i][0] = 0;
        $Li[$i][$_] = round( $mat->element( $i, $_ ) )
            foreach ( 1 .. $self->{n} - 1 );
    }

    $self->{det} = $det;
    $self->{Li} = \@Li;
    $self->{ginverses}{$q} = \@Li;
}

```

```

sub _div2array {
    my @div = @empty_divisor;

    foreach ( split( '␣', $_[0] ) ) {
        @_ = split 'v';
        $div[ $_[1] ] += $_[0];
    }

    return @div;
}

sub array2div {
    my @div;
    for (@vertices) {
        push @div, $_[$_] . 'v' . $_[$_] if ( $_[$_] );
    }

    return join( '␣', @div );
}

sub _convert_div {
    my $ite = shift;
    my @div = @empty_divisor;
    my $j = 0;
    for my $i (@$ite) {
        if ($i) {
            $div[$j]++;
        }
        else {
            $j++;
        }
    }
    return @div;
}

sub _matrix2string {
    return join "\n", map { join "\t", @{$$_} } @_;
}

sub _string2matrix {
    return [ map { [ split "\t" ] } split "\n", $_[0] ];
}

sub _links2string {
    return join "\n", map { "$_->[1],$_->[2]_" } @_;
}

sub _links2matrix {
    my @mat;
    my ( $n, $i, $j );
    $n = 0;

    map {
        ( $i, $j ) = split ' ', $_;
        $mat[$i][$j]++;
        $mat[$j][$i]++;
        $n = $j if ( $n < $j );
        $n = $i if ( $n < $i );
    }
    split '␣'
    foreach (@_);

    foreach my $ref (@mat) {
        @$ref = map { ( $ref->[$_] ) ? $ref->[$_] : 0 } ( 0 .. $n );
    }

    return \@mat;
}

sub _all_divisors_iterator1 {
    my $c = shift();

```

```

my $s = $n + $c - 1;
my $site = [];
push( @$site, $_ ) for ( 1 .. $c );

push( @$site, (0) x ( $s - $c ) );
my $last = $c - 1;
my $previous_failure = 0;
my $div_last = 0;
my $is_finished;
my $burns = 1;

my @div = _convert_div($site);
$div[0] += 1;

my @divisors;
my $reduced_divisors;

my $t0 = [gettimeofday];

while (1) {
    if ( $burns and !_burn( 0, \@div ) ) {
        $reduced_divisors++;

        my $rank = _rank( \@div );
        if ( $rank->[0] > 0 ) {
            my $div_string = array2div(@div);
            push @divisors, [ $div_string, $rank->[0] ];
        }
    }
    else { $burns = 1 }

    last if ( $is_finished );
    if ( $last < ( $s - 1 ) ) {
        $site->[$last] = 0;
        $last++;
        $site->[$last] = $last + 1;
    }
    else {
        my $sum = 0;
        for ( 1 .. $c ) {
            last if !$site->[ $s - $_ ];
            $sum++;
        }

        my $next;
        for ( 0 .. ( $last - $sum ) ) {
            $next = $_ if $site->[$_];
        }
        $site->[$next] = 0;
        $site->[ $next + 1 ] = $next + 2;

        $site->[ $last + 1 - $_ ] = 0 for ( 1 .. $sum );
        $site->[ $next + 1 + $_ ] = $next + 2 + $_ for ( 1 .. $sum );

        $last = $next + 1 + $sum;
    }
    @div = _convert_div($site);
    $is_finished = $div[-1] == $c;

    $div[0] += 1;
    $div_last = $last - $c + 1;
    next
    if ( $div[$previous_failure] >= $degrees[$previous_failure]
        and $previous_failure != 0
        and !( $burns = 0 ) );
    if ( $div[$div_last] >= $degrees[$div_last]
        and $div_last != 0
        and !( $burns = 0 ) )
    {
        $previous_failure = $div_last;
        next;
    }
}

```



