



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

ALGORITMOS ALEATORIOS PARA SISTEMAS SEPARADORES

Por: Vicente Pablo Ignacio Daza Echeverría

Tesis presentada a la Facultad de Ciencias Físicas y Matemáticas de la
Universidad de Concepción para optar al título profesional de Ingeniero Civil
Matemático

Septiembre 2024
Concepción, Chile

Dr. Nicolás Sanhueza Matamala

© 2024, Vicente Daza

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

AGRADECIMIENTOS

Quiero comenzar expresando mi sincero agradecimiento a mi profesor guía, Nicolás Sanhueza, por su paciencia y empatía a lo largo de este proceso. Asimismo, extiendo mi gratitud a todos los profesores que me orientaron durante mi formación en esta carrera.

Agradezco profundamente a mi familia por su apoyo incondicional, especialmente a mi madre, quien ha sido mi ejemplo y fuente de motivación en los momentos más difíciles.

A mi segunda familia, mis amigos de la universidad, les agradezco por los inolvidables momentos compartidos y por su compañía a lo largo de esta etapa. Iván, Carlos, Francisca, Raúl, Constanza, Ricardo, Ivette, Isidora, Catalina, Patricio y todos los amigos que conocí durante esta etapa de mi vida, les agradezco por crecer conmigo y ser parte fundamental de mis logros.

Índice general

AGRADECIMIENTOS	I
Resumen	VII
1 Introducción	1
1.1 Sistemas separadores débiles y fuertes	1
1.2 Sistemas separadores débiles y fuertes de caminos	3
1.3 Organización de este trabajo	8
2 Preliminares	10
3 Sistemas separadores	14
3.1 Sistemas separadores fuertes de caminos para el grafo $K_{\alpha N, (1-\alpha)N}$.	14
4 Algoritmo determinista de sistemas separadores para el grafo $K_{n,m}$	21
4.1 Construcción de un sistema separador para el grafo $K_{n,m}$	21
4.2 Implementación computacional	31
4.2.1 Experimentos computacionales para la cota superior $\lceil \frac{m}{n} \rceil (11n - 4)$	31
4.2.2 Tiempos de ejecución	32
5 Grafos auxiliares separadores	36
5.1 Grafo auxiliar para sistemas separadores débiles	36
5.2 Digrafo auxiliar para sistemas separadores fuertes	39
5.3 Algoritmos secuenciales	43
6 Análisis del algoritmo aleatorio secuencial	46
6.1 Análisis teórico	47
6.2 Probabilidad de borrar un arco (xy, uv) en $D_{\mathcal{F}}$	53
6.3 Cota superior para la cantidad esperada de iteraciones	75
7 Implementación computacional del algoritmo secuencial	78
7.1 Implementación computacional	78
7.2 Resultados experimentales:	78
7.2.1 Medias experimentales de la cantidad de iteraciones t del Algoritmo aleatorio secuencial	79

7.2.2	Experimentos para la esperanza de aristas no cubiertas $E(N_t)$	86
7.2.3	Experimentos para la esperanza de arcos borrados $\frac{2}{m}E(M_t)$	90
7.2.4	Tiempos de ejecución	95
8	Conclusión	98
9	Anexo	101
9.1	Algoritmo determinista	101
9.1.1	Funciones preliminares	101
9.1.2	Función Algoritmo determinista	107
9.2	Algoritmo aleatorio	108
9.2.1	Funciones preliminares:	108
9.2.2	Función Algoritmo aleatorio	111
	Referencias	114

Índice de Tablas

4.2.1	Tabla tiempos de ejecución para $N = 100$	33
4.2.2	Tabla tiempos de ejecución para $N = 250$	34
4.2.3	Tabla tiempos de ejecución para $N = 500$	34
4.2.4	Tabla tiempos de ejecución para $N = 750$	34
4.2.5	Tabla tiempos de ejecución para $N = 1000$	35
7.2.1	Tabla de valores de la media experimental de iteraciones con $N = 100$	81
7.2.2	Tabla comparativa para la media experimental de iteraciones con $N = 100$	81
7.2.3	Tabla de valores de la media experimental de iteraciones con $N = 250$	81
7.2.4	Tabla comparativa para la media experimental de iteraciones con $N = 250$	82
7.2.5	Tabla de valores de la media experimental de iteraciones con $N = 500$	82
7.2.6	Tabla comparativa para la media experimental de iteraciones con $N = 500$	82
7.2.7	Tabla de valores de la media experimental de iteraciones con $N = 750$	83
7.2.8	Tabla comparativa para la media experimental de iteraciones con $N = 750$	83
7.2.9	Tabla de valores de la media experimental de iteraciones con $N = 1000$	83
7.2.10	Tabla comparativa para la media experimental de iteraciones con $N = 1000$	84
7.2.11	Cantidad de aristas restantes por cubrir iteración t_c para $N = 1000$	90
7.2.12	Cantidad de arcos borrados iteración t_c para $N = 1000$	93
7.2.13	Cantidad de arcos borrados iteración t_b para $N = 1000$	93
7.2.14	Tabla tiempos de ejecución para $N = 100$	95
7.2.15	Tabla tiempos de ejecución para $N = 250$	95
7.2.16	Tabla tiempos de ejecución para $N = 500$	96
7.2.17	Tabla tiempos de ejecución para $N = 750$	96
7.2.18	Tabla tiempos de ejecución para $N = 1000$	96

