



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Métodos algebraicos para la determinación de grafos uniformemente más confiables

Informe de Proyecto de Grado presentado por

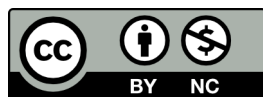
Juan José Mangado,
Carolina Vega y
Ezequiel Velazquez

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de
la República

Supervisor

Pablo Romero

Montevideo, 29 de diciembre de 2025



**Métodos algebraicos para la determinación de grafos
uniformemente más confiables** por Juan José Mangado,
Carolina Vega y
Ezequiel Velazquez tiene licencia [CC Atribución - No Comercial
4.0](#).

Agradecimientos

Este proyecto no habría sido posible sin el apoyo de quienes estuvieron a nuestro lado en cada paso. Queremos agradecer especialmente a nuestro tutor, Pablo Romero, por su dedicación y por estar siempre dispuesto a guiarnos con paciencia y claridad. Su acompañamiento marcó la diferencia en nuestro proceso. También damos las gracias a nuestras familias, quienes con su confianza y constante respaldo nos ayudaron a superar los retos de este camino.

A todos quienes nos acompañaron y ayudaron en este camino, muchas gracias.

Resumen

Este proyecto se enmarca dentro del estudio de confiabilidad de redes. Sean n y m enteros tales que $n \geq 3$ y $n - 1 \leq m \leq \binom{n}{2}$, y sea $\mathcal{C}_{n,m}$ la clase de grafos conexos y simples con n vértices y m aristas. El co-rango de cada uno de los grafos dentro de la clase $\mathcal{C}_{n,m}$ es $m - n + 1$. Sea G un grafo en $\mathcal{C}_{n,m}$ y sea $p \in [0, 1]$. La confiabilidad $R_G(p)$ es la probabilidad de que el subgrafo que se obtiene de G tras retener a cada una de sus aristas independientemente con probabilidad p resulte conexo. Decimos que G es uniformemente más confiable si para cada grafo H en $\mathcal{C}_{n,m}$ y cada $p \in [0, 1]$ se cumple que $R_G(p) \geq R_H(p)$.

Boesch conjeturó en 1986 que cada una de las clases no vacías $\mathcal{C}_{n,m}$ posee algún grafo uniformemente más confiable. Posteriormente, en un trabajo colectivo, Boesch probó que cada una de las clases no vacías $\mathcal{C}_{n,m}$ de co-rango positivo no mayor que 3 posee un único grafo que es uniformemente más confiable. Además, en dicho trabajo se conjetura que, para cada entero n tal que $n \geq 6$, existe un grafo uniformemente más confiable de co-rango 4 con n vértices que se obtiene mediante ciertas subdivisiones elementales del grafo bipartito completo $K_{3,3}$. En 1994, Wang anunció que dicha conjetura es correcta y publicó su demostración. No obstante, un reciente artículo de Landgren y Steif en 2024 sugiere que hay errores en la demostración de Wang y que el resultado no es correcto. Dicho trabajo fue publicado en ArXiv y se encuentra bajo referato científico. De manera casi simultánea, Kahl y Luttrell definen el concepto de grafo Tutte-máximo que se basa en el polinomio de Tutte de un grafo, y prueban que cada grafo Tutte-máximo es uniformemente más confiable.

En este proyecto se desea validar o refutar computacionalmente la afirmación realizada por Wang en 1994 utilizando métodos inspirados en grafos Tutte-máximos. Este proyecto de grado tiene tres objetivos. El primer objetivo consiste en familiarizarnos con los conceptos de grafo Tutte-máximo y de grafo uniformemente más confiable. El segundo objetivo consiste en desarrollar algoritmos eficientes para determinar el polinomio de Tutte para grafos de co-rango reducido. El tercer y último objetivo consiste en refutar o validar computacionalmente el teorema anunciado en 1994 por Wang. Nuestros resultados evidencian que la afirmación de Wang no es correcta y respaldan la corrección propuesta por Landgren y Steif.

Palabras clave: Grafos Tutte-máximos, Grafos uniformemente más confiables.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Estructura del documento	3
2. Grafos uniformemente más confiables	5
2.1. Conceptos básicos	5
2.2. Clases de grafos de estudio	7
2.3. Usos prácticos de la confiabilidad	10
3. Grafos Tutte-máximos	13
3.1. Polinomio de Tutte	13
3.2. Grafos Tutte-máximos	16
3.3. Resultados Preliminares	18
4. Aportes	21
4.1. Cálculo del polinomio de Tutte	21
4.1.1. Algoritmo Básico	22
4.1.2. Algoritmo Propuesto	23
4.1.3. Terminación y Correctitud	25
4.1.4. Eficiencia computacional	26
4.2. Búsqueda de grafos Tutte-máximos	28
4.2.1. Grafos 2-conexos y destilaciones	29
4.2.2. Algoritmo Propuesto	32
4.2.3. Terminación y Correctitud	33
4.3. Conclusiones del Capítulo	34
5. Refutación a la afirmación de Wang	35
5.1. Determinación de un grafo Tutte-máximo	35
5.2. Comparación entre W^s y L^s	36
6. Conclusiones y trabajo futuro	39
A. Anexo 1	45

Capítulo 1

Introducción

En este breve capítulo introductorio se presenta una motivación histórica del problema de estudio seguido de la organización del documento.

1.1. Motivación

El estudio de confiabilidad de redes tiene registro desde 1956 a partir de un artículo publicado por Moore y Shannon [12]. La motivación principal de sus autores era el desarrollo de computadores mediante el uso de electrónica basada en relés. Cada relé tenía una probabilidad de falla, y el objetivo consistía en mantener comunicados 2 puntos concretos del circuito que alimentaba el computador.

En 1986, Boesch formaliza una noción de maximización de la confiabilidad utilizando el lenguaje de la teoría de grafos. Boesch, en la reconocida revista *Journal of Graph Theory* [3], no solamente define el concepto de grafo uniformemente más confiable, sino que también propone diversas conjeturas. Hasta la fecha se sabe que la mayoría de las conjeturas propuestas por Boesch son falsas, y que pese a ello han dado motivo no solamente a diversas investigaciones, sino también han permitido generar nuevas preguntas y florecer el área de investigación de la *síntesis en confiabilidad*, también conocido como *confiabilidad uniforme*.

Una de las conjeturas de Boesch afirma que en cada una de las clases no vacías de grafos conexos y simples con n vértices y m aristas existe un grafo uniformemente más confiable. En un posterior trabajo colectivo desarrollado en 1991, Boesch et al. [2] demostraron que en cada una de las clases de grafos conexos y simples cuyo co-rango es no mayor que 3 existe precisamente un grafo que es uniformemente más confiable, y conjeturaron que todos los grafos uniformemente más confiables de co-rango 4 con al menos 6 vértices provienen de subdivisiones elementales del grafo bipartito completo $K_{3,3}$. En 1994, Wang

anunció que dicha conjetura es correcta, y publicó su demostración [20]. En lo que sigue denotaremos con el símbolo W_n al único grafo con n vértices y co-rango 4 obtenido por Wang.

Esto brinda evidencia de que la conjetura de Boesch relativa a la existencia de grafos uniformemente más confiables es parcialmente cierta para las clases de grafos de co-rango 4 o menor. No obstante, hasta la fecha se conocen infinitos contraejemplos a dicha conjetura [4, 10, 13, 17]. En particular, existe una cantidad finita de grafos uniformemente más confiables de co-rango 5, contrastando fuertemente lo que ocurre en clases cuyo co-rango es menor que 5.

Nuestro tutor ha publicado un *Survey* en el área de grafos uniformemente más confiables, donde reúne resultados obtenidos hasta el año 2021 relativos a la existencia y construcción de grafos uniformemente más confiables ante modelos de fallas de caídas de vértices o de aristas [15]. Inmediatamente luego de la publicación de dicho Survey surge en la literatura científica un nuevo enfoque algebraico para la determinación de grafos uniformemente más confiables que se basa en el polinomio de Tutte. Dicho enfoque se sustenta, primeramente, en un trabajo publicado por Kahl en la reconocida revista titulada *Journal of Combinatorial Theory, Series B*, donde define una relación de orden parcial dentro de ciertas clases de equivalencia de grafos conexos y simples con una cantidad preestablecida de vértices y aristas [8]. Posteriormente, Kahl y Luttrell [9] definen el concepto de grafo Tutte-máximo basado en el primer trabajo de Kahl. A partir del teorema receta de los polinomios de Tutte se desprende directamente que cada grafo Tutte-máximo es uniformemente más confiable. Kahl y Luttrell probaron que cada uno de los grafos uniformemente más confiables obtenidos por Boesch et al. en [2] son Tutte-máximos. Kahl y Luttrell luego conjeturaron que en cada clase de grafos no vacía con al menos 6 vértices y co-rango 4 existe un único grafo que es Tutte-máximo.

Curiosamente, el estudiante de doctorado de la Universidad de Chalmers, Lorents Landgren, en colaboración con su orientador Jeffrey Steif, dejaron disponible en ArXiv [11] una prepublicación que arremete contra los fundamentos de la confiabilidad uniforme. Dicho artículo se encuentra bajo referato ¹. Concretamente, Lorents Landgren y Jeffrey Steif desarrollan un nuevo método para obtener grafos uniformemente más confiables. Luego, identifican algunas falencias inherentes a la demostración del teorema de Wang [20], generando incertidumbre sobre la validez de resultados publicados más de 3 décadas atrás y con abundante cantidad de citaciones en el área de confiabilidad uniforme. Por último presentan, para cada entero n tal que $n \geq 6$, un grafo con n vértices y co-rango 4 que aquí denotaremos L_n (por la inicial del primer autor, Landgren), y prueban que L_n es el único grafo uniformemente más confiable dentro de todos los grafos conexos y simples con la misma cantidad de vértices y aristas que L_n . Cabe aclarar que los grafos L_n

¹Un borrador se encuentra disponible en ArXiv: <https://arxiv.org/pdf/2407.20217>

y W_n coinciden excepto para aquellos enteros positivos n que son congruentes con 2 en módulo 9.

Este proyecto se propone estudiar, en base a métodos algebraicos y computacionales, si es válida la corrección propuesta por Landgren y Steif a la afirmación de Wang. La principal fuente de inspiración será el nuevo método desarrollado por Kahl [8], y por Kahl y Luttrell [9], para obtener grafos Tutte-máximos.

Este proyecto de grado tiene tres objetivos específicos. El primer objetivo consiste en familiarizarnos con los conceptos de grafo Tutte-máximo y de grafo uniformemente más confiable. El segundo objetivo del proyecto consiste en desarrollar algoritmos eficientes para determinar el polinomio de Tutte de un grafo de co-rango reducido. El tercer objetivo consiste en comparar los grafos W_n y L_n propuestos en respectivas publicaciones, por un lado por Wang [20] y por otro lado por Landgren y Steif. Concretamente, nos interesa saber si el grafo L_n es más confiable que el grafo W_n para aquellos enteros positivos n que son congruentes con 2 módulo 9.

1.2. Estructura del documento

Este documento se organiza de la siguiente manera. En el Capítulo 2 se presentan conceptos relativos al área de confiabilidad uniforme y se definen, para cada entero n tal que $n \geq 6$, los grafos W_n y L_n . En el Capítulo 3 se presenta el polinomio de Tutte y su propiedad de universalidad, también conocido en la literatura como el teorema receta. Luego se presenta el concepto de grafo Tutte-máximo y se incluye una demostración de que todo grafo Tutte-máximo es uniformemente más confiable. Cabe señalar que tanto el Capítulo 2 como el Capítulo 3 se obtienen a partir de una recopilación de trabajos previos.

Los aportes de este proyecto de grado se reúnen en los Capítulos 4, 5, y 6. En el Capítulo 4 se describen dos algoritmos novedosos que fueron desarrollados en el transcurso de este proyecto para la búsqueda de grafos Tutte-máximos. El primer algoritmo emplea propiedades del polinomio de Tutte para hallar dicho polinomio para cualquier pseudografo en su entrada. Algunos experimentos computacionales sugieren que su rendimiento es competitivo con otras técnicas de cómputo del polinomio de Tutte para clases de grafos de co-rango reducido. Cabe señalar que el cálculo del polinomio de Tutte pertenece a la clase de problemas #P-Complejos. Pese a ello, veremos que el algoritmo propuesto es eficiente para el cálculo del polinomio de Tutte en grafos de co-rango fijo reducido, incluso ante grafos con una cantidad masiva de vértices, lo que asegura escalabilidad. El segundo algoritmo desarrollado recibe dos enteros positivos n y m tales que $n + 1 \leq m \leq \frac{3n}{2}$ y entrega un grafo conexo y simple que es Tutte-máximo en caso de existir, o bien un indicador de que no existe dicho grafo dentro de la clase de grafos conexos y simples con n vértices y m aristas. En el Capítulo 5 se presentan ejecuciones de los programas desarrollados que se hallan en armonía con la corrección propuesta por los autores Landgren y Steif. En el Capítulo 6

señalamos las principales conclusiones de este trabajo y brindamos posibles líneas de trabajo futuro.

Capítulo 2

Grafos uniformemente más confiables

En este capítulo se incluyen nociones sobre confiabilidad uniforme que serán centrales para el desarrollo del presente proyecto. En la Sección 2.1 se repasan conceptos básicos de teoría de grafos que se pueden encontrar en libros clásicos [2, 3]. En la Sección 2.2 se presenta el concepto de confiabilidad de un grafo y de grafo uniformemente más confiable, comentando brevemente aspectos de complejidad computacional. Asimismo se definen, para cada entero n tal que $n \geq 6$, los dos grafos W_n y L_n que fueron propuestos respectivamente por Wang, y por Landgren y Steif.

2.1. Conceptos básicos

Esta sección ofrece una breve revisión de los conceptos fundamentales de teoría de grafos con el objetivo de facilitar la comprensión de los capítulos posteriores. Si el lector ya está familiarizado con estos contenidos, puede avanzar directamente a la Sección 2.2 sin pérdida de continuidad en la exposición principal.

A lo largo de todo el documento trabajaremos únicamente con grafos finitos y no dirigidos. Recordemos que un *multiconjunto* es un conjunto en el que los elementos pueden repetirse. Un *grafo simple* G consiste en un conjunto finito de vértices, denotado por $V(G)$, y un conjunto de aristas, denotado por $E(G)$, que son pares no ordenados de elementos de $V(G)$. Utilizaremos la notación uv en lugar de $\{u, v\}$ para denotar a la arista cuyos extremos son los vértices u y v . Diremos que dos vértices de G son *adyacentes* cuando son extremos de alguna arista de G , y que dicha arista es *incidente* a cada uno de sus extremos. El *grado de un vértice v de G* es la cantidad de aristas incidentes a v . En un *multigrafo* admitimos que las aristas sean múltiples, por lo que $E(G)$ es un multiconjunto. En un *pseudografo* admitimos además la presencia de *lazos*, que son aristas de la forma vv para algún vértice v de G .