```

print "l", "\t", scalar @divisors, "\t", tv_interval($t0), "\t";
return {
    rankl           => \@divisors,
    time_interval  => tv_interval($t0),
    reduced_divisors => $reduced_divisors
};
}

sub _all_divisors_iterator2 {

    my $c          = shift;
    my $test       = shift;
    my @arguments  = @_;
    my $s          = $n + $c - 1;
    my $ite        = [];
    push( @$ite, $ _ ) for ( 1 .. $c );

    push( @$ite, (0) x ( $s - $c ) );
    my $last       = $c - 1;
    my $previous_failure = 0;
    my $div_last   = 0;
    my $is_finished;
    my @div = _convert_div($ite);
    my $burns;

    my @divisors;
    my $reduced_divisors;

    my $t0 = [gettimeofday];

    while (1) {
        push @divisors, $ _ if ( $ _ = &$test( \@div, @arguments ) );
        last if ( $is_finished );

        if ( $last < ( $s - 1 ) ) {
            $ite->[$last] = 0;
            $last++;
            $ite->[$last] = $last + 1;
        }
        else {
            my $sum = 0;
            for ( 1 .. $c ) {
                last if !$ite->[ $s - $ _ ];
                $sum++;
            }

            my $next;
            for ( 0 .. ( $last - $sum ) ) {
                $next = $ _ if $ite->[ $ _ ];
            }
            $ite->[$next] = 0;
            $ite->[ $next + 1 ] = $next + 2;

            $ite->[ $last + 1 - $ _ ] = 0 for ( 1 .. $sum );
            $ite->[ $next + 1 + $ _ ] = $next + 2 + $ _ for ( 1 .. $sum );

            $last = $next + 1 + $sum;
        }
        @div = _convert_div($ite);
        $is_finished = $div[-1] == $c;
    }
    return @divisors;
}

sub _all_divisors_iterator3 {

    my $c          = shift;
    my $test       = shift;
    my @arguments  = @_;
    my $s          = $n + $c - 1;
    my $ite        = [];
    push( @$ite, $ _ ) for ( 1 .. $c );

    push( @$ite, (0) x ( $s - $c ) );

```

```

my $last          = $c - 1;
my $previous_failure = 0;
my $div_last     = 0;
my $is_finished;
my @div = _convert_div($site);
my $burns;

my @divisors;
my $reduced_divisors;

my $t0 = [gettimeofday];

while (1) {
  push @divisors, $_ if ( $_ = &$test( \@div, @arguments ) );
  last if ( $is_finished );

  if ( $last < ( $s - 1 ) ) {
    $site->[$last] = 0;
    $last++;
    $site->[$last] = $last + 1;
  }
  else {
    my $sum = 0;
    for ( 1 .. $c ) {
      last if !$site->[ $s - $_ ];
      $sum++;
    }

    my $next;
    for ( 0 .. ( $last - $sum ) ) {
      $next = $_ if $site->[$_];
    }
    $site->[$next] = 0;
    $site->[ $next + 1 ] = $next + 2;

    $site->[ $last + 1 - $_ ] = 0 for ( 1 .. $sum );
    $site->[ $next + 1 + $_ ] = $next + 2 + $_ for ( 1 .. $sum );

    $last = $next + 1 + $sum;
  }
  @div = _convert_div($site);
  $is_finished = $div[-1] == $c;
}
return @divisors;
}

sub _context() {
my $self = shift;
$n       = $self->{n};
@mat     = @{ $self->{matrix} };
@lap     = @{ $self->{lap} };
@support = @{ $self->{support} };
@support_lap = @{ $self->{support_lap} };
@vertices = @{ $self->{vertices} };
@vertices_sorted = @{ $self->{vertices_sorted} };
@empty_divisor = @{ $self->{empty_divisor} };
@degrees  = @{ $self->{degrees} };
$c        = $self->{c};
@Li      = @{ $self->{Li} } if ( defined $self->{Li} );
$det     = $self->{det};
}

sub _clean_context() {
$n       = undef;
@mat     = undef;
@lap     = undef;
@support = undef;
@support_lap = undef;
@vertices = undef;
@vertices_sorted = undef;
@empty_divisor = undef;
@degrees  = undef;
$c        = undef;
}

```