Índice de figuras

1.2.1	Ejemplo de sistemas separadores de caminos.	5
1.2.2	Ejemplo de indicatrices de sistemas separadores de caminos.	5
2.0.1	Ejemplo de un grafo $G = (\{v_1, v_2, v_3\}, \{\{v_1v_2\}, \{v_2, v_3\}\})$	10
2.0.2	Grafo bipartito completo $K_{3,2}$	12
3.1.1	Grafos inducidos $K_{\alpha N, \alpha N}^i$ en $K_{\alpha N, (1-\alpha)N}$	15
3.1.2	Funciones $F_1(\alpha) = (1 - \alpha)$, $F_2(\alpha) = \sqrt{6\alpha(1 - \alpha) + 1}$	20
4.1.1	Ejemplo del grafo H para el grafo $K_{n,m}$	23
4.1.2	Ejemplo del grafo G' para el grafo $K_{n,m}$	24
4.1.3	Ejemplo del proceso para encontrar un sistema separador fuerte para el grafo $K_{n,m}$	24
4.1.4	Ejemplos de conjuntos F_0, F_1, F_2, F_3 para el caso de $n = 4$	25
4.1.5	Ejemplo de camino Q_k para $k = 4n + 1$	28
4.1.6	Ejemplo de camino P_k para $k = 2n + 1$	28
4.1.7	Ejemplo de camino P_k para $k = 2n + 3$	29
4.2.1	Cota superior y tamaño experimental para el sistema separador para $l = 0, 2$	32
4.2.2	Cota superior y tamaño experimental para el sistema separador para $N = 1000$	33
5.1.1	Ejemplo grafo auxiliar $G_{\{X\}}$	37
5.2.1	Ejemplo digrafo auxiliar $D_{\{X\}}$	41
6.2.1	Intersección de los eventos $\mathcal{E}^{k-1}(xy, uv)$ y $\mathcal{E}^k(xy, uv)$	54
6.2.2	Ejemplos de los 4 subcasos para el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$	55
6.2.3	Par de permutaciones $(\sigma, \tau) \in S^l(xy, uv)$ antes y después de aplicar T^l	57
6.2.4	Par de permutaciones (σ, τ) antes y después de aplicar W	58
6.2.5	Las aristas no comparten vértices y además i, j son consecutivos.	63
6.2.6	Las aristas no comparten vértices y además i, j no son consecutivos.	64
6.2.7	Las aristas comparten un vértice $y = v$ y además i, j son consecutivos.	66
6.2.8	Las aristas comparten un vértice $y = v$ y además i, j no son consecutivos.	67
6.2.9	Las aristas comparten el vértice $x = u$	69

7.2.1	Media experimental y esperanza teórica de iteraciones del algoritmo para $N = 1000$	80
7.2.2	Media y esperanza teórica de iteraciones para $\alpha = 0,45$	85
7.2.3	Gráfica de la media de selecciones necesarias para el problema del coleccionista de cupones y la cota superior $m \ln(mn)$	86
7.2.4	Media y esperanza teórica de aristas restantes por cubrir para $N = 1000$ y $\alpha = 0,45$	88
7.2.5	Error de aristas restantes por cubrir para $N = 1000$ y $\alpha = 0,45$	88
7.2.6	Media y esperanza teórica de arcos borrados $N = 1000$ y $\alpha = 0,45$	91
7.2.7	Error de arcos borrados para $N = 1000$ y $\alpha = 0,45$	92
7.2.8	Error proporcional: Media y esperanza de arcos borrados para $N = 1000$ y $\alpha = 0,45$	94

Resumen

Dado un grafo G , una colección \mathcal{P} de caminos de G es un *sistema separador fuerte de caminos* si para cada par de aristas distintas e y f hay un camino en \mathcal{P} que contiene e pero no f . El proyecto enmarca el estudio de manera teórica y algorítmica de los sistemas separadores fuertes de caminos para el grafo bipartito completo $K_{n,m}$. Por el lado teórico, encontramos cotas para el tamaño mínimo de un sistema separador fuerte de caminos de $K_{n,m}$, y además proporcionamos grafos y dígrafos auxiliares que permiten el estudio de los sistemas separadores de manera general. A partir de estos grafos y dígrafos diseñamos algoritmos para encontrar sistemas de caminos separadores fuertes, realizando su análisis teórico para el grafo $K_{n,m}$. Finalmente logramos implementar computacionalmente estos algoritmos, corroborando los resultados teóricos obtenidos por medio de un análisis experimental.

Capítulo 1

Introducción

1.1. Sistemas separadores débiles y fuertes

El concepto de separación es clave en el campo de la combinatoria extremal. En general, la combinatoria extremal se centra en el estudio de una colección de objetos que satisfacen ciertas propiedades, y en particular encontrar el tamaño de dicha colección. Dentro de esta área de estudio se encuentran los sistemas separadores débiles de conjuntos. Dado un conjunto S diremos que una familia \mathcal{F} de subconjuntos de S es un *sistema separador débil* de S si para cada par de elementos distintos $a, b \in S$, existe un conjunto $X \in \mathcal{F}$ que contiene exactamente uno de a, b .

Rényi [Rén61] inicio el estudio de los sistemas separadores débiles en 1961 y tuvo como principal objetivo encontrar el tamaño mínimo de un sistema separador débil de un conjunto de n elementos, denotaremos a este tamaño por $ws(n)$.

Rényi logró demostrar que $ws(n) = \lceil \log_2(n) \rceil$. Para demostrar esta igualdad podemos proceder de la siguiente manera. Sea S un conjunto de n elementos y $\mathcal{F} = \{X_1, \dots, X_k\}$ un sistema separador débil de S , sea además $\mathcal{F}(i) \in \{0, 1\}^k$ el vector indicatriz para cada elemento i de S con respecto a \mathcal{F} . Dado que \mathcal{F} es un sistema separador débil de S , es necesario que $\mathcal{F}(i)$ sea distinto para cada $i \in S$, de esta forma debemos tener al menos tantos vectores indicatrices distintos como elementos en S . Dado que la cantidad de vectores indicatrices distintos que podemos formar con una familia de tamaño k es 2^k se deduce que $2^k \geq n$, y a partir de esto, se tiene que $k \geq \lceil \log_2(n) \rceil$, y por lo tanto $ws(n) \geq \lceil \log_2(n) \rceil$. Por otro

lado, si consideramos una familia \mathcal{F} de subconjuntos de S de tamaño $\lceil \log_2(n) \rceil$ tal que $\mathcal{F}(i)$ sea distinto para cada $i \in S$ obtenemos un sistema separador débil de S . Esta familia existe dado que basta seleccionar n vectores distintos en $\{0, 1\}^{\lceil \log_2(n) \rceil}$, y de esta manera, la familia que posee estos vectores como vectores indicatrices es la familia deseada, demostrando así que $\text{ws}(n) \leq \lceil \log_2(n) \rceil$.