Sea G un grafo. Un *subgrafo de G* es un grafo que cumple que su conjunto de vértices está contenido en $V(G)$ y su conjunto de aristas está contenido en $E(G)$. La *sustracción de la arista e en G* se denota $G - e$ y consiste en el grafo cuyo conjunto de vértices es $V(G)$ y cuyo conjunto de aristas es $E(G) - e$. La *contracción de la arista e en G* , que se denota $G * e$, es el grafo que se obtiene de $G - e$ tras identificar a los vértices que son extremos de e . La *remoción de un vértice v en G* es el subgrafo de G cuyo conjunto de vértices es $V(G) - \{v\}$ y su conjunto de aristas es precisamente el conjunto de aristas del grafo G menos las aristas que son incidentes al vértice v . Un *subgrafo recubridor* de G es un subgrafo H de G tal que $V(H) = V(G)$. Dos grafos G y H son *isomorfos* si existe una biyección $\psi : V(G) \rightarrow V(H)$ tal que $uv \in E(G)$ si y solo si $\psi(u)\psi(v) \in E(H)$. Un *camino en G* es una secuencia de vértices y aristas de G de la forma $v_0, e_1, v_1, e_2, v_2, \dots, e_r, v_r$, donde $v_0, v_1, \dots, v_r \in V(G)$ y $e_1, e_2, \dots, e_r \in E(G)$. El *largo* de este camino es r , sus *extremos* son los vértices v_0 y v_r , y sus *vértices internos* son los vértices del camino que no son extremos, si los hay. Denotamos P_n al grafo que consiste en un único camino de n vértices. Para cada entero positivo n denotamos C_n al grafo (o multigrafo) cuyo conjunto de vértices es v_1, \dots, v_n tal que $v_i v_{i+1} \in E$ cuando $i \in \{1, \dots, n-1\}$ y $v_n v_1 \in E$ (notar que C_1 tiene un único vértice y un lazo, mientras que C_2 es un multigrafo). Decimos que G es *conexo* si para cada par de vértices u y v del grafo G existe algún camino cuyos extremos son precisamente los vértices u y v . Una *componente conexa* del grafo G es un subgrafo que es conexo y maximal por inclusión, es decir, es un subgrafo conexo de G que no está contenido propiamente en ningún otro subgrafo conexo de G ; equivalentemente, no es posible agregar a dicho subgrafo ningún vértice ni arista de G sin perder la propiedad de ser conexo. Decimos que un vértice v de G es un *punto de corte* si $G - v$ tiene más componentes conexas que G . Decimos que e es una *arista puente* si el grafo $G - e$ tiene más componentes conexas que G . Decimos que el grafo G es 2-conexo si G es un grafo conexo con al menos 3 vértices que no tiene puntos de corte. Llamamos *árbol* a aquel grafo que es conexo y no posee ciclos. Decimos que un grafo G es *cúbico* si todos sus vértices tienen grado 3. Definimos *grafo completo* de n vértices, y lo denotamos K_n , al grafo con n vértices donde cada par de vértices distintos está conectado por exactamente una arista. Adicionalmente, decimos que un grafo es *no separable* si es conexo y no tiene puntos de corte; de lo contrario, se considera *separable*. Los subgrafos no separables maximales por inclusión se denominan *bloques* del grafo. Cuando un grafo conexo G se compone de N bloques G_1, \dots, G_N diremos que G pertenece a la clase de grafos $G_1 \cdot G_2 \cdots G_N$; dependiendo del contexto simplemente diremos que $G = G_1 \cdot G_2 \cdots G_N$. Para cada par de pseudografos G y H y vértices $v \in V(G)$ y $w \in V(H)$, denotamos $G_v \cdot H_w$ al pseudografo que se obtiene tras identificar a los vértices v y w en $G \cup H$. A lo largo del documento, utilizaremos la notación $\mathcal{M}_{n,m}$ y $\mathcal{B}_{n,m}$ para referirnos, respectivamente, a las clases de todos los multigrafos y multigrafos 2-conexos con n vértices y m aristas.

Además, para cada par de enteros positivos n y m tales que $n-1 \leq m \leq \binom{n}{2}$, denotamos mediante $\mathcal{C}_{n,m}$ (respectivamente $\mathcal{C}_{n,m}^2$) a la clase de todos los grafos simples y conexos (respectivamente, grafos simples y 2-conexos) con n vértices y

m aristas. Para cada pseudografo G con $\kappa(G)$ componentes conexas, definimos el *rango de G* , y lo denotamos $r(G)$, como $|V(G)| - \kappa(G)$. De manera similar, definimos el *co-rango de G* , y lo denotamos $c(G)$, como $|E(G)| - |V(G)| + \kappa(G)$. Observar que cada uno de los grafos pertenecientes a $\mathcal{C}_{n,m}$ tiene rango $n - 1$ y co-rango $m - n + 1$. Por conveniencia diremos que el *rango de la clase de grafos $\mathcal{C}_{n,m}$* es igual a $n - 1$, mientras que el *co-rango de la clase de grafos $\mathcal{C}_{n,m}$* es igual a $m - n + 1$. Un *anillo* es un conjunto no vacío R provisto de dos operaciones binarias, suma $+$ y producto \cdot , tales que:

1. $(R, +)$ es un grupo abeliano, es decir, un grupo en el cual la suma es conmutativa.
2. El producto es asociativo.
3. El producto es distributivo respecto de la suma.

Un invariante bajo isomorfismos, o simplemente, un *invariante* es una función f cuyo dominio es una clase de grafos cerrada bajo las operaciones de contracción y sustracción, y cuyo codominio es algún anillo, tal que $f(G) = f(H)$ siempre que G y H sean grafos isomorfos.

2.2. Clases de grafos de estudio

En esta sección se presentan conceptos fundamentales del área de confiabilidad uniforme que serán utilizados en este proyecto. Asimismo, presentaremos los grafos definidos por Wang que denotamos W_n , como también los grafos definidos por Landgren y Steif, que denotamos L_n , los cuales conforman nuestro caso de estudio. Para dar una definición precisa será necesario introducir el concepto de cadenas y de destilaciones presentado por Romero y Safe [17].

Definición 2.2.1. Sean n y m dos enteros positivos tales que $n - 1 \leq m \leq \binom{n}{2}$. Sea G un grafo en $\mathcal{C}_{n,m}$ y sea $p \in [0, 1]$. La *confiabilidad de G evaluada en p* , que denotamos $R_G(p)$, es la probabilidad de que el subgrafo recubridor aleatorio resultante de retener a cada una de las aristas de G independientemente con probabilidad p sea conexo.

La búsqueda de grafos con máxima confiabilidad es de interés desde mediados del Siglo XX, pero fue formalizada utilizando el lenguaje de teoría de grafos recién en 1986 tras un artículo seminal publicado por Frank Boesch [3].

Definición 2.2.2. Un grafo G en $\mathcal{C}_{n,m}$ es *uniformemente más confiable* (UMRG por sus siglas en inglés) si para cada grafo H en $\mathcal{C}_{n,m}$ y todo p en $[0, 1]$ se cumple que $R_G(p) \geq R_H(p)$.

Introduciremos ahora una serie de conceptos necesarios para definir de forma precisa los grafos comprendidos en nuestro caso de estudio [17]. Sea G un multigrafo en $\mathcal{B}_{n,m}$ con más aristas que vértices. Una *cadena de G* , que

denotamos γ , es el conjunto de aristas de un camino P en G cuyos extremos tienen grado 3 o más mientras que cada uno de sus vértices internos (si los hay) tienen grado 2. Los *extremos de γ* son precisamente los extremos del camino P y los *vértices internos de γ* son precisamente los vértices internos del camino P . Sea γ una cadena de G . El *largo de γ* es igual a su cardinal, que denotamos $|\gamma|$. Cuando nos referimos a *eliminar la cadena γ de G* nos referimos a eliminar cada una de las aristas de γ y también sus vértices internos, pero no sus extremos. La *remoción de la cadena γ en G* se denota $G \ominus \gamma$. La *contracción de la cadena γ en G* , que se denota $G * \gamma$, es el grafo que se obtiene de $G \ominus \gamma$ tras identificar los vértices extremos de γ . Definimos *colapsar la cadena γ de G* al grafo que se obtiene de $G \ominus \gamma$ tras agregar una sola arista cuyos extremos coinciden con los de γ . La *destilación del grafo G* , que denotaremos $D(G)$, es el grafo que se obtiene de G tras colapsar a cada una de sus cadenas. Notar que G se obtiene a partir de $D(G)$ tras una secuencia finita de subdivisiones. Denotaremos por \mathcal{M}_c al conjunto de destilaciones de los grafos de $\mathcal{C}_{n,m}$, siendo $c = m - n + 1$. Llamaremos *subdividir k veces una arista xy* a sustituir dicha arista por $k + 1$ aristas de la forma $xz_1, z_1z_2, \dots, z_{k-1}z_k, z_ky$, donde z_1, z_2, \dots, z_k son k nuevos vértices. Por último, denotamos mediante θ_{l_1, l_2, l_3} al grafo compuesto por dos vértices y tres cadenas que los unen, de largos l_1, l_2 y l_3 respectivamente. De forma análoga, definimos el *grafo θ generalizado $\theta_{l_1, l_2, \dots, l_n}$* como el grafo formado por dos vértices unidos por n cadenas de largos l_1, l_2, \dots, l_n .

Definición 2.2.3 (Romero y Safe [17]). Sea G un grafo cúbico y 2-conexo y sea X un subconjunto de $E(G)$. Para cada entero positivo s denotamos $G^s(X)$ al grafo que se obtiene de G tras subdividir s veces cada arista de X y $s - 1$ veces cada arista de $E(G) - X$.

En el artículo [2] se conjetura que cada UMRG de co-rango 4 con al menos 6 vértices se obtiene a partir de subdivisiones del grafo bipartito completo $K_{3,3}$. En 1994, Wang [20] definió la siguiente clase de grafos de co-rango 4.

Definición 2.2.4 (Wang [20]). Sea n un entero tal que $n \geq 6$, y sean r y s los únicos enteros tales que $r \in \{0, 1, \dots, 8\}$ y $n + 3 = 9s + r$. Consideremos las aristas e_1, e_2, \dots, e_9 del grafo $K_{3,3}$ enumeradas tal como se indica en la Figura 2.1. Definimos X_r como el conjunto vacío si $r = 0$, mientras que X_r es igual a $\{e_1, e_2, \dots, e_r\}$ en caso contrario. Definimos el grafo W_n como $K_{3,3}^s(X_r)$.

En 2024, Landgren y Steif definieron la siguiente clase de grafos.

Definición 2.2.5 (Landgren y Steif). Sea n un entero tal que $n \geq 6$, y sean r y s los únicos enteros tales que $r \in \{0, 1, \dots, 8\}$ y $n + 3 = 9s + r$. Si $r = 0$ definimos Y_r como el conjunto vacío. Si $r = 5$ definimos Y_r como el conjunto de aristas $\{e_1, e_2, e_3, e_4, e_7\}$. En caso contrario, definimos Y_r como $\{e_1, e_2, \dots, e_r\}$. Definimos el grafo L_n como $K_{3,3}^s(Y_r)$.

Para cada entero positivo s denotamos por L^s y W^s a los grafos L_{9s+2} y W_{9s+2} respectivamente. A modo de ejemplo, cuando $n = 11$ se tiene que

$n + 3 = 9s + r$, donde $s = 1$ y $r = 5$. Esto significa que cada una de las cadenas de W_{11} y L_{11} tienen largo igual a 1 o a 2. Por un lado, para el grafo W_{11} las únicas cadenas que tienen largo igual a 2 se corresponden con las aristas de X_5 dentro de $K_{3,3}$ dadas por el conjunto $\{e_1, e_2, e_3, e_4, e_5\}$. Por otro lado, para el grafo L_{11} tenemos que las únicas cadenas que tienen largo igual a 2 se corresponden con las aristas de Y_5 dentro de $K_{3,3}$ dadas por el conjunto $\{e_1, e_2, e_3, e_4, e_7\}$. La Figura 2.2 ilustra a los grafos W_{11} y L_{11} .

Observemos que los grafos W_{11} y L_{11} no son isomorfos. De hecho, basta con mostrar que W_{11} tiene un ciclo hamiltoniano mientras que L_{11} no tiene ningún ciclo hamiltoniano. Recordemos que un ciclo hamiltoniano de un grafo es un ciclo que incluye cada vértice del grafo exactamente una vez. Todo ciclo hamiltoniano en W_{11} o en L_{11} debe incluir a cada una de las aristas pintadas en rojo en la Figura 2.2. Por un lado, W_{11} tiene un ciclo hamiltoniano que incluye a todas las aristas que tienen color rojo más una arista que tiene color negro. Por otro lado, L_{11} tiene un conjunto de aristas con color rojo que ya forman un ciclo que no incluye a todas las aristas, y por lo tanto L_{11} no tiene ningún ciclo hamiltoniano. Como consecuencia, L_{11} y W_{11} no son isomorfos, y los siguientes enunciados propuestos separadamente por Wang [20] y por Landgren y Steif no pueden ser simultáneamente correctos.

Afirmación 1 (Wang [20]). *Para cada entero n tal que $n \geq 6$ se tiene que W_n es el único UMRG en $\mathcal{C}_{n,n+3}$.*

Afirmación 2 (Landgren y Steif). *Para cada entero n tal que $n \geq 6$ se tiene que L_n es el único UMRG en $\mathcal{C}_{n,n+3}$.*

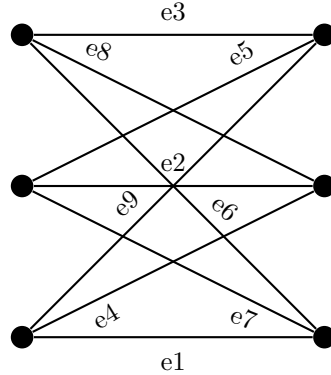


Figura 2.1: Grafo $K_{3,3}$ etiquetado.

En el Capítulo 5 probaremos que la Afirmación 1 no es correcta. Dicha prueba no solo contribuye a esclarecer la controversia actual entre ambas afirmaciones, sino que también establece un precedente para futuras investigaciones en UMRG.

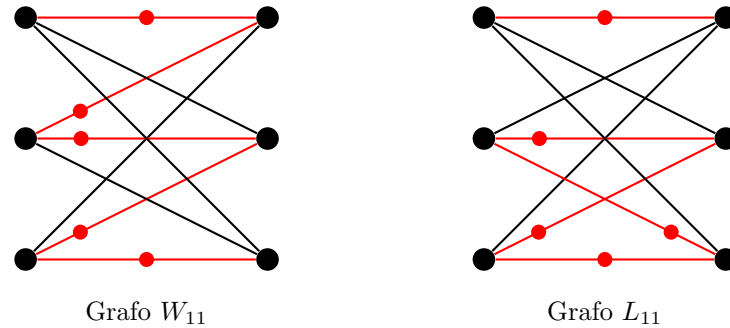


Figura 2.2: Grafos W_{11} y L_{11}

2.3. Usos prácticos de la confiabilidad

Para concluir este capítulo, es pertinente analizar los usos prácticos del concepto de confiabilidad aplicado a grafos. Esta sección tiene como objetivo destacar su valor en el diseño de estructuras capaces de resistir fallas aleatorias.