```

    @Li          = undef;
    $det         = undef;
}

sub use_database {

    my $db_file = $_[1];
    $table_name = $_[2];

    $dbh = DBI->connect("dbi:SQLite:dbname=$db_file")
        or die $DBI::errstr;

    $dbh->do(
        "CREATE_TABLE_IF_NOT_EXISTS_$table_name\_rank1_(graph\_TEXT,\_divisors\_TEXT,\_rank1\_INT,\_rank2\_INT,\
        \_total\_INT,\_reduced\_INT,\_g\_INT,\_v\_INT,\_signature\_TEXT,\_det\_INT,\_factorization\_TEXT,\_time\_INT
        )"
    );
    $dbh->do(
        "CREATE_INDEX_IF_NOT_EXISTS_$table_name\_total\_ON\_table_name\_rank1_(total)"
    );
}

sub use_table {
    $table_name = $_[1];

    $dbh->do(
        "CREATE_TABLE_IF_NOT_EXISTS_$table_name\_rank1_(graph\_TEXT,\_divisors\_TEXT,\_rank1\_INT,\_rank2\_INT,\
        \_total\_INT,\_reduced\_INT,\_g\_INT,\_v\_INT,\_signature\_TEXT,\_det\_INT,\_factorization\_TEXT,\_time\_INT
        )"
    );
    $dbh->do(
        "CREATE_INDEX_IF_NOT_EXISTS_$table_name\_total\_ON\_table_name\_rank1_(total)"
    );
}

sub commit_database {

    my $suffix = $_[1];
    my $values = $_[2];

    $dbh->do("INSERT_INTO_$table_name\__$suffix_VALUES_$values");
}

sub BUILD {
    ChipFiring->use_database( 'graphs.sql', 'graphs' ) unless ($dbh);
}