Encontrar el tamaño mínimo de un sistema separador débil se vuelve una pregunta mucho más interesante y desafiante cuando se imponen restricciones sobre los elementos de \mathcal{F} . Por ejemplo, encontrar un sistema separador débil \mathcal{F} donde los elementos de \mathcal{F} tengan cardinalidad a lo más k fue estudiado originalmente por Katona [Kat66]. Definimos $\text{ws}(n, k)$ como el tamaño mínimo de un sistema separador débil \mathcal{F} de un conjunto S , donde $|S| = n$ y además para cada $X \in \mathcal{F}$, $|X| \leq k$. Katona demostró que

$$\frac{\log(n)}{\log(en/k)} \frac{n}{k} \leq \text{ws}(n, k) \leq \frac{\log(2n)}{\log(n/k)} \frac{n}{k}.$$

Años más tarde Wegener [Weg79] proporcionaría una demostración más corta y sencilla para la cota inferior haciendo uso de un argumento de entropía, y además una mejor cota superior.

A la par del estudio de los sistemas separadores débiles también se han estudiado nociones más fuertes de separación. Dado un conjunto S , decimos que una familia \mathcal{F} de subconjuntos de S es un *sistema separador fuerte* de S si para cada par de elementos distintos $a, b \in S$, existen los conjuntos X_a y X_b en \mathcal{F} , tal que $a \in X_a$, $b \notin X_a$ y $b \in X_b$, $a \notin X_b$.

Similar al caso débil, buscamos encontrar el tamaño mínimo de un sistema separador fuerte de un conjunto de n elementos, que denotaremos por $\text{ss}(n)$. Dado que todo sistema separador fuerte es un sistema separador débil, es claro que $\text{ws}(n) \leq \text{ss}(n)$.

El estudio de $\text{ss}(n)$ comenzó por Dickson [Dic69], logrando demostrar que

$$\lim_{n \rightarrow \infty} \frac{\text{ss}(n)}{\log_2(n)} = 1.$$

Luego, un par de años después Spencer [Spe70] proporcionaría un resultado importante en relación al valor de $\text{ss}(n)$. Spencer encontraría el valor exacto de

$ss(n)$, siendo $ss(n)$ igual a $t = \log_2(n) + \log_2 \log_2 n + \frac{1}{2} \log_2(\frac{\pi}{2}) + o(1)$, donde t corresponde al mínimo entero positivo que satisface la desigualdad

$$\binom{t}{\lfloor t/2 \rfloor} \geq n.$$

Dado que la demostración dada por Spencer es sencilla y elegante, procederemos a demostrarlo.

Demostración. Sea $\mathcal{F} = \{X_1, \dots, X_t\}$ un sistema separador fuerte de S . Sea A una matriz de dimensión $n \times t$ con entradas $\{0,1\}$ tal que $A_{ij} = 1$, si y solo si, $i \in X_j$. Notemos que \mathcal{F} es un sistema separador fuerte de S , si y solo si, para cada par $i, j \in \{1, \dots, n\}$, existe un $k \in \{1, \dots, t\}$, tal que $A_{ik} = 1$ y $A_{jk} = 0$. Sea $U_i = \{j : A_{ij} = 1\} \subseteq \{1, \dots, t\}$ para cada $i \in \{1, \dots, n\}$. Notemos que las condiciones sobre la matriz A indican que para cada par $i \neq i' \in \{1, \dots, n\}$, se debe cumplir que $U_{i'} \not\subseteq U_i$. De esta forma, la familia $\{U_1, \dots, U_n\}$ es una anticadena y por el lema de Sperner [Spe28] esta familia existe, si y solo si,

$$\binom{t}{\lfloor t/2 \rfloor} \geq n,$$

demostrando así lo deseado. □

La comunidad siguió interesándose en los sistemas separadores débiles y fuertes, Ramsay y Roberts [RR96] y Kündgen, Mubayi y Tetali [KMT01] estudiaron los variantes de las cantidades $ws(n)$ y $ss(n)$ apoyándose en el trabajo de Spencer, pero sobre sistemas separadores donde sus elementos poseían a lo más k elementos o exactamente k . Hansel [Han64], y Bollobás y Scott [BS07], entre otros investigadores ajustarían aun más las cotas antes mencionadas tanto para el caso débil como el fuerte.

1.2. Sistemas separadores débiles y fuertes de caminos

Dado un grafo $G = (V, E)$, deseamos encontrar sistemas separadores \mathcal{P} débiles o fuertes de $E(G)$, donde además cada elemento de \mathcal{P} sea un camino en el grafo G . En general estos tipos de sistemas separadores \mathcal{P} son llamados *sistema separador*

débil de caminos y *sistema separador fuerte de caminos* respectivamente. En lo que sigue, el tamaño mínimo de un sistema separador débil de caminos para un grafo G lo denotaremos por $wsp(G)$, análogamente se denota $ssp(G)$ para el caso fuerte.