Uno de los principales campos de aplicación es el diseño de redes de comunicación. En particular, los grafos UMRG son grafos que maximizan la confiabilidad en toda la gama de probabilidades de falla del enlace. Tal propiedad los vuelve relevantes para la ingeniería de redes donde no se conoce a priori la tasa de falla de los enlaces. En particular, nuestro tutor Pablo Romero en un trabajo de 2019 [14], probó que los grafos de Wagner y Petersen son UMRG en sus respectivas clases, lo cual los convierte en topologías óptimas para redes pequeñas, independientemente de la confiabilidad individual de sus enlaces.

Otro uso destacado de la confiabilidad es su aplicación como criterio de comparación entre distintas topologías de red. Por ejemplo, Yang y Han [21] establecen una fórmula cerrada para el número de árboles recubridores de hipercubos generalizados. Dado que los UMRG deben maximizar esta cantidad, dicha métrica puede utilizarse para comparar de forma objetiva distintas arquitecturas de interconexión. Esto resulta útil al momento de elegir topologías con mejor comportamiento esperado ante fallas, sin necesidad de realizar simulaciones costosas para cada caso.

Finalmente, los UMRG también se han utilizado como referencia para estimar mejoras en la confiabilidad al agregar redundancia estructural a una topología dada. Este enfoque es especialmente útil en escenarios donde el presupuesto impone un límite estricto en la cantidad de enlaces que pueden añadirse. En estos casos, comparar la confiabilidad de la red real con la del UMRG de su clase permite cuantificar cuán cerca se encuentra del máximo alcanzable, y por ende, evaluar si la incorporación de una o más aristas adicionales resulta justificable.

En todos estos contextos, los resultados teóricos en torno a los grafos UMRG permiten fundamentar decisiones de diseño orientadas a la robustez estructural, especialmente en situaciones donde se desconoce o varía la probabilidad de falla de los enlaces. Si bien muchas de estas aplicaciones se desarrollan a nivel simulado o computacional, su utilidad práctica reside en ofrecer criterios objetivos para comparar y seleccionar topologías con comportamiento superior bajo incertidumbre.

Capítulo 3

Grafos Tutte-máximos

La determinación de grafos Tutte-máximos es central en el área de confiabilidad uniforme puesto que cada grafo Tutte-máximo es un UMRG. Este resultado será una consecuencia inmediata de la propiedad de universalidad del polinomio de Tutte.

Este capítulo se organiza de la siguiente manera. La Sección 3.1 presenta el polinomio de Tutte y sus propiedades, incluyendo su universalidad. La Sección 3.2 incluye el concepto de grafo Tutte-máximo y sus propiedades más notables. En particular, todo grafo Tutte-máximo es un UMRG, lo que brinda así una estrecha conexión con el área de confiabilidad uniforme. En la Sección 3.3 se presentan propiedades adicionales del polinomio de Tutte que serán de utilidad para las pruebas de correctitud de los algoritmos desarrollados a lo largo de este proyecto.

3.1. Polinomio de Tutte

El polinomio de Tutte es un polinomio en dos variables reales que codifica información relevante de un pseudografo.

Definición 3.1.1 (W. Tutte [18]). El polinomio de Tutte $T_G(x, y)$ de un pseudografo G se define recursivamente de la siguiente manera:

$$T_G(x, y) = \begin{cases} T_{G-e}(x, y) + T_{G*e}(x, y) & \text{si } e \text{ no es un lazo ni un puente,} \\ x \cdot T_{G*e}(x, y) & \text{si } e \text{ es un puente,} \\ y \cdot T_{G-e}(x, y) & \text{si } e \text{ es un lazo,} \\ 1 & \text{si } G \text{ no tiene aristas.} \end{cases}$$

Desde el punto de vista del cálculo del polinomio de Tutte adoptaremos el siguiente concepto de grafos equivalentes.

Definición 3.1.2 (Kahl y Luttrell [9]). Dos grafos G y H en $\mathcal{C}_{n,m}$ son *Tutte-equivalentes* si para cada par de números reales x e y se cumple que $T_G(x, y) = T_H(x, y)$.

Para afianzar conceptos presentamos el Ejemplo 3.1.1, que ilustra gráficamente el cálculo del polinomio de Tutte correspondiente al grafo $\theta_{1,1,1}$.

Ejemplo 3.1.1. Un ejemplo del cálculo del polinomio de Tutte usando la Definición 3.1.1 se muestra en la Figura 3.1. La arista que se elimina o contrae en cada paso se resalta con un mayor grosor. El polinomio de Tutte de $\theta_{1,1,1}$ es $T_{\theta_{1,1,1}}(x, y) = T_{P_2}(x, y) + T_{C_1}(x, y) + T_{C_1 \cdot C_1}(x, y) = x + y + y^2$.

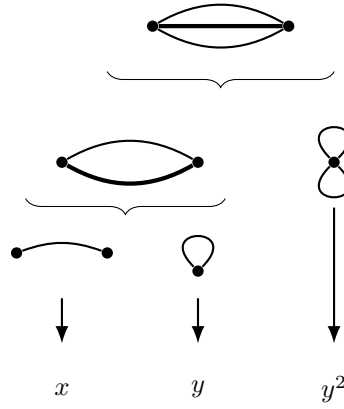


Figura 3.1: Empleo de la Definición 3.1.1 para hallar $T_{\theta_{1,1,1}}(x, y)$.

Observación 1. Como se observa en la Figura 3.1, su cálculo presenta un crecimiento exponencial de operaciones con respecto al número de aristas del grafo. De hecho, Leslie Valiant demostró en su influyente trabajo *The Complexity of Computing the Permanent* que este tipo de problema pertenece a la clase de problemas #P-completos [19].

Una observación pertinente al considerar la definición del polinomio de Tutte es que, si G es un pseudografo arbitrario y e es una arista de G que no es ni un puente ni un lazo, entonces el polinomio $T_G(x, y)$ satisface la denominada *propiedad de arista contracción-sustracción*, la cual afirma que

$$T_G(x, y) = T_{G-e}(x, y) + T_{G*e}(x, y).$$

Resulta sorprendente que una inmensa variedad de invariantes, que también satisfacen la propiedad de arista contracción-sustracción, se puedan obtener mediante evaluaciones del polinomio de Tutte. Este fenómeno se conoce como la *universalidad del polinomio de Tutte*. A continuación, se define una clase de invariantes que corresponden, precisamente, a aquellas funciones derivadas de evaluaciones del polinomio de Tutte sobre un pseudografo.

Definición 3.1.3. Sea \mathcal{G} una clase de pseudografos que es cerrada bajo las operaciones de sustracción y contracción de aristas que incluye a K_1 . Sea R un anillo conmutativo con unidad. Decimos que un invariante de grafos $f : \mathcal{G} \rightarrow R$ es de *Tutte–Grothendieck* si cumple simultáneamente con las siguientes condiciones:

- $f(K_1) = 1$.
- Existen $a, b \in R$ tales que para todo grafo G en \mathcal{G} y toda arista e de G que no es puente ni lazo se cumple que $f(G) = a \cdot f(G - e) + b \cdot f(G * e)$.
- Para todo par de grafos G y H en \mathcal{G} tales que $G \cup H \in \mathcal{G}$ (resp. $G_v \cdot H_w \in \mathcal{G}$), se cumple que $f(G \cup H) = f(G)f(H)$ (resp. $f(G_v \cdot H_w) = f(G)f(H)$).

Estamos en condiciones de enunciar la propiedad de *universalidad del polinomio de Tutte* que será fundamental para el desarrollo de nuestro proyecto.

Teorema 2 (Universalidad del polinomio de Tutte [6]). *Sea \mathcal{G} una clase de pseudografos cerrada bajo las operaciones de arista contracción y sustracción que incluye tanto a P_2 como a C_1 . Sea R un anillo conmutativo con unidad. Para cada invariante de Tutte–Grothendieck $f : \mathcal{G} \rightarrow R$ se cumple que $f(G) = a^{c(G)}b^{r(G)}T_G(x_0b^{-1}, y_0a^{-1})$, donde $x_0 = f(P_2)$, $y_0 = f(C_1)$ y los elementos a y b del anillo R están dados por la Definición 3.1.3 del invariante f .*

En el libro *Handbook of the Tutte Polynomial and Related Topics* [6], el Teorema 2 es referido como *teorema receta*. Este descriptivo alude a su capacidad para determinar cualquier invariante de Tutte–Grothendieck específico mediante una evaluación del polinomio de Tutte. En concordancia con dicha terminología, en este trabajo adoptaremos el mismo nombre y, en adelante, utilizaremos la expresión *teorema receta* para referirnos al Teorema 2.

El *teorema receta* establece que cualquier invariante de Tutte–Grothendieck puede obtenerse a partir de una evaluación apropiada del polinomio de Tutte. Esta afirmación resulta especialmente significativa para nuestro estudio, ya que con base en dicho teorema demostraremos que la confiabilidad de un grafo G es un invariante de Tutte–Grothendieck, por tanto es posible expresarla en función al polinomio de Tutte asociado al grafo.

Lema 3. *Para cada grafo G en $\mathcal{C}_{n,m}$ y cada p en $(0, 1)$ se tiene que*

$$R_G(p) = p^{n-1}(1-p)^{m-n+1}T_G\left(1, \frac{1}{1-p}\right) \quad (3.1)$$

Demostración. La estrategia de la demostración consiste en mostrar que el polinomio de confiabilidad es un invariante de Tutte–Grothendieck dentro de la clase de todos los pseudografos, para posteriormente aplicar el resultado obtenido en la subclase $\mathcal{C}_{n,m}$.

Sea p un número real en $[0, 1]$. Notemos primero que $R_{P_2}(p) = p$ y que $R_{C_1}(p) = 1$ para todo p en $[0, 1]$. Sea \mathcal{G} el conjunto que consiste en todos los

pseudografos. Tomemos un pseudografo cualquiera H en \mathcal{G} y una arista e de H que no sea puente ni lazo. Puesto que la arista e puede fallar con probabilidad $1 - p$ o no fallar con probabilidad p , por la fórmula de probabilidad total se sigue que

$$R_H(p) = (1 - p) \cdot R_{H-e}(p) + p \cdot R_{H*e}(p). \quad (3.2)$$

La Ecuación (3.2) refleja que la confiabilidad de cualquier pseudografo satisface la propiedad de arista contracción-sustracción. Por lo tanto, el polinomio $R_G(p)$ es una invariante de Tutte-Grothendieck. Aplicando el teorema receta con $a = 1 - p$, $b = p$, $x_0 = p$ e $y_0 = 1$ se tiene que

$$R_H(p) = p^{r(H)}(1 - p)^{n(H)}T_H\left(1, \frac{1}{1 - p}\right) \quad (3.3)$$

Finalmente, sea G un grafo cualquiera en $\mathcal{C}_{n,m}$. Sabemos que su rango es $n - 1$ mientras que su co-rango $m - n + 1$, por lo que tras reemplazar en la Ecuación (3.3) se sigue el resultado de la Ecuación (3.1), como queríamos demostrar. \square

El Lema 3 establece una estrecha relación entre el polinomio confiabilidad de un grafo y su polinomio de Tutte. A continuación, en la Sección 3.2 se introduce una relación de orden parcial en cada clase $\mathcal{C}_{n,m}$ utilizando el concepto de polinomio de Tutte. Como consecuencia del Lema 3 resultará que, si dicha relación de orden parcial tiene máximo, entonces tendremos un UMRG.

3.2. Grafos Tutte-máximos

En esta sección se presenta el concepto de *grafo Tutte-máximo* tal como fue definido originalmente por Kahl y Luttrell [9]. Asimismo, se presenta un resultado fundamental para nuestro trabajo, el cual establece una relación directa entre los UMRG y los grafos Tutte-máximos de la clase $\mathcal{C}_{n,m}$.

Definición 3.2.1 (Kahl y Luttrell [9]). Dados dos grafos G y H en $\mathcal{C}_{n,m}$, denotamos $H \preceq G$ cuando $T_G(x, y) - T_H(x, y) = (x + y - xy)P(x, y)$ para algún polinomio $P(x, y)$ con coeficientes reales no negativos. En tal caso, diremos que G *domina* a H en la jerarquía de Tutte.

Definición 3.2.2 (Kahl y Luttrell [9]). Un grafo G en $\mathcal{C}_{n,m}$ es *Tutte-máximo* si para todo grafo H en $\mathcal{C}_{n,m}$ se cumple que $H \preceq G$.

Denotaremos $H \prec G$ cuando $H \preceq G$ y además H y G no son Tutte-equivalentes. En este caso diremos que G *domina estrictamente* a H en la jerarquía de Tutte.

En la Sección 3.1 se demostró que la confiabilidad de un grafo $R_G(p)$, es un invariante de Tutte-Grothendieck, y por lo tanto, puede expresarse en función del polinomio de Tutte. Usando este resultado, enunciaremos el Lema 4 que nos

permitirá establecer que, si un grafo G es Tutte-máximo entonces también es un UMRG dentro de su clase.

Lema 4. *Cada grafo Tutte-máximo en $\mathcal{C}_{n,m}$ es un UMRG en $\mathcal{C}_{n,m}$.*

Demostración. Sea G un grafo Tutte-máximo en $\mathcal{C}_{n,m}$ y sea H un grafo cualquiera en $\mathcal{C}_{n,m}$. Es claro que $R_G(0) = R_H(0)$ y que $R_G(1) = R_H(1)$. Sea p en $(0, 1)$. Basta con probar que $R_G(p) \geq R_H(p)$. Como G es Tutte-máximo, existe algún polinomio $P(x, y)$ con coeficientes no negativos tal que

$$T_G(x, y) - T_H(x, y) = (x + y - xy)P(x, y). \quad (3.4)$$

Aplicando el Lema 3 tanto a G como a H obtenemos que

$$\begin{aligned} R_G(p) - R_H(p) &= p^{n-1}(1-p)^{m-n+1} \left(T_G\left(1, \frac{1}{1-p}\right) - T_H\left(1, \frac{1}{1-p}\right) \right) \\ &= p^{n-1}(1-p)^{m-n+1} \cdot \left(1 + \frac{1}{1-p} - \frac{1}{1-p} \right) P\left(1, \frac{1}{1-p}\right) \\ &= p^{n-1}(1-p)^{m-n+1} \cdot P\left(1, \frac{1}{1-p}\right) \geq 0, \end{aligned}$$

donde se utilizó tanto la Ecuación (3.4) como el hecho de que $P(x, y)$ tiene todos sus coeficientes no negativos. \square

El Lema 4 es relevante para nuestro trabajo, ya que establece una relación entre la búsqueda de grafos Tutte-máximos y la existencia de UMRGs. En efecto, si logramos determinar un grafo Tutte-máximo en una clase entonces hemos obtenido un UMRG. Cabe aclarar que el recíproco del Lema 4 no es cierto (los autores Kahl y Luttrell señalan en [9] que en $\mathcal{C}_{7,11}$ existe UMRG pero no existe ningún grafo Tutte-máximo). A partir de este momento, el enfoque del trabajo se centrará en la búsqueda de un grafo Tutte-máximo de una clase, en lugar de abordar directamente la obtención del UMRG.