1;

```


Bibliografía

- [1] Y. An, M. Baker, G. Kuperberg y F. Shokrieh. Canonical representatives for divisor classes on tropical curves and the matrix-tree theorem, 2014. DOI: 10.1017/fms.2014.25. arXiv: 1304.4259v2.
- [2] R. Bacher, P. d. La Harpe y T. Nagnibeda. The lattice of integral flows and the lattice of integral cuts on a finite graph. *Bulletin de la société mathématique de france*, 125(2):167-198, 1997. URL: http://archive.numdam.org/ARCHIVE/BSMF/BSMF_1997__125_2/BSMF_1997__125_2_167_0/BSMF_1997__125_2_167_0.pdf.
- [3] S. Backman. Riemann-roch theory for graph orientations. Ver. 3. 2015. arXiv: 1401.3309v3.
- [4] M. Baker. Specialization of linear systems from curves to graphs. 2007. arXiv: math/0701075.
- [5] M. Baker. Riemann-roch for graphs and applications. Blog. 2013. URL: <https://mattbakerblog.wordpress.com/2013/10/18/riemann-roch-for-graphs-and-applications/> (visitado 05-18-2016).
- [6] M. Baker. Reduced divisors and riemann-roch for graphs. Blog. 2014. URL: <https://mattbakerblog.wordpress.com/2014/01/12/reduced-divisors-and-riemann-roch-for-graphs/#more-269> (visitado 06-10-2016).
- [7] M. Baker y X. Faber. Metrized graphs, electrical networks, and fourier analysis. Ver. 2. 2005. arXiv: math/0407428v2.
- [8] M. Baker y X. Faber. Metric properties of the tropical abel-jacobi map. Ver. 4. 2010. arXiv: 0905.1679v2.
- [9] M. Baker y S. Norine. Harmonic morphisms and hyperelliptic graphs. Ver. 2. 2007. arXiv: 0707.1309v2.
- [10] M. Baker y S. Norine. Riemann-roch and abel-jacobi theory on a finite graph. Ver. 3. 2007. arXiv: math/0608360v3.

- [11] M. Baker y F. Shokrieh. Chip-firing games, potential theory on graphs, and spanning trees. Ver. 2. 2012. arXiv: 1107.1313v2.
- [12] B. Benson, D. Chakrabarty y P. Tetali. g -parking functions, acyclic orientations and spanning trees, 2010. arXiv: 0801.1114v3.
- [13] N. L. Biggs. Chip-firing and the critical group of a graph. *Journal of algebraic combinatorics*, 9(1):25-45, 1999.
- [14] A. Björner, L. Lovász y P. W. Shor. Chip-firing games on graphs. 1991. URL: <http://www.cs.elte.hu/~lovasz/morepapers/chips.pdf>.
- [15] B. Bollobas. *Modern graph theory*. De *Graduate Texts in Mathematics*. Springer New York, 1998. ISBN: 9780387984889.
- [16] F. Cools y J. Draisma. On metric graphs with prescribed gonality. 2016. arXiv: 1602.05542.
- [17] F. Cools, J. Draisma, S. Payne y E. Robeva. A tropical proof of the Brill-Noether theorem, 2012. arXiv: 1001.2774.
- [18] D. Dhar. Self-organized critical state of sandpile automaton models. *Phys. rev. lett.*, 64:1613-1616, 14, abr. de 1990. DOI: 10.1103/PhysRevLett.64.1613. URL: <http://link.aps.org/doi/10.1103/PhysRevLett.64.1613>.
- [19] W. Fulton. *Algebraic curves, an introduction to algebraic geometry*, 2008. URL: <http://www.math.lsa.umich.edu/~wfulton/CurveBook.pdf>.
- [20] P. Griffiths y J. Harris. On the variety of special linear systems on a general algebraic curve. *Duke math. j*, 47(1):233-272, 1980.
- [21] J. Harant. A lower bound on the independence number of a graph. *Discrete mathematics*, 188(1):239-243, 1998.
- [22] R. Hartshorne. *Algebraic geometry*. De *Graduate Texts in Mathematics*. Springer, 1977. ISBN: 9780387902449.
- [23] S. Hellbusch. Riemann-roch theorie auf graphen und anwendungen. 2013.
- [24] J. Hladký, D. Král' y S. Norine. Rank of divisors on tropical curves. Ver. 4. 2016. arXiv: 0709.4485v4.
- [25] M. Hoekstra. Chip-firing en de kritieke groep op een graaf. 2011. URL: http://scripties.fwn.eldoc.ub.rug.nl/FILES/scripties/Wiskunde/Bachelor/2011/Hoekstra.M./Bacheloronderzoek_eindversie.pdf.

- [26] B. Hujter y L. Tóthmérész. Chip-firing based methods in the riemann–roch theory of directed graphs. 2015. arXiv: 1511.03568v2.
- [27] D. Lorenzini. Smith normal form and laplacians. *Journal of combinatorial theory, series b*, 98(6):1271-1300, 2008.
- [28] D. J. Lorenzini. A finite group attached to the laplacian of a graph. *Discrete mathematics*, 91(3):277-282, 1991.
- [29] D. J. Lorenzini. Arithmetical properties of laplacians of graphs. 1999. arXiv: math/9903206.
- [30] Y. Luo. Rank-determining sets of metric graphs. Ver. 2. 2009. arXiv: 0906.2807v2.
- [31] N. Niezink. Abelian sandpiles. 2008. URL: http://scripties.fwn.eldoc.ub.rug.nl/FILES/scripties/Wiskunde/Bachelor/2008/Niezink.N./Nynke_Niezink_WB_2008.pdf.
- [32] S. Payne, D. Jensen y col. Lecture notes of seminar course on tropical brill–noether theory. 2016. URL: <http://users.math.yale.edu/~sp547/Math665.html> (visitado 03-07-2016).
- [33] D. Perkinson, J. Perlman y J. Wilmes. Primer for the algebraic geometry of sandpiles. 2011. arXiv: 1112.6163v2.
- [34] F. Shokrieh. The monodromy pairing and discrete logarithm on the jacobian of finite graphs. 2011. arXiv: 0907.4764v2.
- [35] J. Spencer. Balancing vectors in the max norm. *Combinatorica*, 6(1):55-65, 1986.