Es de principal interés entender en completitud estos sistemas separadores de caminos puesto que ellos surgen naturalmente en el contexto del diseño de redes. En general podemos plantearnos la siguiente situación. Se nos presenta una red de comunicación con un (y como máximo uno) enlace defectuoso y nuestro objetivo es identificar este enlace. Para encontrarlo se podrían probar todos los enlaces, pero esto no es muy eficiente; ¿Podemos hacerlo mejor? Una prueba natural a realizar sería enviar un mensaje entre un par de nodos a lo largo de un camino predeterminado, si el mensaje no llega a su destino previsto, podemos concluir que el enlace defectuoso se encuentra en este camino. Encontrar este conjunto de caminos para realizar las correspondientes pruebas es un problema que ha sido considerado en la literatura teórica en el área de la informática. Estos conjuntos de caminos poseen múltiples nombres en la literatura informática, incluyendo “identifying families”, “test covers”, “test families”, y “test sets” (Para más detalles recomendamos leer el artículo [De +03]). Si modelamos la red de comunicación que debemos analizar como un grafo G , sabemos que un conjunto fijo de tales pruebas logra localizar cualquier enlace defectuoso, si y sólo si, la familia de caminos correspondiente es un sistema separador de caminos de G . Dado que buscamos eficiencia, la pregunta natural que surge es: ¿Cuál es el tamaño mínimo de un sistema separador de caminos para un grafo G ? Es decir, encontrar $wsp(G)$ o $ssp(G)$, dependiendo del contexto.

Consideraremos un ejemplo, observe la Figura 1.2.1, en ella se encuentra un árbol T de 4 vértices. En el lado izquierdo de la figura con color azul se encuentra representado un sistema separador débil de caminos del grafo T . Este sistema no es un sistema separador fuerte, esto debido a que no existe un camino que posea la arista v_1v_2 y no la arista v_1v_3 , además de esto, la arista v_1v_4 no pertenece a ningún camino del sistema. En cambio, a su derecha se encuentra representado con líneas rojas un sistema separador fuerte de caminos de T .

Si representamos los caminos por medio de una tabla (vea la Figura 1.2) donde se muestra un 1 si la arista pertenece al camino y un 0 si es que la arista no pertenece al camino, entonces podemos de manera sencilla identificar una posible arista

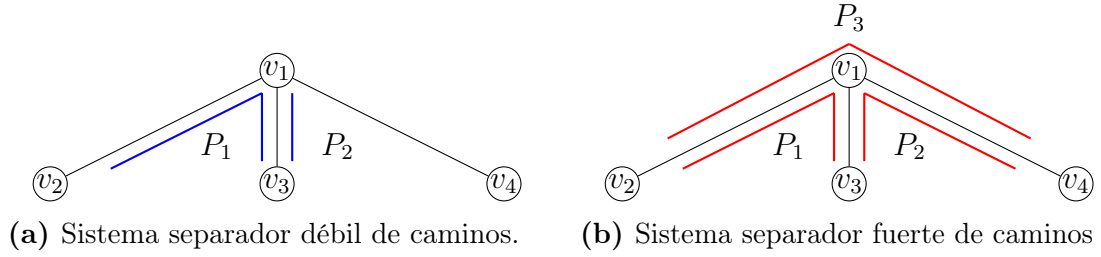


Figura 1.2.1: Ejemplo de sistemas separadores de caminos.

defectuosa por medio del envío de algunos mensajes a través de estos caminos. Para ello, supongamos que existe una falla en la arista v_1v_4 . Luego, procederemos a enviar mensajes a través de los caminos de nuestro sistema separador débil o fuerte, si un mensaje es enviado exitosamente a través del camino P_1 y otro mensaje no llega a su destino a través del camino P_2 , podemos concluir exitosamente que la falla se encuentra en la arista v_1v_4 cumpliendo así nuestro objetivo.

Aristas \ Caminos	P_1	P_2
v_1v_2	1	0
v_1v_3	1	1
v_1v_4	0	0

(a) Sistema separador débil de caminos.

Aristas \ Caminos	P_1	P_2	P_3
v_1v_2	1	0	1
v_1v_3	1	1	0
v_1v_4	0	1	1

(b) Sistema separador fuerte de caminos.

Figura 1.2.2: Ejemplo de indicatrices de sistemas separadores de caminos.

El uso de los sistemas separadores para realizar esta tarea fue sugerido por primera vez por Zakrevski y Karpovsky [ZK98]. Posteriormente, Honkala, Karpovsky y Litsyn [HKL03] y de forma independiente Rosendahl [Ros03] también realizaron estudios de este problema. Rosendahl considero este problema tanto en vértices como en aristas usando ciclos, investigando este problema en familias específicas de grafos, como hipercubos, grafos bipartitos completos y grillas.

Foucaud y Kovše [Mat13] consideraron el problema de encontrar sistemas separadores de caminos para cubrir los vértices de un grafo. Foucaud y Kovše estudiaron estos sistemas desde un punto de vista algorítmico y combinatorio. Por el lado algorítmico, demostraron que si las familias de caminos a considerar también tienen un tamaño acotado, entonces el problema de optimización asociado es APX-completo. Algunos otros ejemplos de estudios sobre este tipo de sistemas separadores en el área de la informática son [ARK09; Har+07; TRH12].

El estudio de los sistemas separadores de caminos está relacionado con la descomposición por caminos de un grafo. La famosa conjetura de Gallai [Bon14] afirma que todo grafo conexo en n vértices puede descomponerse usando a lo más $\lfloor \frac{n}{2} \rfloor$ caminos; aquí por descomposición queremos decir que cada arista del grafo está cubierta exactamente una vez. Lovász [Lov68] logró demostrar una versión ligeramente más débil de la conjetura de Gallai. Lovász demostró que cada grafo de n vértices se puede descomponer usando a lo más $n/2$ caminos y ciclos. Dado que cada ciclo podemos separarlo en 2 caminos, cada grafo de n vértices se puede descomponer usando a lo más n caminos. Este resultado es usado múltiples veces en el estudio de los sistemas separadores para grafos.