Para finalizar la sección cabe señalar que un grafo Tutte-máximo no solamente maximiza la confiabilidad sino que también maximiza simultáneamente otros invariantes de Tutte-Grothendieck que serán de utilidad.

Proposición 5 (Kahl y Luttrell [9]). *Cada grafo Tutte-máximo G maximiza simultáneamente todos los invariantes que se indican a continuación:*

- El número de árboles recubridores, y se evalúa $T_G(1, 1)$.
- El número de bosques recubridores, y se evalúa $T_G(2, 1)$.
- El número de subgrafos conexos recubridores, y se evalúa $T_G(1, 2)$.
- El número de orientaciones acíclicas, y se evalúa $T_G(2, 0)$.

3.3. Resultados Preliminares

El cálculo del polinomio de Tutte será esencial para el desarrollo de este proyecto. Dicho cálculo se simplifica considerablemente en algunas clases de grafos que se presentan en esta sección.

Cada una de las aristas de un árbol es puente y aporta un factor x al polinomio de Tutte. Como cada árbol con n vértices tiene exactamente $n - 1$ aristas, la siguiente observación es correcta.

Observación 6. Cada árbol G con n vértices cumple que $T_G(x, y) = x^{n-1}$.

Procedemos a hallar el polinomio de Tutte del ciclo C_n . Utilizaremos la convención de que las sumatorias (productorias) que no incluyen sumandos (factores) son iguales a 0 (resp. iguales a 1).

Lema 7. Para cada entero positivo n se cumple que $T_{C_n}(x, y) = y + \sum_{k=1}^{n-1} x^k$.

Demostración. Vamos a emplear el principio de inducción completa sobre n .

Por un lado, C_1 consta de un único lazo y $T_{C_1}(x, y) = y$. Por otro lado, $y + \sum_{k=1}^0 x^k = y$, por lo que el paso base es cierto.

Asumiendo que el enunciado es cierto para n , basta con probar que el enunciado es cierto para $n + 1$. Sea e una arista de C_{n+1} . Por un lado, $C_{n+1} * e$ es isomorfo a C_n y por hipótesis inductiva tenemos que $T_{C_{n+1} * e}(x, y) = T_{C_n}(x, y) = y + \sum_{k=1}^{n-1} x^k$. Por otro lado, $C_{n+1} - e$ es un árbol con $n + 1$ vértices y por la Observación 6 tenemos que $T_{C_{n+1} - e}(x, y) = x^n$. Como e no es ni lazo ni arista puente, concluimos que

$$T_{C_{n+1}}(x, y) = T_{C_{n+1} * e}(x, y) + T_{C_{n+1} - e}(x, y) = y + \sum_{k=1}^{n-1} x^k + x^n = y + \sum_{k=1}^n x^k.$$

Como es cierto tanto el paso base como el paso inductivo, el enunciado se sigue del principio de inducción completa. \square

El polinomio de Tutte factoriza por bloques [6].

Lema 8 (Ellis y Merino [6]). Si G_1, G_2, \dots, G_r son los bloques de un pseudografo G entonces $T_G(x, y) = \prod_{i=1}^r T_{G_i}(x, y)$.

En el artículo de Haggard et al. [7] se enuncia la Proposición 9. Los autores indican que su demostración se sigue tras emplear el principio de inducción completa sobre la cantidad de cadenas dentro de cada grafo 2-conexo y no incluyen una demostración. Dada la importancia de este resultado y por cuestiones de completitud, aquí se demuestra la Proposición 9.

Proposición 9 (Teorema 3 en [7]). Sean n y m enteros positivos tales que $m > n$. Para cada grafo G en $\mathcal{B}_{n,m}$ y cada cadena γ de G con k aristas se cumple que

$$T_G(x, y) = \left(\sum_{i=0}^{k-1} x^i \right) T_{G \ominus \gamma}(x, y) + T_{G * \gamma}(x, y). \quad (3.5)$$

Demostración. Vamos a emplear el principio de inducción completa sobre n .

Sea G un grafo en las condiciones del enunciado que tiene una cadena γ con tan solo una arista, es decir que $\gamma = \{e\}$. Como G es un grafo 2-conexo, G no tiene puntos de corte y en particular tampoco tiene aristas puente por lo que e no es un puente. Como G es un multigrafo, G no tiene lazos. Luego e no es ni un puente ni un lazo. Por la propiedad de arista contracción-sustracción del polinomio de Tutte de G aplicado a la arista e se tiene que:

$$T_G(x, y) = T_{G-e}(x, y) + T_{G*e}(x, y) = T_{G \ominus \gamma}(x, y) + T_{G*\gamma}(x, y),$$

por lo que el paso base es cierto.

Supongamos ahora que el enunciado se cumple cuando k es igual a h . Sea ahora un grafo G que cumple con las condiciones del enunciado que tiene alguna cadena γ con exactamente $h + 1$ aristas, por lo que $\gamma = \{e_1, e_2, \dots, e_{h+1}\}$. Como G es un multigrafo 2-conexo, sabemos que G no tiene ni puentes ni lazos. Aplicando la propiedad de arista contracción-sustracción del polinomio de Tutte de G aplicado a la arista e_{h+1} se tiene que:

$$T_G(x, y) = T_{G-e_{h+1}}(x, y) + T_{G*e_{h+1}}(x, y). \quad (3.6)$$

Observemos que, para cada $i \in \{1, 2, \dots, h\}$, la arista e_i es un puente en el grafo $G - e_{h+1}$. Por un lado, aplicando reiteradamente la definición recursiva del polinomio de Tutte a cada uno de los puentes del grafo $G - e_{h+1}$,

$$T_{G-e_{h+1}}(x, y) = x^h T_{G \ominus \gamma}(x, y). \quad (3.7)$$

Por otro lado, observemos que el grafo $G * e_{h+1}$ pertenece a $\mathcal{B}_{n-1, m-1}$. Además, $G * e_{h+1}$ tiene la cadena γ' definida como $\gamma - \{e_{h+1}\}$, cuyas aristas son $\{e_1, e_2, \dots, e_h\}$. Observemos que $(G * e_{h+1}) \ominus \gamma' = G \ominus \gamma$, mientras que $(G * e_{h+1}) * \gamma' = G * \gamma$. Aplicando la hipótesis inductiva al grafo $G * e_{h+1}$ se tiene que:

$$\begin{aligned} T_{G*e_{h+1}}(x, y) &= \left(\sum_{i=0}^{h-1} x^i \right) T_{(G*e_{h+1}) \ominus \gamma'}(x, y) + T_{(G*e_{h+1}) * \gamma'}(x, y) \\ &= \left(\sum_{i=0}^{h-1} x^i \right) T_{G \ominus \gamma}(x, y) + T_{G*\gamma}(x, y). \end{aligned} \quad (3.8)$$

Sustituyendo las ecuaciones (3.7) y (3.8) en (3.6), obtenemos:

$$\begin{aligned} T_G(x, y) &= x^h T_{G \ominus \gamma}(x, y) + \left(\sum_{i=0}^{h-1} x^i \right) T_{G \ominus \gamma}(x, y) + T_{G*\gamma}(x, y) \\ &= \left(\sum_{i=0}^h x^i \right) T_{G \ominus \gamma}(x, y) + T_{G*\gamma}(x, y). \end{aligned}$$

por lo que el enunciado es cierto cuando k es igual a $h + 1$ culminando así la prueba por la aplicación del principio de inducción completa. \square

Capítulo 4

Aportes

Este capítulo presenta los aportes algorítmicos de este proyecto de grado.

La Sección 4.1 presenta un algoritmo diseñado para calcular el polinomio de Tutte de cualquier pseudografo conexo, aprovechando tanto las propiedades de contracción y sustracción de cadenas como la factorización por bloques. A lo largo de esta sección se incluye el pseudocódigo correspondiente, se ofrece una demostración de su correctitud y terminación, y se realiza un análisis de su rendimiento mediante una comparación experimental con otros algoritmos.

La Sección 4.2 presenta un algoritmo de búsqueda de grafos Tutte-máximos dentro de cada una de las clases no vacías $\mathcal{C}_{n,m}$ de grafos conexos y simples. Se proporcionan resultados que permiten restringir el espacio de búsqueda únicamente a aquellos grafos en $\mathcal{C}_{n,m}^2$. Se diseña un algoritmo que combina esta reducción del espacio de búsqueda con el cálculo del polinomio de Tutte presentado en la Sección 4.1, logrando explotar las propiedades de contracción y sustracción de cadenas y de factorización por bloques. A lo largo de esta sección se incluye el pseudocódigo junto con una demostración de su correctitud y terminación.

Finalmente, la Sección 4.3 presenta las conclusiones generales del capítulo, resumiendo los principales aportes así como los resultados obtenidos.

4.1. Cálculo del polinomio de Tutte

A partir de la Observación 1 del Capítulo 2, sabemos que el cálculo del polinomio de Tutte es un problema #P-completo. El objetivo de esta sección consiste en desarrollar un algoritmo eficiente para el cálculo del polinomio de Tutte cuando restringimos nuestro universo de instancias de entrada a aquellos pseudografos conexos que poseen co-rango reducido. Para comenzar, presentamos un algoritmo básico que se desprende directamente de la definición del polinomio de Tutte. Seguidamente, presentamos un algoritmo más elaborado que será utilizado para cumplir con los objetivos propuestos en este proyecto.

4.1.1. Algoritmo Básico

El algoritmo *tutte_poly_naif*(G) sigue un esquema goloso para implementar el cálculo del polinomio de Tutte basado en su definición.

Algoritmo *tutte_poly_naif*(G):

Entrada: Un pseudografo conexo G
Salida: El polinomio de Tutte $T_G(x, y)$

```

1 if  $G$  no tiene aristas then
2   return 1;
3 if  $e$  es un puente en  $G$  then
4   return  $x \cdot \textit{tutte\_poly\_naif}(G * e)$ ;
5 if  $e$  es un lazo en  $G$  then
6   return  $y \cdot \textit{tutte\_poly\_naif}(G - e)$ ;
7 else
8   return  $\textit{tutte\_poly\_naif}(G - e) + \textit{tutte\_poly\_naif}(G * e)$ ;

```

La Figura 4.1 ilustra un árbol que representa los primeros tres pasos recursivos del Algoritmo *tutte_poly_naif*($\theta_{2,2,2}$). Estos grafos se obtienen de realizar contracciones y sustracciones de aristas. Es posible apreciar que la cantidad de llamados recursivos del algoritmo básico ante un grafo de entrada G es de orden exponencial con respecto a la cantidad de aristas del grafo del mismo, lo que torna prohibitiva su aplicación ante grafos de gran tamaño.

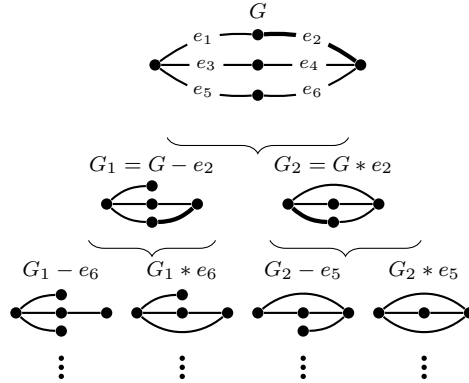


Figura 4.1: Aplicación del algoritmo básico al grafo $\theta_{2,2,2}$.

4.1.2. Algoritmo Propuesto

El algoritmo que se presenta en esta sección calcula el polinomio de Tutte $T_G(x, y)$ de un pseudografo conexo G . Este algoritmo explota propiedades estructurales del grafo, lo que permite una reducción significativa en el orden de cómputo en comparación con el algoritmo *tutte_poly_naif*(G).

Antes de presentar su pseudocódigo es necesario definir el concepto de *pseudografo etiquetado*.

Definición 4.1.1 (*Pseudografo etiquetado*). Para cada pseudografo G cuyo conjunto de aristas es $E(G) = \{e_1, e_2, \dots, e_r\}$ y cada tupla de enteros positivos (x_1, x_2, \dots, x_r) , denotamos $G(x_1, x_2, \dots, x_r)$ al pseudografo obtenido de G tras subdividir $x_i - 1$ veces a la arista e_i donde $i \in \{1, \dots, r\}$. Diremos que $G(x_1, x_2, \dots, x_r)$ es el *pseudografo etiquetado* G con etiquetas (x_1, x_2, \dots, x_r) .

A modo de ejemplo, el grafo $\theta_{1,1,1}(\ell_1, \ell_2, \ell_3)$ es el grafo $\theta_{\ell_1, \ell_2, \ell_3}$. En la figura 4.2 se presenta el grafo G definido como $\theta_{1,1,1}$ y el grafo $G(2, 2, 4)$.

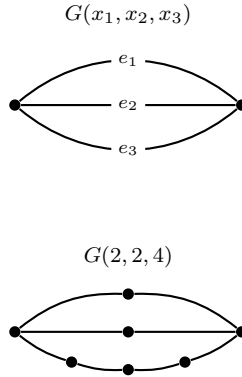


Figura 4.2: Ejemplo de un pseudografo etiquetado.

A continuación, presentaremos el Algoritmo *tutte_poly*(G). Este toma como entrada un pseudografo etiquetado G y retorna su polinomio de Tutte $T_G(x, y)$. En lo que sigue, se detalla el pseudocódigo correspondiente. El Algoritmo *tutte_poly*(G) sigue una estructura recursiva que, frente a una nueva entrada, analiza cuatro posibles condiciones de retorno. El **Bloque 1** (líneas 1-2) detecta si G es un árbol de n vértices y en tal caso retorna x^{n-1} . El **Bloque 2** (líneas 3-6) detecta si G tiene un lazo, que llamamos e . En dicho caso, retorna el producto entre el polinomio de Tutte del ciclo $T_{C_a}(x, y)$ (donde a es valor de la etiqueta de e) y la llamada recursiva con entrada $G - e$. El **Bloque 3** (líneas 7-9) detecta si G tiene más de un bloque, y en tal caso retorna el producto de los polinomios de Tutte de cada uno de sus bloques. El **Bloque 4** (líneas 10-12) detecta una cadena de G y retorna el polinomio de Tutte de G según la Proposición 9.