El problema atraería la atención de la comunidad dedicada a la combinatoria debido a que Gyula Katona en agosto de 2013 en el 5th Emléktábla Workshop en Budapest, plantearía la siguiente pregunta. ¿Cuál es el máximo tamaño mínimo de un sistema separador de caminos sobre todos los grafos de n vértices? Denotaremos estas cantidades por $wsp(n)$ y $ssp(n)$ para el caso débil y fuerte respectivamente. Falgas-Ravry, Kittipassorn, Korándi, Letzter, y Narayanan [Fal+14] estudiaron la versión débil de la pregunta, encontrando los valores de $wsp(G)$ para diversos grafos. Dado que un sistema separador débil de caminos de un grafo G es por definición un sistema separador débil de $E(G)$, se sigue que $wsp(n) \geq \lceil \log_2(|E(G)|) \rceil$. En particular, para cualquier grafo G de n vértices conexo se tiene que $wsp(n) = \Omega(\log n)$.

Falgas-Ravry, Kittipassorn, Korándi, Letzter, y Narayanan lograron acotar $wsp(n)$ inferiormente como sigue; Considere el grafo completo con n vértices K_n , supongamos que tenemos un sistema separador débil de caminos \mathcal{P} de K_n , con k caminos. Notemos que como máximo una arista de K_n queda descubierta por los caminos de \mathcal{P} , y además, a lo más k aristas de K_n pertenecen exactamente a un camino de \mathcal{P} . Dado que cualquier camino de K_n tiene longitud como máximo $n - 1$, se concluye que

$$k(n - 1) \geq 1 + k + 2 \left(\binom{n}{2} - k - 1 \right),$$

o equivalentemente, $k \geq n - 1 - 1/n$. En base a esto, se concluye que $wsp(n) \geq n - 1$. Debido a este resultado, ellos conjeturaron que $wsp(n) = O(n)$, es decir, existe una constante C tal que para todo grafo G de n vértices $wsp(n) \leq Cn$. Ellos lograron

demostrar esta conjetura para grafos en n vértices con grado mínimo acotado y n suficientemente grande, grafos aleatorios de Erdős-Rényi $G(n, p)$, entre otros. Dentro de sus resultados, ellos demostraron que para arboles T de n vértices, con $n \geq 4$, se cumple que

$$\left\lceil \frac{n+1}{3} \right\rceil \leq \text{wsp}(T) \leq \left\lfloor \frac{2(n-1)}{3} \right\rfloor,$$

donde además para ciertos arboles estas cotas se alcanzan con igualdad.

La pregunta plateada por Gyula Katona también fue abordada simultáneamente para los sistemas separadores fuertes de caminos por Balogh, Csaba, Martin, and Pluhár [Bal+16]. Ellos lograron plantear la misma conjetura del grupo Falgas-Ravry, Kittipassorn, Korándi, Letzter, y Narayanan pero en su versión fuerte, es decir $\text{ssp}(n) = O(n)$. Dentro de sus resultados calcularon el valor $\text{ssp}(G)$ de manera exacta para los bosques en términos de sus hojas y vértices de grado 2. Un corolario de este resultado nos dice que G es un árbol de n vértices con $\text{ssp}(G) = n - 1$, si y solo si, G es una subdivisión de una estrella (Note que el sistema separador fuerte de caminos de la Figura 1.2.1 respeta esto). Ellos además demostraron que la conjetura en su versión fuerte, también es cierta para el grafo completo K_n , hipercubos d -dimensionales y para el grafo aleatorio de Erdős-Rényi $G(n, p)$.

Letzter [Let24] aportaría un importante avance en la demostración de la conjetura demostrando que todo grafo G de n vértices satisface que $\text{ssp}(G) = O(n \log^* n)$, donde \log^* corresponde al logaritmo iterado, siendo este igual al número de veces que la función logaritmo debe aplicarse iterativamente antes de que el resultado sea menor o igual a 1. La conjetura fue finalmente demostrada recientemente por Bonamy, Botler, Dross, Naia, y Skokan [Bon+23], siendo ellos capaces de demostrar que $\text{ssp}(G) \leq 19n$ para cualquier grafo G de n vértices.

Dado que las cotas conocidas de $\text{ssp}(n)$ y $\text{wsp}(n)$ sobre diversos grafos difieren bastante del valor $19n$ encontrado por Bonamy, Botler, Dross, Naia, y Skokan y aún se desconocen cotas para muchos otros grafos, sigue siendo de interés en la actualidad estimar o encontrar los valores $\text{ssp}(n)$ y $\text{wsp}(n)$. Recientemente, Arrepol, Asenjo, Astete, Cartes, Gajardo, Henríquez, Opazo, Sanhueza-Matamala, y Thraves Caro [Arr+23] encontraron el valor exacto de $\text{wsp}(T)$ para los grafos arboles T . Además de esto, también lograron acotar para los árboles su respectiva

variante para la separación de vértices. Por otro lado, recientemente B.Wickes [Wic24b] se dedicó al estudio de $\text{wsp}(K_n)$ para el grafo completo K_n en profundidad, demostrando que $\text{wsp}(K_n) \leq n$ siempre que n o $n - 1$ es un número primo, y de manera general, B.Wickes logró demostrar que para cualquier grafo completo K_n se tiene $\text{wsp}(K_n) \leq (21/16 + o(1))n$.

Fernandes, Mota, y Sanhueza [CS23] también estudiaron el grafo K_n y lograron demostrar que $\text{ssp}(K_n) = (1 + o(1))n$. Este grupo además estudió los sistemas separadores de caminos para ciertas clases de grafos αn -regulares de n vértices y (δ, L) -robustamente conectados, donde un grafo G es (δ, L) -robustamente conectados si para todo $x, y \in V(G)$, existe $1 \leq \ell \leq L$ tal que hay al menos δn^ℓ caminos de x a y con exactamente ℓ vértices internos cada uno.

El grupo de Fernandes, Mota, y Sanhueza para los grafos G de n vértices αn -regulares y (δ, L) -robustamente conectados encontraron sistemas separadores de caminos fuertes de tamaño $(\sqrt{3\alpha + 1} - 1 + o(1))n$. Dentro de los grafos αn -regulares y (δ, L) -robustamente conectados se encuentra el grafo bipartito completo en n vértices $K_{n/2, n/2}$. El grafo bipartito completo $K_{n/2, n/2}$ es αn -regular con $\alpha = 1/2$, esto es debido a que cada vértice de un lado de la partición es adyacente a todos los vértices del lado contrario de la partición, de esta forma todos los vértices poseen exactamente $n/2$ vecinos. Por otro lado, para cada par de vértices x, y de distintos lados de la partición existen $((n/2) - 1)^2 \geq n/5$ caminos que los conectan con 2 vértices interiores, siendo así el grafo $K_{n/2, n/2}$ $(1/5, 2)$ -robustamente conectado. Considerando estos valores Fernandes, Mota, y Sanhueza estimaron que $\text{ssp}(K_{n/2, n/2}) = (\sqrt{5/2} - 1 + o(1))n$. Dentro de las preguntas que se plantean a futuro por el grupo de Fernandes, Mota, y Sanhueza, ellos se cuestionan si es cierto que $\text{ssp}(G) \leq (1 + o(1))n$ para todo grafo de n vértices, siendo así de interés analizar grafos no regulares, como por ejemplo el grafo bipartito completo $K_{n, m}$, con $n \neq m$.

1.3. Organización de este trabajo

En el Capítulo 2 presentamos definiciones generales de grafos y la notación usada en este trabajo. En el Capítulo 3 proporcionamos cotas inferiores y superiores para $\text{ssp}(K_{n, m})$, cuando $n \neq m$. En el capítulo 4 planteamos y analizaremos las estrategias del grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23]

para encontrar sistemas separadores fuertes enfocándonos en el caso del grafo $K_{n,m}$, para posteriormente diseñar un algoritmo determinista que construye sistemas separadores fuertes para el grafo $K_{n,m}$. En el Capítulo 5 definiremos grafos y digrafos auxiliares, que permiten diseñar algoritmos de búsqueda de sistemas separadores $K_{n,m}$. El Capítulo 6 corresponde análisis teórico de un algoritmo aleatorio, donde estudiamos de manera probabilística la cantidad de pasos esperados del algoritmo para encontrar sistemas separadores, entre otras cantidades de interés. En el Capítulo 7 implementamos computacionalmente nuestro algoritmo aleatorio y presentamos resultados experimentales que respaldan nuestros resultados teóricos.

Capítulo 2

Preliminares

Dado un conjunto de elementos V , denotamos $[V]^k$ al conjunto de subconjuntos de V de tamaño k .

Un *grafo* es un par $G = (V, E)$ de conjuntos tales que $E \subseteq [V]^2$; así, los elementos de E son subconjuntos de V de 2 elementos. Los elementos de V son los *vértices* del grafo G y los elementos de E son sus *aristas*. La forma habitual de representar un grafo es dibujando un punto para cada vértice y uniendo dos de estos puntos por una línea si sus vértices correspondientes forman una arista.

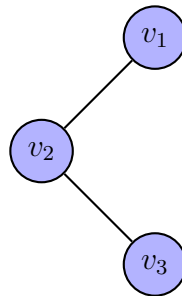


Figura 2.0.1: Ejemplo de un grafo $G = (\{v_1, v_2, v_3\}, \{\{v_1v_2\}, \{v_2, v_3\}\})$

Un grafo con un conjunto de vértices V se dice que es un grafo *en* V . El conjunto de vértices de G se denota por $V(G)$ y el conjunto de aristas se denota por $E(G)$. El número de vértices de un grafo G es su *orden*, escrito como $|V(G)|$, su número de aristas se denota por $|E(G)|$. Para el grafo vacío (\emptyset, \emptyset) simplemente escribimos \emptyset . Un grafo de orden 0 o 1 se denomina *trivial*.

Un vértice v es *incidente* con una arista e si $v \in e$. Los dos vértices incidentes de una arista son sus *vértices finales* o *extremos*. Una arista $\{x, y\}$ generalmente se

escribe como xy (o yx). Dos vértices x, y de G son *adyacentes*, o *vecinos*, si xy es una arista de G . Dos aristas $e \neq f$ son *adyacentes* si tienen un vértice en común. Si todo par de vértices distintos de G son adyacentes, entonces G es *completo*. Denotaremos por K_n al grafo completo de n vértices.

Sea $G = (V, E)$ un grafo (no vacío). El conjunto de vecinos de un vértice v en G se denota por $N_G(v)$, o brevemente por $N(v)$ si el contexto esta claro. De forma más general, para un conjunto $U \subseteq V$, definimos $N_G(U) := \left(\bigcup_{v \in U} N_G(v) \right) \setminus U$. Este conjunto de vértices corresponde a los vecinos en el conjunto $V \setminus U$ de los vértices en U . Nos refiriéremos a estos vértices como *vecinos de U* y si el contexto esta claro se denotaran por $N(U)$. Además, para cada vértice $v \in G$ y $U \subseteq G$ definimos $N_G(v, U) := N_G(v) \cap U$. Este conjunto corresponde a los vecinos del v en el grafo G que además pertenecen al conjunto de vértices U .

El *grado* $d_G(v)$ de un vértice v es igual al número de vecinos de v , nuevamente si el contexto esta claro denotaremos al grado de v por $d(v)$. Diremos que un vértice es *aislado* si este posee grado 0. El numero $\delta(G) := \text{mín}\{d(v) : v \in V\}$ es el *grado mínimo de G* y el numero $\Delta(G) := \text{máx}\{d(v) : v \in V\}$ es el *grado máximo de G* . Si todos los vértices de G tienen el mismo grado k , entonces G es *k -regular*.