Algoritmo `tutte_poly(G)`:

Entrada: Un pseudografo conexo y etiquetado G

Salida: El polinomio de Tutte $T_G(x, y)$ parametrizado en las etiquetas de las aristas

```

1 if  $G$  es un árbol con  $n$  vértices then
2   return  $x^{n-1}$ ;
3 if  $G$  contiene un lazo  $e$ , etiquetado con  $a$  then
4    $T_1 \leftarrow y + \sum_{k=1}^{a-1} x^k$ ;
5    $G' \leftarrow G - e$ ;
6   return  $T_1 \cdot \text{tutte\_poly}(G')$ ;
7 if  $G$  tiene  $k$  bloques then
8   Tomar los  $k$  bloques  $G_1, \dots, G_k$  de  $G$ ;
9   return  $\prod_{i=1}^k \text{tutte\_poly}(G_i)$ ;
10 else
11   //  $G$  es un multigrafo conexo, y contiene una cadena
12   etiquetada con  $a$ 
13    $\gamma \leftarrow \text{cadena}(G)$ ;
14   return  $\left( \sum_{i=0}^{a-1} x^i \right) \text{tutte\_poly}(G \ominus \gamma) + \text{tutte\_poly}(G * \gamma)$ ;

```

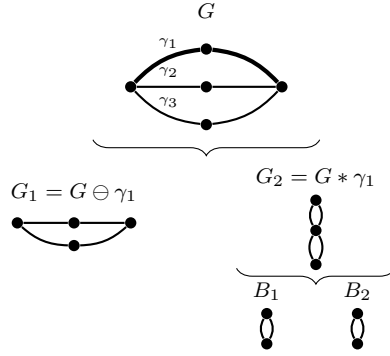


Figura 4.3: Aplicación del algoritmo propuesto al grafo $\theta_{2,2,2}$

Antes de adentrarnos en el estudio de la terminación y la correctitud, conviene analizar el ejemplo de la Figura 4.3 en contraste con el de la Figura 4.1. Ambos muestran el árbol recursivo del cálculo del polinomio de Tutte para el grafo $\theta_{2,2,2}$. Cabe observar la notable reducción en el número de operaciones necesarias cuando se emplea el Algoritmo `tutte_poly($\theta_{2,2,2}$)` en lugar del enfoque básico del Algoritmo `tutte_poly_naif($\theta_{2,2,2}$)`. En efecto, al aplicar `tutte_poly($\theta_{2,2,2}$)`, solo se requieren dos operaciones para alcanzar los casos base asociados al ciclo.

4.1.3. Terminación y Correctitud

En este apartado demostraremos la terminación y la correctitud del Algoritmo *tutte_poly(G)*. El siguiente concepto será esencial para el análisis de terminación.

Definición 4.1.2 (*Grafo reducido*). Dado un multigrafo conexo cualquiera G , decimos que G_r es un *grafo reducido* de G si cumple alguna de las siguientes condiciones:

1. $G_r = G \ominus \gamma$, donde γ es una cadena en G .
2. $G_r = G * \gamma$, donde γ es una cadena en G .
3. G no es 2-conexo y G_r es un bloque de G .

Lema 10. *Sea G un multigrafo conexo con alguna arista. Si G_r es un grafo reducido de G entonces se cumple que $|E(G_r)| < |E(G)|$.*

Demostración. Sea G un grafo en las condiciones del enunciado y sea G_r un grafo reducido de G . Por un lado, si G_r se obtiene mediante la aplicación de la regla 1 o 2 de la Definición 4.1.2 para alguna cadena γ de G entonces G_r tiene $|\gamma|$ aristas menos que G . Por otro lado, cada bloque de un grafo 2-conexo tiene al menos una arista. Luego, si G_r se obtiene mediante la aplicación de la regla 3 de la Definición 4.1.2 entonces G_r es uno de los bloques de G . En cualquiera de los casos, el reducido G_r de G tiene menos aristas que G , como queríamos demostrar. \square

Teorema 11 (Terminación del Algoritmo). *Dado un pseudografo G conexo con aristas etiquetadas, el Algoritmo *tutte_poly(G)* termina su ejecución en tiempo finito.*

Demostración. El Algoritmo *tutte_poly(G)* se compone de cuatro bloques. En particular, los Bloques 2, 3 y 4 son recursivos, siendo estos los que resulta de interés probar su terminación.

Observemos que en todos los pasos recursivos, el Algoritmo *tutte_poly(G)* retorna reducciones del grafo G . Apoyándonos en el Lema 10 podemos afirmar que $|E(G_r)| < |E(G)|$. Por lo tanto, en cada paso recursivo se llama nuevamente al Algoritmo *tutte_poly(G)* con un grafo con menos aristas. Puesto que el pseudografo de entrada G es finito, tras llamados recursivos obtendremos un grafo conexo minimal, que es un árbol, y se alcanzará su terminación en el Bloque 1. Por lo tanto, el algoritmo *tutte_poly(G)* finaliza tras una cantidad finita de pasos, como queríamos demostrar. \square

Estamos en condiciones de probar la correctitud del algoritmo *tutte_poly(G)*.

Teorema 12 (Correctitud del Algoritmo). *Dado un pseudografo G con aristas etiquetadas, el polinomio retornado por *tutte_poly(G)* corresponde al polinomio de Tutte de G .*

Demostración. Para demostrar la correctitud del algoritmo estructuraremos la prueba en bloques, verificando que lo retornado en cada uno de ellos es válido y se ajusta a la hipótesis de los lemas utilizados.

- *Bloque 1:* en este caso el pseudografo G es un árbol con n vértices. Por la Observación 6, su polinomio de Tutte es x^{n-1} , que es efectivamente lo que retorna este bloque, por lo que es correcto.
- *Bloque 2:* Sea G el pseudografo etiquetado de entrada, y sea X su conjunto de etiquetas. En este caso, G contiene un lazo e etiquetado con el valor a . Luego, existe un pseudografo G_1 tal que $G(X) = G_1 \cdot C_a$. Por el Lema 8, $T_{G(X)}(x, y) = T_{G_1}(x, y) \cdot T_{C_a}(x, y)$. Por el Lema 7 se tiene que $T_{G(X)}(x, y) = T_{G'}(x, y) \cdot \left(y + \sum_{k=1}^{a-1} x^k\right)$. Esta expresión coincide con el resultado retornado por el algoritmo, y es correcto por los Lemas 7 y 8.
- *Bloque 3:* en este caso G contiene k bloques. Bajo estas hipótesis, el algoritmo hace uso del Lema 8 y retorna el producto del polinomio de Tutte de cada uno de los bloques del grafo G .
- *Bloque 4:* este bloque toma efecto únicamente cuando los Bloques 1, 2, o 3 no tomaron efecto. Es simple probar que el pseudografo de entrada al Bloque 4 tiene más aristas que vértices y además pertenece a la clase $\mathcal{B}_{n,m}$. El polinomio de Tutte que se retorna en la línea 12 de este bloque es correcto por la Proposición 9.

La correctitud del algoritmo se sigue su terminación y de la correctitud de cada uno de sus bloques. \square

4.1.4. Eficiencia computacional

En este apartado analizamos cuán eficiente resulta nuestra implementación para calcular el polinomio de Tutte. Para ello, se realizó una comparación directa con dos algoritmos: por un lado, el algoritmo BHK [1], considerado entre los más eficientes hasta la fecha, y por otro lado, una versión del Algoritmo básico (*tutte_poly_naif*(G)) desarrollada en Python.

Para la implementación en código del algoritmo descrito en *tutte_poly*(G), se optó por utilizar el lenguaje Python, en su versión 3.12. Esta elección se basó principalmente en dos motivos: por un lado, la amplia disponibilidad de bibliotecas especializadas tanto para el tratamiento de polinomios como para la manipulación de grafos; por otro, la simplicidad del lenguaje, que permite desarrollar prototipos funcionales con pocas líneas de código, evitando así la complejidad asociada al manejo manual de memoria u otros aspectos de bajo nivel.

Todas las ejecuciones experimentales se realizaron en un entorno controlado, utilizando un ordenador con las siguientes especificaciones técnicas:

- **Procesador:** Intel Core i3-1215U (6 núcleos).
- **Memoria RAM:** 8 GB DDR4 a 3200 MHz.
- **Sistema operativo:** Windows 11 Home.
- **Entorno de ejecución:** Windows Subsystem for Linux (WSL)

Para llevar a cabo las pruebas computacionales, se seleccionaron dos familias distintas de grafos: grafos completos y grafos $\theta_{l_1, l_1, l_1, l_1}$. A modo de ejemplo, se presentan las Figuras 4.4 y 4.5, que ilustran un grafo representativo de cada familia utilizada en la comparación.

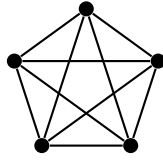


Figura 4.4: Grafo completo de 5 vértices.

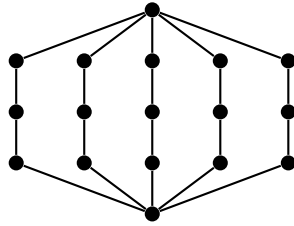


Figura 4.5: Grafo $\theta_{4,4,4,4}$

Las Tablas 4.1 y 4.2 resumen los resultados obtenidos por cada uno de los algoritmos estudiados ante instancias de grafos completos con 2, 3, 4, 5, o 6 vértices, y grafos de la clase $\theta_{\ell_1, \ell_1, \ell_1, \ell_1}$ con $\ell_1 \in \{4, 5, 8, 10, 15\}$, respectivamente. Vale destacar que en cada caso se realizó una gran cantidad de corridas preliminares, en las cuales se observó consistentemente resultados acordes a los presentados a continuación. Los resultados mostrados en las tablas corresponden a una única corrida representativa, y no a un promedio de las corridas realizadas.

Cantidad de vértices	BHKK (s)	Algoritmo propuesto (s)	Algoritmo básico (s)
2	0.007500	0.000482	0.000172
3	0.006845	0.001211	0.003750
4	0.006113	0.019639	0.011367
5	0.006043	0.818198	0.050758
6	0.008538	72.932995	0.480127
7	0.015731	N/A	10.74871

Tabla 4.1: Tiempos de ejecución para grafos completos de n vértices.

Largo de cadenas	BHKK (s)	Algoritmo propuesto (s)	Algoritmo básico (s)
4	3.755088	0.275767	1.051911
5	257.659786	0.467448	2.979167
8	N/A	0.436766	20.559151
10	N/A	0.496000	61.231551
15	N/A	0.912857	627.110134

Tabla 4.2: Tiempos de ejecución de los tres algoritmos para $\theta_{l_1, l_1, l_1, l_1, l_1}$

La Tabla 4.1 presenta los resultados para grafos completos. Aquí, nuestro algoritmo enfrenta su mayor desafío, ya que no existen puntos de corte ni cadenas de largo mayor que uno. Aunque nuestro algoritmo inicia con tiempos competitivos, a partir de K_6 los tiempos de ejecución se incrementan considerablemente. Para el grafo K_7 su ejecución es finalizada de forma forzada por el sistema sin retornar un polinomio debido al excesivo uso de memoria. En cambio, tanto el Algoritmo básico como el algoritmo BHKK muestran comportamientos más estables.

El otro caso analizado (Tabla 4.2), corresponde a los grafos $\theta_{\ell_1, \ell_1, \ell_1, \ell_1, \ell_1}$. En este escenario, nuestro algoritmo demuestra su máximo potencial. Supera ampliamente a los otros enfoques, manteniendo tiempos acotados incluso en instancias con más de 30 vértices, donde BHKK ni siquiera puede operar. En contraste, el Algoritmo básico sufre un notable incremento en el tiempo en función del crecimiento en el largo de las cadenas.

En resumen, nuestro algoritmo se desempeña con gran eficiencia y escalabilidad en grafos de bajo co-rango, donde puede explotar plenamente su estrategia de optimización. Sin embargo, su rendimiento se ve comprometido en grafos de alto co-rango, en los cuales la ausencia de cadenas y bloques limita su eficacia y conduce a un uso excesivo de recursos.

4.2. Búsqueda de grafos Tutte-máximos

El objetivo de esta sección es presentar un algoritmo que permita determinar la existencia de grafos Tutte-máximos dentro de $\mathcal{C}_{n,m}$ y, en caso de existir, retornar uno de ellos.

Esta sección se desarrolla en dos etapas. En primer lugar, mostramos que

es posible restringir la búsqueda de grafos Tutte-máximos en $\mathcal{C}_{n,m}$ a la clase $\mathcal{C}_{n,m}^2$. Adicionalmente, determinaremos un conjunto de grafos de menor cardinal que, a partir de subdivisiones de sus aristas, permite representar a cada uno $\mathcal{C}_{n,m}^2$. En segundo lugar, presentamos el pseudocódigo del algoritmo de búsqueda. Por último, demostraremos la terminación y la correctitud del algoritmo desarrollado.

4.2.1. Grafos 2-conexos y destilaciones

En este apartado mostraremos por qué podemos restringir el universo de búsqueda de grafos Tutte-máximos en $\mathcal{C}_{n,m}$ a grafos en $\mathcal{C}_{n,m}^2$. Luego, veremos que para cada grafo en $\mathcal{C}_{n,m}^2$ existe un grafo destilación D en \mathcal{M}_c y un conjunto de etiquetas X tal que $G = D(X)$. Este hecho permitirá reducir notoriamente el esfuerzo computacional del algoritmo de búsqueda de grafos Tutte-máximos.

Teorema 13 (Kahl y Luttrell [9]). *Sea G un grafo en $\mathcal{C}_{n,m} - \mathcal{C}_{n,m}^2$ tal que $m > n$ y sean G_1 y G_2 dos bloques de G . Supongamos además que v es un vértice de corte común a ambos bloques, y que $uv \in E(G_1)$ y $vw \in E(G_2)$. Sea H un grafo tal que*

$$H = G - uv + vw.$$

Entonces se tiene que $G \preceq H$. Además, si $G_1 \neq K_2$ entonces $G \prec H$.