Dado dos grafos $G = (V, E)$ y $G' = (V', E')$ definimos $G \cup G' := (V \cup V', E \cup E')$ y $G \cap G' := (V \cap V', E \cap E')$. Si $G \cap G' = \emptyset$, entonces G y G' son *disjuntos*. Si $V \subseteq V'$ y $E \subseteq E'$, entonces G es *subgrafo* de G' (y G' es *supergrafo* de G), escrito como $G \subseteq G'$. Más formalmente, diremos que G' *contiene* a G .

Un *camino* es un grafo no vacío $P = (V, E)$ de la forma

$$V = \{x_0, x_1, \dots, x_k\}, \quad E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\},$$

donde los vértices x_i son todos distintos. Los vértices x_0 y x_k están *unidos* por P y se llaman sus *vértices extremos* o *extremos* de P ; los vértices x_1, \dots, x_{k-1} son los vértices *internos* de P . El número de aristas de un camino es su *largo*, y el camino de largo k se denota por P_k . Tenga en cuenta que se permite que k sea cero; así, $P_0 = K_1$. A menudo nos referimos a un camino por la secuencia natural de sus vértices escribiendo $P = x_0x_1 \cdots x_k$ y llamando a P un camino *de x_0 a x_k* .

Si $P = x_0 \cdots x_{k-1}$ es un camino y $k \geq 3$, entonces el grafo C , con $V(C) = V(P)$

y con $E(C) := E(P) \cup \{x_{k-1}x_0\}$ se llama *ciclo*. Al igual que con los caminos, a menudo denotamos un ciclo por su secuencia (cíclica) de vértices; el ciclo C anterior podría escribirse como $x_0 \cdots x_{k-1}x_0$. El *largo* de un ciclo es su número de aristas (o vértices); el ciclo de largo k se denota por C_k .

Sea $r \geq 2$ un número entero. Un grafo $G = (V, E)$ se llama *r-partito* si V admite una partición en r partes de modo que cada arista tenga sus extremos en diferentes partes. En lugar de ‘2-partito’, se suele decir *bipartito*. Un grafo r -partito se llama *completo* si todo par de vértices de diferentes partes de la partición son adyacentes. El grafo bipartito completo con partición $V = A \cup B$, donde $|A| = n$ y $|B| = m$ se denota por $K_{n,m}$.

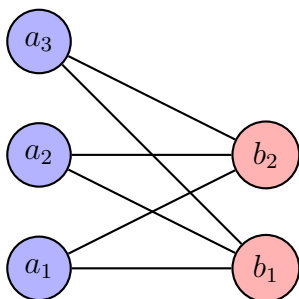


Figura 2.0.2: Grafo bipartito completo $K_{3,2}$.

Un *grafo dirigido* (o *dígrafo*) es un par $D = (V, E)$ de conjuntos disjuntos (de *vértices* y *arcos*), donde $E \subseteq V \times V$. Diremos que un arco $(a, b) \in E$ tiene vértice *inicial* a y vértice *final* b .

El conjunto de vértices de D se denota por $V(D)$ y el conjunto de arcos se denota por $E(D)$. El número de vértices de un dígrafo D es su *orden*, escrito como $|V(D)|$ y su número de arcos se denota por $|E(D)|$. Para cada vértice $v \in V(D)$ diremos que el conjunto $N_D^-(v) := \{u : (u, v) \in E(D)\}$ es la *vecindad de entrada de v* y análogamente, el conjunto $N_D^+(v) := \{u : (v, u) \in E(D)\}$ la *vecindad de salida de v* . Un *dígrafo completo*, es un dígrafo $D = (V, E)$ tal que para todo par de vértices distintos $x, y \in V$, existe el arco $(x, y) \in E$.

A continuación, presentamos las definiciones necesarias para entender lo realizado en este trabajo.

Definición 2.0.1. Dado un conjunto S , diremos que una familia \mathcal{F} de subconjuntos de S *separa fuertemente* un par de elementos distintos $a, b \in S$ si existen los conjuntos X_a y X_b en \mathcal{F} , tal que $a \in X_a$, $b \notin X_a$ y $b \in X_b$, $a \notin X_b$. Si

\mathcal{F} separa fuertemente a todos los pares de elementos distintos de S , diremos que \mathcal{F} es un *sistema separador fuerte* de S .

Definición 2.0.2. Dado un conjunto S , diremos que una familia \mathcal{F} de subconjuntos de S *separa débilmente* un par de elementos distintos $a, b \in S$ si existe un conjunto $X \in \mathcal{F}$, tal que X contiene exactamente a uno de los dos. Si \mathcal{F} separa débilmente a todos los pares de elementos distintos de S , diremos que \mathcal{F} es un *sistema separador débil* de S .

Nuestro principal objeto de estudio son los sistemas separadores fuertes de caminos.

Definición 2.0.3. Dado un grafo $G = (V, E)$, decimos que una familia de caminos \mathcal{P} de G es un *sistema separador débil de caminos de G* si $\{E(P) : P \in \mathcal{P}\}$ es un sistema separador débil de $E(G)$. Análogamente, \mathcal{P} es un *sistema separador fuerte de caminos de G* si $\{E(P) : P \in \mathcal{P}\}$ es un sistema separador fuerte de $E(G)$.

Capítulo 3

Sistemas separadores

3.1. Sistemas separadores fuertes de caminos para el grafo $K_{\alpha N, (1-\alpha)N}$

En esta sección denotaremos al grafo bipartito $K_{n,m}$ en N vértices por $K_{\alpha N, (1-\alpha)N}$, donde $0 < \alpha < \frac{1}{2}$, $n = \alpha N$ y $m = (1 - \alpha)N$. Para poder estudiar cotas del valor $\text{ssp}(K_{\alpha N, (1-\alpha)N})$, nos apoyaremos en el estudio realizado por el grupo de C. G. Fernandes, G. O. Mota, y N. Sanhuesa-Matamala [CS23]. El grupo de C. G. Fernandes, G. O. Mota, y N. Sanhuesa-Matamala introducen los grafos (δ, L) -robustamente conectados.