Proposición 14. *Sean n y m dos enteros positivos tales que $m > n$. Para cada grafo G en $\mathcal{C}_{n,m} - \mathcal{C}_{n,m}^2$ existe otro grafo H en $\mathcal{C}_{n,m}^2$ tal que $G \prec H$.*

Demostración. Sea G un grafo en las condiciones del enunciado. Notemos que al menos uno de sus bloques no es P_2 , pues si cada bloque fuese P_2 entonces G sería un árbol, contradiciendo que G tiene más aristas que vértices. Sea G_1 un bloque de G que no sea P_2 , y sea G_2 otro bloque de G que tenga un vértice en común con G_1 . Por el Teorema 13 sabemos que existe G' tal que $G \prec G'$. Si G' aún presenta algún punto de corte podemos aplicar nuevamente el Teorema 13 y repetir el procedimiento hasta llegar a un grafo H sin puntos de corte tal que $G \prec H$. Nótese además que H pertenece a $\mathcal{C}_{n,m}$, no tiene puntos de corte, y tiene al menos 3 vértices, por lo que H es 2-conexo y pertenece a $\mathcal{C}_{n,m}^2$ como queríamos demostrar. \square

La Proposición 14 permite restringir el espacio de búsqueda de grafos Tutte-máximos de $\mathcal{C}_{n,m}$ a aquellos grafos en $\mathcal{C}_{n,m}^2$. A continuación, mostraremos que es posible obtener a cada uno de los grafos en $\mathcal{C}_{n,m}^2$ mediante subdivisiones elementales de alguno de los grafos en \mathcal{M}_c , reduciendo así el cardinal del espacio de búsqueda. Para cumplir este objetivo, es necesario introducir el Lema 15.

Lema 15 (Romero [16]). *Sean n y m dos enteros positivos tales que $n + 1 \leq m \leq 3n/2$ y sea G en $\mathcal{C}_{n,m}^2$. Sea $c = m - n + 1$. Cada una de las siguientes afirmaciones es cierta.*

1. *La destilación de G , que denotamos $D(G)$, tiene a lo sumo $2c - 2$ vértices y para todo $v \in V(D(G))$ el grado de v es mayor que o igual a 3.*

2. Si $D(G)$ tiene exactamente $2c - 2$ vértices entonces $D(G)$ es cúbico.
3. El grafo $D(G)$ tiene no más que $3c - 3$ aristas. La igualdad ocurre si y solo si $D(G)$ es cúbico.

Observemos que si un grafo 2-conexo simple G tiene más aristas que vértices entonces existe su destilación. Puesto que G es 2-conexo y se obtiene mediante una cantidad finita de subdivisiones elementales de $D(G)$, resulta que $D(G)$ es también 2-conexo. Por lo tanto, $D(G)$ es un multigrafo y no puede tener lazos.

Observación 16. La destilación de cada grafo 2-conexo simple con más aristas que vértices es un multigrafo 2-conexo.

Estamos en condiciones de probar el siguiente lema.

Lema 17. Sean n y m dos enteros positivos tales que $n + 1 \leq m \leq 3n/2$. Definamos $c = m - n + 1$. Para cada G en $\mathcal{C}_{n,m}^2$ existe un multigrafo G' en \mathcal{M}_c tal que $D(G) = D(G')$.

Demostración. Sean n y m y G en las condiciones del enunciado. Por la Observación 16, existe la destilación $D(G)$ de G , que es un multigrafo. Por el Lema 15(1) sabemos que $D(G)$ tiene a lo sumo $2c - 2$ vértices. Si $D(G)$ tiene exactamente $2c - 2$ vértices entonces por el Lema 15(2) tenemos que $D(G)$ es cúbica y por lo tanto tiene $3c - 3$ aristas. En este caso $D(G)$ pertenece al conjunto \mathcal{M}_c y basta con elegir $G' = G$. Si no, como el co-rango de G y de $D(G)$ coinciden, tendremos que $D(G)$ es un multigrafo con n' vértices y $n' + (c - 1)$ aristas para algún entero n' tal que $n' < 2c - 2$. Definamos $k = 2c - 2 - n'$. Tomemos una arista arbitraria e de $D(G)$. Construyamos G' que se obtiene de G tras aplicar k subdivisiones elementales de la arista e . Observemos que G' es un multigrafo que tiene exactamente k vértices y k aristas más que $D(G)$, por lo que tiene exactamente $2c - 2$ vértices y $3c - 3$ aristas. Entonces, G' es un multigrafo en \mathcal{M}_c y su destilación es precisamente $D(G)$, como queríamos demostrar. \square

Observación 18. Sean n y m enteros positivos tales que $n + 1 \leq m \leq \frac{3n}{2}$, y sea $c = m - n + 1$. Si D es una destilación en \mathcal{M}_c entonces existe $G \in \mathcal{B}_{2c-2, 3c-3}$ que puede obtenerse mediante subdivisiones elementales de D .

Al combinar la Proposición 14, el Lema 17 y la Observación 18, es posible restringir el espacio de búsqueda de grafos Tutte-máximos de $\mathcal{C}_{n,m}$ únicamente a aquellas destilaciones provenientes de grafos en $\mathcal{B}_{2c-2, 3c-3}$. Por último, introducimos la relación de dominancia entre polinomios de Tutte que será utilizada en el algoritmo de búsqueda.

Definición 4.2.1. Para cada par de enteros n y m tales que $\mathcal{C}_{n,m}$ es no vacía, definimos la colección de polinomios $\mathcal{P}_{n,m}$ como $\{T_G(x, y) : G \in \mathcal{C}_{n,m}\}$.

Definición 4.2.2. Sean $T_G(x, y)$ y $T_H(x, y)$ dos polinomios en $\mathcal{P}_{n,m}$. Decimos que T_G domina a T_H , y lo denotamos $T_H \preceq T_G$, cuando $T_G(x, y) - T_H(x, y) = (x + y - xy)P(x, y)$ para algún polinomio $P(x, y)$ con coeficientes no negativos.

Observación 19. La relación de precedencia \preceq en el conjunto $\mathcal{P}_{n,m}$ es de orden parcial.

- **Reflexividad:** Para todo polinomio $T_G(x, y)$ se cumple $T_G(x, y) - T_G(x, y) = (x + y - xy) \cdot 0 = 0$, por lo que $T_G \preceq T_G$.
- **Antisimetría:** Si $T_G \preceq T_H$ y $T_H \preceq T_G$, entonces existen dos polinomios $P_1(x, y), P_2(x, y)$ con coeficientes no negativos tales que:

$$T_H(x, y) - T_G(x, y) = (x + y - xy) P_1(x, y) \quad (4.1)$$

$$T_G(x, y) - T_H(x, y) = (x + y - xy) P_2(x, y) \quad (4.2)$$

Sumando las expresiones (4.1) y (4.2) se obtiene que $0 = (x + y - xy)(P_1(x, y) + P_2(x, y))$. Luego, $P_1(x, y) + P_2(x, y) = 0$. Como $P_1(x, y)$ y $P_2(x, y)$ tienen coeficientes no negativos, la única posibilidad es que $P_1(x, y) = P_2(x, y) = 0$, por lo que $T_G = T_H$.

- **Transitividad:** Si $T_H \preceq T_{G'}$ y $T_{G'} \preceq T_G$, entonces existen dos polinomios $P_1(x, y), P_2(x, y)$ con coeficientes no negativos tales que:

$$T_{G'}(x, y) - T_H(x, y) = (x + y - xy) P_1(x, y) \quad (4.3)$$

$$T_G(x, y) - T_{G'}(x, y) = (x + y - xy) P_2(x, y) \quad (4.4)$$

Sumando las expresiones (4.3) y (4.4) se obtiene que $T_G(x, y) - T_H(x, y) = (x + y - xy)(P_1(x, y) + P_2(x, y))$, donde $P_1(x, y) + P_2(x, y)$ también tiene coeficientes no negativos. Por tanto, $T_H \preceq T_G$.

El siguiente concepto de polinomio máximo en $\mathcal{P}_{n,m}$ tiene una correspondencia directa con el de grafo Tutte-máximos en $\mathcal{C}_{n,m}$.

Definición 4.2.3. Decimos que un polinomio T_G en $\mathcal{P}_{n,m}$ es *máximo* si para todo polinomio T_H en $\mathcal{P}_{n,m}$ se cumple que $T_H \preceq T_G$.

Observación 20. A partir de la Observación 19 sabemos que si existe un polinomio máximo entonces es único.

Observación 21. Si G es un grafo Tutte-máximo en $\mathcal{C}_{n,m}$ entonces T_G es polinomio máximo en $\mathcal{P}_{n,m}$.

Estamos en condiciones de presentar el algoritmo de búsqueda de grafos Tutte-máximos en la clase $\mathcal{C}_{n,m}$.

4.2.2. Algoritmo Propuesto

Algoritmo `search_tutte(n, m)`:

Entrada: Dos enteros positivos n y m tales que $n + 1 \leq m \leq \frac{3n}{2}$

Salida: Grafo Tutte-máximo en $\mathcal{C}_{n,m}$ si existe; \emptyset si no.

```

1  $c \leftarrow m - n + 1$ ;
2  $\mathcal{M}_c \leftarrow \text{obtener\_destilaciones}(c)$ ;
3  $\mathcal{T} \leftarrow \emptyset$ ;
4 for  $D \in \mathcal{M}_c$  do
5    $D(X) \leftarrow \text{etiquetar}(D)$ ;
6    $T_{D(X)}(x, y) \leftarrow \text{tutte\_poly}(D(X))$ ;
7    $\mathcal{L} \leftarrow \left\{ (l_1, l_2, \dots, l_{|E(D)|}) \in \mathbb{Z}_+^{|E(D)|} : \sum_{i=1}^{|E(D)|} l_i = m \right\}$ ;
8   for  $l \in \mathcal{L}$  do
9     if  $T_{D(l)}(x, y) \notin \mathcal{T}$  then
10       $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_{D(l)}(x, y)\}$ ;
11  $P_1 \leftarrow \{T_G \in \mathcal{T} : \nexists T_H \in \mathcal{T}, T_H(1, 1) > T_G(1, 1)\}$ ;
12  $P_2 \leftarrow \{T_G \in \mathcal{T} : \nexists T_H \in \mathcal{T}, T_H(2, 1) > T_G(2, 1)\}$ ;
13  $P_3 \leftarrow \{T_G \in \mathcal{T} : \nexists T_H \in \mathcal{T}, T_H(1, 2) > T_G(1, 2)\}$ ;
14  $P_4 \leftarrow \{T_G \in \mathcal{T} : \nexists T_H \in \mathcal{T}, T_H(2, 0) > T_G(2, 0)\}$ ;
15  $P' \leftarrow P_1 \cap P_2 \cap P_3 \cap P_4$ ;
16 if  $P' == \emptyset$  then
17   return  $\emptyset$ ;
18  $T_H \leftarrow \text{obtener\_un\_elemento\_cualquiera\_de}(P')$ ;
19 for  $T_G \in P'$  do
20   if  $T_H \prec T_G$  then
21      $T_H \leftarrow T_G$ ;
22 for  $T_G \in \mathcal{T}, T_G \neq T_H$  do
23   if not  $T_G \preceq T_H$  then
24     return  $\emptyset$ ;
25 return  $H$ ;

```

El Algoritmo de búsqueda `search_tutte(n, m)` obtiene el conjunto de destilaciones \mathcal{M}_c en el **Bloque 1** (líneas 1 - 2). En la práctica, este conjunto se determina a partir de cada uno de los grafos en $\mathcal{B}_{2c-2, 3c-3}$, obteniendo sus destilaciones y luego eliminando aquellas que son isomorfas. En el **Bloque 2** (líneas 3 - 10), se obtienen de manera eficiente todos los polinomios de Tutte de los grafos en $\mathcal{C}_{n,m}^2$. Notar que basta con calcular el polinomio de Tutte solamente para las destilaciones etiquetadas. Luego, evaluando las diferentes combinaciones válidas de etiquetas se obtienen todos los polinomios de Tutte de los grafos en $\mathcal{C}_{n,m}^2$. Dichos polinomios son almacenados en el conjunto \mathcal{T} . El **Bloque 3** (líneas 11 - 17) corresponde al proceso de búsqueda de polinomios candidatos a ser máximos, partiendo del conjunto \mathcal{T} . En primer lugar, el algoritmo evalúa los cuatro invariantes de Tutte-Grothendieck presentados en

la Proposición 5 para cada uno de los polinomios del conjunto \mathcal{T} . A partir de ello, identifica el conjunto de aquellos polinomios de Tutte cuyos grafos maximizan cada invariante. Luego, se obtiene la intersección de esos cuatro conjuntos. Si dicha intersección contiene más de un polinomio, el algoritmo determina si un candidato domina a los restantes (líneas 18 - 21). Notar que T_H , al finalizar el algoritmo, almacenará el polinomio máximo en $\mathcal{C}_{n,m}$ en caso de existir. Finalmente, en el **Bloque 4** (líneas 22 - 25) se verifica computacionalmente si T_H domina al resto de los polinomios de Tutte de los grafos en $\mathcal{C}_{n,m}^2$, y de ser así, retorna su grafo asociado.

4.2.3. Terminación y Correctitud

En esta sección enunciaremos y demostraremos dos teoremas. El primero está relacionado con la terminación del Algoritmo `search_tutte(n, m)`, mientras que el segundo aborda su correctitud.

Teorema 22 (Terminación del Algoritmo). *El Algoritmo `search_tutte(n, m)` siempre termina su ejecución en tiempo finito.*

Demostración. Observemos que cada bloque del algoritmo opera sobre conjuntos finitos y realiza un número finito de operaciones:

- *Bloque 1:* Como el conjunto $\mathcal{B}_{2c-2,3c-3}$ es finito, se tiene en particular que el conjunto \mathcal{M}_c es finito. Asimismo, la determinación computacional de \mathcal{M}_c se lleva a cabo en un número finito de operaciones, puesto que se construye mediante el colapso de los vértices de grado 2 en cada grafo de $\mathcal{B}_{2c-2,3c-3}$, cuyo conjunto de vértices es, por definición, finito.
- *Bloque 2:* Por el Teorema 11, el cálculo del polinomio de Tutte a través del Algoritmo `tutte_poly(G)` termina en tiempo finito. Además, para cada destilación $D \in \mathcal{M}_c$ y cada entero positivo m se tiene que la cantidad de combinaciones válidas de etiquetas positivas cuya suma es exactamente igual a m es finita.
- *Bloques 3 y 4:* La evaluación del polinomio de Tutte para el cálculo de invariantes, la búsqueda del valor máximo dentro de un conjunto finito y la comparación de polinomios son todas operaciones computables en tiempo finito.

La terminación del algoritmo en una cantidad finita de pasos se sigue de la terminación de cada uno de sus bloques en una cantidad finita de pasos. \square

Teorema 23 (Correctitud del Algoritmo). *Si existe al menos un grafo Tutte-máximo en $\mathcal{C}_{n,m}$ entonces el Algoritmo `search_tutte(n, m)` retorna uno de ellos.*

Demostración. Llamaremos G a algún grafo Tutte-máximo de $\mathcal{C}_{n,m}$.

- *Bloque 1:* Por la Proposición 14, $G \in \mathcal{C}_{n,m}^2$. Adicionalmente, por la Proposición 17, existe $D \in \mathcal{M}_c$ y $X_G = (x_1, x_2, \dots, x_{|E(D)|}) \in \mathbb{Z}_+^{|E(D)|}$ tal

que $D(x_1, x_2, \dots, x_{|E(D)|}) = G$. Por la Observación 18, \mathcal{M}_c puede obtenerse a partir de subdivisiones partiendo de un grafo en $\mathcal{B}_{2c-2, 3c-3}$.

- *Bloque 2:* Observemos que X_G pertenece al conjunto de etiquetas \mathcal{L} que fue generado en la línea 7 del algoritmo. En consecuencia, una vez finalizada la ejecución del *Bloque 2*, el polinomio T_G formará parte del conjunto \mathcal{T} construido en la línea 10.
- *Bloque 3:* Por la Proposición 5 se sigue que T_G pertenece a la intersección de los conjuntos P' presentada en la línea 15. Además, por las Observaciones 20 y 21, T_G será el único máximo seleccionado en el proceso comparativo descrito en las líneas 19 - 21.
- *Bloque 4:* Como T_G es máximo, no se alcanzará la línea 24 y el algoritmo retornará un grafo H cuyo polinomio de Tutte es precisamente T_G . Dado que G es Tutte-máximo y H es Tutte-equivalente a G , se puede afirmar que H también es Tutte-máximo.

Luego el algoritmo retorna un grafo Tutte-máximo, por lo que es correcto. \square

4.3. Conclusiones del Capítulo

En este capítulo se han presentado dos contribuciones que fortalecen el cuerpo central de este trabajo.

En primer lugar, se desarrolló un algoritmo eficiente para el cálculo del polinomio de Tutte en grafos de co-rango bajo que aprovecha las propiedades de contracción y sustracción de cadenas así como también la factorización por bloques. Este enfoque permite reducir significativamente el tiempo de ejecución computacional al tratar el conjunto de las aristas pertenecientes a una misma cadena como una unidad en lugar de procesarlas individualmente. Los resultados experimentales demuestran que esta implementación mantiene tiempos de ejecución estables y competitivos en grafos con estructura de cadenas largas.

En segundo lugar, se desarrolló un algoritmo para la identificación de grafos Tutte-máximos optimizado para operar en clases de grafos de co-rango bajo. Este aporte metodológico combina, de manera innovadora, el cálculo eficiente del polinomio de Tutte con una reducción estratégica del espacio de búsqueda. La estrategia se basó en restringir el estudio a grafos en $\mathcal{C}_{n,m}^2$, representarlos mediante clases de menor cardinal y determinar candidatos mediante la maximización de invariantes de Tutte-Grothendieck.

En el Capítulo 5 pondremos en práctica los algoritmos desarrollados, comenzando por un estudio concreto de la existencia de grafos Tutte-máximos dentro de la clase $\mathcal{C}_{11,14}$. A partir de este análisis, se generalizan los resultados y se alcanza un aporte novedoso: la refutación de la afirmación de Wang para una cantidad infinita de clases de grafos.

Capítulo 5

Refutación a la afirmación de Wang

Este capítulo tiene como objetivo analizar la veracidad de las Afirmaciones 1 y 2. En la Sección 5.1 se determina que el grafo L^1 es Tutte-máximo dentro en $\mathcal{C}_{11,14}$. En la Sección 5.2 se prueba con asistencia computacional que $W^s \preceq L^s$ para cada entero positivo s .

5.1. Determinación de un grafo Tutte-máximo

En esta sección analizaremos las Afirmaciones 1 y 2 sobre una clase particular: $\mathcal{C}_{11,14}$. Esta clase resulta de especial interés por ser la clase con menor cantidad de vértices en la que los grafos L_n y W_n difieren.

Comenzaremos el análisis calculando los polinomios de Tutte de L^1 y W^1 . Por cuestiones de completitud se incluyen en el Anexo las expresiones generales de los polinomios T_{W^s} y T_{L^s} para cada entero positivo s , que se obtienen tras ejecuciones del Algoritmo `tutte.poly(W^s)` y `tutte.poly(L^s)`. Una vez obtenidos dichos polinomios, realizamos una comparación de los mismos cuando $s = 1$, conforme a la Definición 3.2.1. De esta manera obtenemos lo siguiente:

$$T_{L^1}(x, y) - T_{W^1}(x, y) = (x + y - xy)(x^5 + x^4 + x^3)$$

Como se puede observar, el polinomio $P(x, y)$ definido como $x^5 + x^4 + x^3$ presenta únicamente coeficientes reales no negativos. Por lo tanto,

$$W^1 \prec L^1.$$

El resultado anterior basta para refutar la Afirmación 1. Adicionalmente, resulta de interés verificar computacionalmente si la Afirmación 2 es válida.

Para realizar esta verificación se utilizará el Algoritmo [search_tutte\(\$n, m\$ \)](#). En caso de que exista un grafo Tutte-máximo en $\mathcal{C}_{11,14}$, dicho algoritmo retornará un grafo Tutte-máximo en $\mathcal{C}_{11,14}$. Luego, podremos compararlo con L^1 y determinar si son o no son Tutte-equivalentes. Al ejecutar el Algoritmo [search_tutte\(\$n, m\$ \)](#) con $n = 11$ y $m = 14$ se obtiene el grafo H que se ilustra en la Figura 5.1.

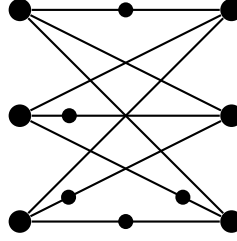


Figura 5.1: Grafo H en $\mathcal{C}_{11,14}$.

Obsérvese que el grafo Tutte-máximo H obtenido es isomorfo al grafo L^1 . En particular, esto implica directamente que L^1 es Tutte-máximo en $\mathcal{C}_{11,14}$.

Antes de concluir esta sección resulta pertinente analizar la eficiencia del Algoritmo [search_tutte\(\$n, m\$ \)](#) en el marco de este ejemplo. A partir de una revisión computacional exhaustiva es posible comprobar que hay exactamente 109844 grafos dentro de la clase $\mathcal{C}_{11,14}$. Por un lado, en caso de no aplicar ninguna técnica de restricción del espacio de búsqueda, se requeriría calcular exactamente 109.844 polinomios de Tutte para luego realizar una comparación entre dichos polinomios. Por otro lado, el Algoritmo [search_tutte\(\$n, m\$ \)](#) restringe el espacio de búsqueda al conjunto \mathcal{M}_4 que consiste en todas las destilaciones de grafos 2-conexos de co-rango 4 y tiene apenas 17 grafos. En consecuencia, a partir de la técnica de destilaciones etiquetadas fue suficiente calcular solamente 17 polinomios de Tutte en este caso. En el Anexo se presentan los 17 grafos dentro del conjunto \mathcal{M}_4 .

Para concluir esta sección, es conveniente destacar dos aportes centrales. En primer lugar verificamos computacionalmente que para la clase $\mathcal{C}_{11,14}$ el grafo propuesto por Landgren y Steif L^1 domina estrictamente en la jerarquía de Tutte al grafo W^1 propuesto por Wang. Este resultado nos permite descartar la validez de la Afirmación 1. En segundo lugar verificamos computacionalmente que L^1 es Tutte-máximo dentro de la clase $\mathcal{C}_{11,14}$, lo que respalda la Afirmación 2 en esta clase particular.

5.2. Comparación entre W^s y L^s

En esta sección probaremos que $W^s \prec L^s$ para cada entero positivo s . Esto es equivalente a probar que para cada entero positivo s se cumple que $T_{L^s} - T_{W^s} = (x + y - xy)P_s(x, y)$, donde $P_s(x, y)$ tiene coeficientes reales no negativos.

Tras ejecutar el Algoritmo $tutte_poly(L^s)$ y $tutte_poly(W^s)$ y expresar sumas geométricas truncadas mediante funciones racionales, se obtienen expresiones para $T_{L^s}(x, y)$ y $T_{W^s}(x, y)$ que se presentan en el Anexo. Tomando la resta $T_{L^s}(x, y) - T_{W^s}(x, y)$ se tiene que para cualquier entero positivo s y cualquier par de números reales x e y tales que $x \neq 1$,

$$T_{L^s}(x, y) - T_{W^s}(x, y) = \frac{y(-x^{3s} + 3x^{3s+1} - 3x^{3s+2} + x^{3s+3} + x^{5s+1} - 3x^{5s+2} + 3x^{5s+3} - x^{5s+4})}{(x-1)^3} + \frac{x^{3s+1} - 3x^{3s+2} + 3x^{3s+3} - x^{3s+4} - x^{5s+2} + 3x^{5s+3} - 3x^{5s+4} + x^{5s+5}}{(x-1)^4} \quad (5.1)$$

Lema 24. Para cada entero positivo s y cada número real x tal que $x \neq 1$,

$$T_{L^s}(x, y) - T_{W^s}(x, y) = (x + y - xy) \sum_{i=3s}^{5s} x^i. \quad (5.2)$$

Demostración. Tras tomar denominador común $(x-1)^4$ en el miembro derecho de la Ecuación (5.1) se deduce que

$$T_{L^s}(x, y) - T_{W^s}(x, y) = \frac{(-yx^{3s}(x-1) + x^{3s+1})(x-1)^3(x^{2s+1} - 1)}{(x-1)^4}.$$

Cancelando el factor $(x-1)^3$ que figura en el numerador y en el denominador del miembro derecho de la última expresión y operando, obtenemos que

$$\begin{aligned} T_{L^s}(x, y) - T_{W^s}(x, y) &= \left(-yx^{3s} + \frac{x^{3s+1}}{x-1}\right)(x^{2s+1} - 1) \\ &= (-y(x-1)x^{3s} + x^{3s+1}) \frac{x^{2s+1} - 1}{x-1} \\ &= (-y(x-1) + x) x^{3s} \sum_{i=0}^{2s} x^i \\ &= (x + y - xy) x^{3s} \sum_{i=0}^{2s} x^i \\ &= (x + y - xy) \sum_{i=3s}^{5s} x^i. \quad \square \end{aligned}$$

Estamos en condiciones de demostrar el siguiente resultado.

Proposición 25. Para cada entero positivo s se cumple que $W^s \preceq L^s$.

Demostración. Por el Lema 24 sabemos que para cualquier entero positivo s y para cualquier elección de números reales x e y tales que $x \neq 1$ se cumple que

$P_s(x, y) = \sum_{i=3s}^{5s} x^i$. Recordemos que dos funciones continuas que coinciden en un conjunto denso de \mathbb{R}^2 son idénticas. Como $\mathbb{R}^2 - \{(x, y) : x, y \in \mathbb{R}, x \neq 1\}$ es denso en \mathbb{R}^2 y los polinomios son continuos concluimos que para cada entero positivo s la expresión $P_s(x, y) = \sum_{i=3s}^{5s} x^i$ se cumple para todo (x, y) en \mathbb{R}^2 . \square

En esta sección hemos demostrado que L^s domina a W^s dentro de la jerarquía de Tutte. Este resultado refuta la Afirmación 1 para una cantidad infinita de clases de grafos. Si bien esto no garantiza que la Afirmación 2 sea válida, hay evidencia computacional para pensar que si podría serlo.

Capítulo 6

Conclusiones y trabajo futuro

El análisis de confiabilidad de redes se originó a mediados del siglo pasado con una motivación muy estrecha con la práctica: diseño de computadores masivos altamente confiables. En aquel entonces era indispensable conectar distintos puntos en un circuito eléctrico compuesto con relés y con una probabilidad de falla extremadamente baja (idealmente nula). Un problema natural consiste en maximizar la probabilidad de conexión entre distintos puntos, bajo una cantidad limitada de recursos (relés). No obstante, el estudio de la confiabilidad uniforme se formaliza mediante el lenguaje de la teoría de grafos recién en 1986 tras un influyente artículo publicado por Frank Boesch. A partir de esa fecha resultó de interés teórico y práctico la búsqueda de grafos uniformemente más confiables.

Boesch en un trabajo colectivo [2] prueba que para cada clase no vacía de grafos conexos $\mathcal{C}_{n,m}$ cuyo co-rango es positivo y no mayor que 3 existe un único grafo uniformemente más confiable. Curiosamente, hasta la fecha desconocemos si en cada clase no vacía $\mathcal{C}_{n,m}$ cuyo co-rango es 4 existe un único grafo uniformemente más confiable. Guifang Wang por un lado, y Lorents Landgren y Jeffrey Steif por otro lado, proponen que la respuesta es afirmativa, y presentan en cada clase de grafos $\mathcal{C}_{n,n+3}$ tal que $n \geq 6$ un único grafo que aquí denominamos W_n y L_n respectivamente.

En este proyecto de grado mostramos mediante evidencia computacional que la afirmación de Guifang Wang es incorrecta. Además, toda evidencia computacional se halla en armonía con la afirmación realizada por Lorents Landgren y Jeffrey Steif. Toda la implementación computacional desarrollada en este proyecto, junto con los experimentos realizados, se encuentra disponible públicamente en el repositorio de GitLab: <https://gitlab.fing.edu.uy/juan.mangado/tesis>.

El artículo actualmente publicado en Arxiv por sus autores se halla bajo referato y, en caso de ser correcto, representa un avance en el entendimiento en la existencia de grafos uniformemente más confiables. Kahl y Luttrell [9] conjeturaron que cada clase no vacía $\mathcal{C}_{n,n+3}$ de co-rango 4 tiene al menos un grafo que es Tutte-máximo. En el transcurso de este proyecto de grado se desarrollaron dos algoritmos. El primer algoritmo permite hallar el polinomio de Tutte de pseudografos. El segundo algoritmo permite encontrar, cada vez que existe, un grafo Tutte-máximo dentro de una clase no vacía $\mathcal{C}_{n,m}$ cuando $n+1 \leq m \leq 3n/2$. Se observa que su ejecución es exitosa en la clase $\mathcal{C}_{11,14}$ y permite deducir que L_{11} es Tutte-máximo en dicha clase. Para hallar dicho grafo Tutte-máximo hemos tomado todas las etiquetas posibles dentro de los 17 grafos destilación de \mathcal{M}_4 . También hemos demostrado que esta realización de todos los grafos $\mathcal{C}_{n,m}^2$ permite hallar algún grafo Tutte-máximo, siempre que exista. No obstante, para clases de co-rango 4 con una cantidad masiva de vértices existen abundantes etiquetados de grafos destilación dentro de \mathcal{M}_4 , lo que torna inviable su ejecución. Un posible trabajo futuro consiste en hallar eficientemente el etiquetado óptimo X dentro de cada destilación posible D en \mathcal{M}_c de modo que $D(Y) \preceq D(X)$ para los restantes etiquetados posibles Y .

Existen numerosos problemas abiertos que no se han abordado en este proyecto de grado. Cheng [5] demostró que todo grafo multipartito completo regular posee la máxima cantidad de árboles recubridores. Kahl y Luttrell [9] conjeturaron que todo grafo multipartito completo regular es Tutte-máximo. Dicha conjetura permanece abierta. Nuestro tutor [16] ha probado recientemente que existe una cantidad finita de clases $\mathcal{C}_{n,n+4}$ que poseen grafos uniformemente más confiables. Por lo tanto, hay una cantidad finita de grafos que son Tutte-máximos de co-rango 5, lo que genera un fuerte contraste con lo que se conoce para clases cuyo co-rango es menor que 5. La determinación de dichos grafos Tutte-máximos es un problema abierto, y la asistencia computacional es un enfoque válido para encontrar algunos (o todos) los grafos Tutte-máximos de co-rango 5.

Un célebre problema abierto postulado por Frank Boesch [3] refiere a *grafos fuertes* que enunciamos a continuación. Para cada G en $\mathcal{C}_{n,m}$ y cada entero k en $\{0, 1, \dots, m\}$ definimos $N_k(G)$ como la cantidad de subgrafos recubridores conexos de G con exactamente k aristas. Un grafo G en $\mathcal{C}_{n,m}$ es *fuerte* si para cada H en $\mathcal{C}_{n,m}$ y cada k en $\{0, 1, \dots, m\}$ se cumple que $N_k(G) \geq N_k(H)$. Es simple probar que todo grafo fuerte es uniformemente más confiable. Frank Boesch conjeturó que el recíproco es cierto, es decir, que todo grafo uniformemente más confiable es fuerte. Es interesante mencionar que todo grafo Tutte-máximo no es solamente uniformemente más confiable, sino que también es fuerte. Por lo tanto, la búsqueda de contraejemplos de la conjetura de Boesch puede realizarse buscando pares de enteros (n, m) para los cuales no existe grafo Tutte-máximo en $\mathcal{C}_{n,m}$. Puesto que nuestro segundo algoritmo permite realizar búsquedas eficientes de grafos Tutte-máximos dentro de clases $\mathcal{C}_{n,m}$ cuando $n+1 \leq m \leq 3n/2$ y para cantidades de vértices reducida, una

posible línea de trabajo futuro consiste en ejecutar nuestro algoritmo de búsqueda, listar aquellos pares (n, m) en los que no existe grafo Tutte-máximo en $\mathcal{C}_{n,m}$, y determinar en cada caso si existe grafo uniformemente más confiable en $\mathcal{C}_{n,m}$ que no sea fuerte. Es atractivo realizar dicha búsqueda para clases de grafos $\mathcal{C}_{n,m}$ de co-rango 5, puesto que allí sabemos que existe una cantidad finita de grafos uniformemente más confiables.

Bibliografía

- [1] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Computing the Tutte polynomial in vertex-exponential time. In *49th Annual IEEE Symposium on Foundations of Computer Science*, pages 677–686. IEEE, 2008.
- [2] F. Boesch, X. Li, and C. Suffel. On the existence of uniformly optimally reliable networks. *Networks*, 21(2):181–194, 1991.
- [3] F. T. Boesch. On unreliability polynomials and graph connectivity in reliable network synthesis. *J. Graph Theory*, 10(3):339–352, 1986.
- [4] J. I. Brown and D. Cox. Nonexistence of optimal graphs for all terminal reliability. *Networks*, 63(2):146–153, 2014.
- [5] C.-S. Cheng. Maximizing the total number of spanning trees in a graph: Two related problems in graph theory and optimum design theory. *Journal of Combinatorial Theory, Series B*, 31(2):240–248, 1981.
- [6] J. A. Ellis-Monaghan, I. Moffatt, C. Merino, and R. Chen, editors. *Handbook of the Tutte Polynomial and Related Topics*. CRC Press, Boca Raton, FL, 2022.
- [7] G. Haggard, D. J. Pearce, and G. Royle. Computing Tutte Polynomials. *ACM Trans. Math. Softw.*, 37(3), Sept. 2010.
- [8] N. Kahl. Extremal graphs for the Tutte polynomial. *Journal of Combinatorial Theory, Series B*, 152:121–152, 2022.
- [9] N. Kahl and K. Luttrell. On maximum graphs in Tutte polynomial posets. *Discrete Applied Mathematics*, 339:78–88, 2023.
- [10] A. Kelmans. On graphs with randomly deleted edges. *Acta Math. Hung.*, 37(1–3):77–88, 1981.
- [11] L. F. Landgren and J. E. Steif. A new framework for identifying most reliable graphs and a correction to the $K_{3,3}$ -theorem, 2024. 44 pages, 26 figures. Last revised 25 Nov 2024 (v2).
- [12] E. Moore and C. Shannon. Reliable circuits using less reliable relays. *Journal of the Franklin Institute*, 262(3):191–208, 1956.

- [13] W. Myrvold, K. H. Cheung, L. B. Page, and J. E. Perry. Uniformly-most reliable networks do not always exist. *Networks*, 21(4):417–419, 1991.
- [14] P. Romero. Wagner and Petersen are uniformly most-reliable graphs. *Pure Mathematics and Applications*, 17(2):137–152, 2019.
- [15] P. Romero. Uniformly optimally reliable graphs: A survey. *Networks*, 80(4):466–481, 2022.
- [16] P. Romero. There are finitely many uniformly most reliable graphs of corank 5. In *Algorithms and Discrete Applied Mathematics*, pages 280–291, Cham, 2025. Springer Nature Switzerland.
- [17] P. Romero and M. D. Safe. Least corank for the nonexistence of uniformly most reliable graphs. *Procedia Computer Science*, 223:88–95, 2023.
- [18] W. T. Tutte. *An algebraic theory of graphs*. Doctoral thesis, University of Cambridge, 1949.
- [19] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- [20] G. Wang. A proof of Boesch’s conjecture. *Networks*, 24(5):277–284, 1994.
- [21] C. S. Yang and L.-C. Han. The number of spanning trees of the generalized hypercube network. *Mathematical and Computer Modelling*, 17(6):97–107, 1993.

Anexo A

Anexo 1

Destilaciones correspondientes a \mathcal{M}_4

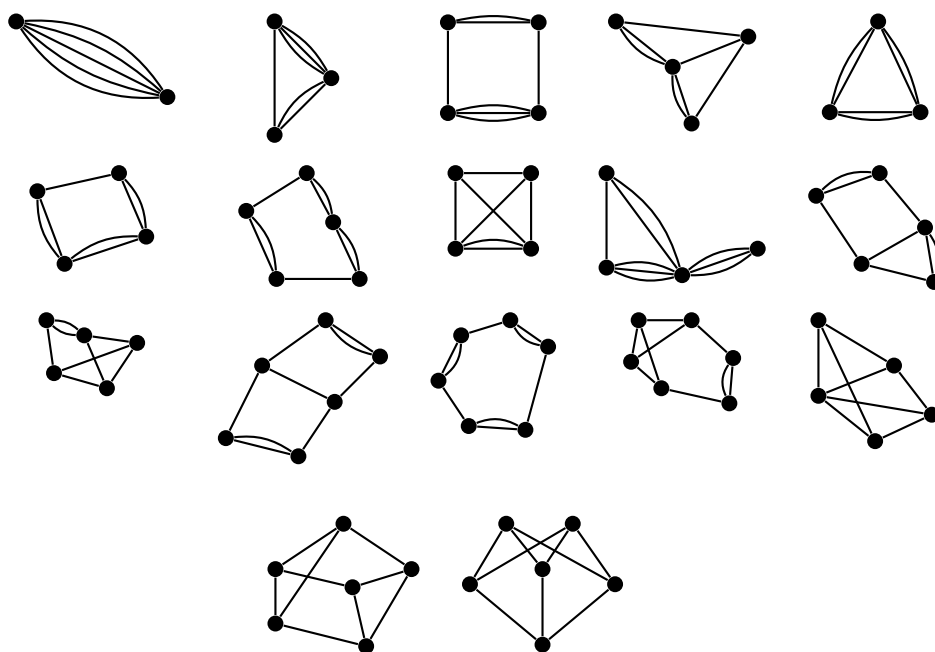


Figura A.1: Destilaciones de \mathcal{M}_4

Polinomios T_{L^s} y T_{W^s}

$$\begin{aligned}
T_{L^s}(x, y) = & y^4 + \\
& y^3(x-1)^{-1}(-4x-5+5x^{s+1}+4x^s) + \\
& y^2(x-1)^{-2}(6x^2+15x+9+4x^{3s+2}+2x^{3s+1} \\
& \quad + 6x^{2s+2}+8x^{2s+1}+4x^{2s}-15x^{s+2}-27x^{s+1}-12x^s) + \\
& y(x-1)^{-3}(-4x^3-15x^2-18x-5+8x^{5s+3}+x^{5s+1} \\
& \quad - 6x^{3s+3}-8x^{3s+2}+2x^{3s+1}-12x^{2s+3}-22x^{2s+2} \\
& \quad - 16x^{2s+1}-4x^{2s}+15x^{s+3}+42x^{s+2}+34x^{s+1}+8x^s) + \\
& (x-1)^{-4}(x^4+5x^3+9x^2+5x+x^{9s+5}-8x^{5s+4}-x^{5s+2} \\
& \quad + 2x^{3s+4}+6x^{3s+3}-2x^{3s+2}+6x^{2s+4}+14x^{2s+3} \\
& \quad + 12x^{2s+2}+4x^{2s+1}-5x^{s+4}-19x^{s+3}-22x^{s+2}-8x^{s+1})
\end{aligned}$$

$$\begin{aligned}
T_{W^s}(x, y) = & y^4 + \\
& y^3(x-1)^{-1}(-4x-5+5x^{s+1}+4x^s) + \\
& y^2(x-1)^{-2}(6x^2+15x+9+4x^{3s+2}+2x^{3s+1} \\
& \quad + 6x^{2s+2}+8x^{2s+1}+4x^{2s}-15x^{s+2}-27x^{s+1}-12x^s) + \\
& y(x-1)^{-3}(-4x^3-15x^2-18x-5+x^{5s+4}+5x^{5s+3}+3x^{5s+2} \\
& \quad - 7x^{3s+3}-5x^{3s+2}-x^{3s+1}+x^{3s}-12x^{2s+3}-22x^{2s+2} \\
& \quad - 16x^{2s+1}-4x^{2s}+15x^{s+3}+42x^{s+2}+34x^{s+1}+8x^s) + \\
& (x-1)^{-4}(x^4+5x^3+9x^2+5x+x^{9s+5}-x^{5s+5}-5x^{5s+4}-3x^{5s+3} \\
& \quad + 3x^{3s+4}+3x^{3s+3}+x^{3s+2}-x^{3s+1}+6x^{2s+4}+14x^{2s+3} \\
& \quad + 12x^{2s+2}+4x^{2s+1}-5x^{s+4}-19x^{s+3}-22x^{s+2}-8x^{s+1})
\end{aligned}$$

Glosario

En esta sección se recopilan *textualmente* las definiciones fundamentales introducidas a lo largo del documento, las cuales son utilizadas de manera recurrente en los diferentes capítulos.

Cadena

Sea G un multigrafo en $\mathcal{B}_{n,m}$ con más aristas que vértices. Una *cadena de*

G es el conjunto de aristas de un camino cuyos extremos tienen grado 3 o más y cuyos vértices internos (si los hay) tienen grado 2.

Clase $\mathcal{C}_{n,m}$

Para cada par de enteros positivos n y m tales que $n - 1 \leq m \leq \binom{n}{2}$, denotamos mediante $\mathcal{C}_{n,m}$ a la clase de todos los grafos simples y conexos con n vértices y m aristas.

Co-rango de un pseudografo

Para cada pseudografo G con $\kappa(G)$ componentes conexas, definimos el *co-rango de G* como $c(G) = |E(G)| - |V(G)| + \kappa(G)$.

Confiabilidad de un grafo

Sean n y m enteros positivos tales que $n - 1 \leq m \leq \binom{n}{2}$. Sea $G \in \mathcal{C}_{n,m}$ y sea $p \in [0, 1]$. La *confiabilidad de G evaluada en p* , que denotamos $R_G(p)$, es la probabilidad de que el subgrafo recubridor aleatorio resultante de retener a cada una de las aristas de G independientemente con probabilidad p sea conexo.

Contracción de una arista

La *contracción de la arista e en G* , que se denota $G * e$, es el grafo que se obtiene de $G - e$ tras identificar a los vértices que son extremos de e .

Destilación

La *destilación del grafo G* , que denotamos $D(G)$, es el grafo que se obtiene de G tras colapsar a cada una de sus cadenas.

Grafo 2-conexo

Decimos que el grafo G es *2-conexo* si es conexo, tiene al menos 3 vértices y no posee puntos de corte.

Grafo conexo

Decimos que un grafo G es *conexo* si para cada par de vértices u y v de G existe algún camino cuyos extremos son precisamente u y v .

Grafo simple

Un *grafo simple* G consiste en un conjunto finito de vértices, denotado por $V(G)$, y un conjunto de aristas, denotado por $E(G)$, que son pares no ordenados de elementos de $V(G)$. No se admiten lazos ni aristas múltiples.

Grafo Tutte-equivalente

Dos grafos G y H en $\mathcal{C}_{n,m}$ son *Tutte-equivalentes* si para todo par de números reales x e y se cumple que $T_G(x, y) = T_H(x, y)$.

Grafo Tutte-máximo

Un grafo G en $\mathcal{C}_{n,m}$ es *Tutte-máximo* si para todo grafo H en $\mathcal{C}_{n,m}$ se cumple que $H \preceq G$.

Grafo uniformemente más confiable (UMRG)

Un grafo G en $\mathcal{C}_{n,m}$ es *uniformemente más confiable* si para cada grafo H en $\mathcal{C}_{n,m}$ y todo $p \in [0, 1]$ se cumple que $R_G(p) \geq R_H(p)$.

Polinomio de Tutte

El *polinomio de Tutte* $T_G(x, y)$ de un pseudografo G se define recursivamente como:

$$T_G(x, y) = \begin{cases} T_{G-e}(x, y) + T_{G*e}(x, y), & \text{si } e \text{ no es un lazo ni un puente,} \\ x T_{G*e}(x, y), & \text{si } e \text{ es un puente,} \\ y T_{G-e}(x, y), & \text{si } e \text{ es un lazo,} \\ 1, & \text{si } G \text{ no tiene aristas.} \end{cases}$$

Rango de un pseudografo

Para cada pseudografo G con $\kappa(G)$ componentes conexas, definimos el *rango de G* como $r(G) = |V(G)| - \kappa(G)$.

Relación de dominancia de Tutte

Dados dos grafos G y H en $\mathcal{C}_{n,m}$, denotamos $H \preceq G$ cuando $T_G(x, y) - T_H(x, y) = (x + y - xy)P(x, y)$ para algún polinomio $P(x, y)$ con coeficientes reales no negativos.

Subgrafo

Un *subgrafo de G* es un grafo cuyo conjunto de vértices está contenido en $V(G)$ y cuyo conjunto de aristas está contenido en $E(G)$.

Sustracción de una arista

La *sustracción de la arista e en G* se denota $G - e$ y consiste en el grafo cuyo conjunto de vértices es $V(G)$ y cuyo conjunto de aristas es $E(G) - e$.