Definición 3.1.1. Sea G un grafo, entonces G es (δ, L) -robustamente conectado si para todo $x, y \in V(G)$, existe $1 \leq \ell \leq L$ tal que hay al menos δn^ℓ caminos de x a y con exactamente ℓ vértices internos cada uno.

Dada esta definición, el grupo de C. G. Fernandes, G. O. Mota, y N. Sanhuesa-Matamala obtiene el siguiente teorema

Teorema 3.1.2. *Sea $\alpha, \delta \in (0, 1)$ y $L \geq 1$, considere un grafo G de N vértices que es αN -regular y (δ, L) -robustamente conectado. Entonces*

$$\text{ssp}(G) = (\sqrt{3\alpha + 1} - 1 + o(1))N.$$

A partir de este teorema, ellos obtienen el siguiente corolario para el grafo bipartito

completo de N vértices $K_{N/2, N/2}$.

Corolario 3.1.3. Para cada $\varepsilon > 0$ y N suficientemente grande, se cumple que $\text{ssp}(K_{N/2, N/2}) \leq (\sqrt{5/2} - 1 + \varepsilon)N$.

El Corolario 3.1.3 nos permite encontrar un cota superior de $\text{ssp}(K_{\alpha N, (1-\alpha)N})$ para nuestros casos de interés con $\alpha \in (0, 1/2)$.

Proposición 3.1.4. Sea $\alpha \in (0, 1/2)$, $N \in \mathbb{N}$ suficientemente grande y $K_{\alpha N, (1-\alpha)N}$ el grafo bipartito en N vértices. Entonces para todo $\varepsilon > 0$ se tiene que

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \leq \left\lceil \frac{1-\alpha}{\alpha} \right\rceil \left((\sqrt{5/2} - 1 + \varepsilon)2\alpha N \right).$$

Demostración. Sea $l = \lceil (1-\alpha)/\alpha \rceil$ y sea A, B los conjuntos que conforman la partición del grafo bipartito $K_{\alpha N, (1-\alpha)N}$, con $B = \{1, \dots, (1-\alpha)N\}$. Dado que $\alpha \in (0, 1/2)$ se tiene que $l > 1$ y podemos seleccionar l subconjuntos B_i de B de tamaño αN que forman una partición de B , donde $l-1$ de ellos sean disjuntos. Cada uno de estos conjuntos de vértices B_i unidos los vértices de A permiten inducir grafos bipartitos completos $K_{\alpha N, \alpha N}^i$ en el grafo $K_{\alpha N, (1-\alpha)N}$, para más detalles observe la Figura 3.1.1 donde se muestran estos subgrafos $K_{\alpha N, \alpha N}^i$ con color rojo.

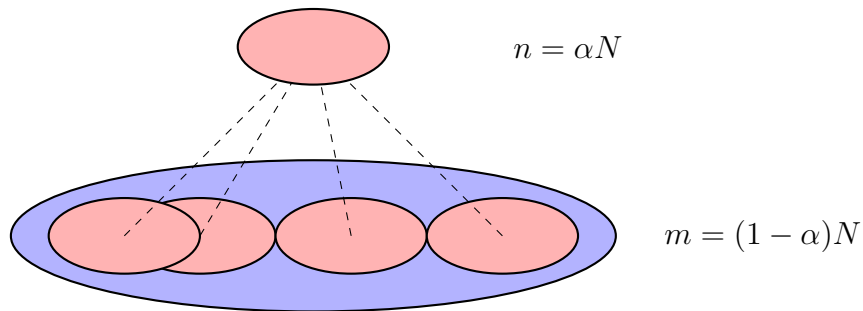


Figura 3.1.1: Grafos inducidos $K_{\alpha N, \alpha N}^i$ en $K_{\alpha N, (1-\alpha)N}$.

Para cada uno de los grafos inducidos $K_{\alpha N, \alpha N}^i$ podemos aplicar el Corolario 3.1.3 de forma que

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \leq l \left((\sqrt{5/2} - 1 + \varepsilon)2\alpha N \right) = \left\lceil \frac{1-\alpha}{\alpha} \right\rceil \left((\sqrt{5/2} - 1 + \varepsilon)2\alpha N \right),$$

demostrando así lo deseado. \square

Además, para los casos tales que $(\frac{1-\alpha}{\alpha}) \in \mathbb{N}$, la Proposición 3.1.4 nos asegura que

$$\begin{aligned} \text{ssp}(K_{\alpha N, (1-\alpha)N}) &\leq \left\lceil \frac{1-\alpha}{\alpha} \right\rceil \left((\sqrt{5/2} - 1 + \varepsilon) 2\alpha N \right) \\ &= \left(\frac{1-\alpha}{\alpha} \right) \left(\sqrt{5/2} - 1 + \varepsilon \right) 2\alpha N \\ &= (1-\alpha) \left(\sqrt{5/2} - 1 + \varepsilon \right) 2N. \end{aligned}$$

Ahora bien, para encontrar cotas inferiores de $\text{ssp}(K_{\alpha N, (1-\alpha)N})$ introduciremos las siguientes definiciones.

Definición 3.1.5. Dado un conjunto S y una familia \mathcal{P} de subconjuntos de S , para cada $x \in S$ definimos $\mathcal{P}(x) = \{P \in \mathcal{P} : x \in P\}$.

Definición 3.1.6. Dado un conjunto S y una familia \mathcal{P} de subconjuntos de S . Definimos para cada $n \in \mathbb{N}$

$$\mathcal{P}_n = \{P \in \mathcal{P} : |P| = n\},$$

y

$$E_n = \{x \in S : |\mathcal{P}(x)| = n\}.$$

A partir de estas definiciones podemos deducir la siguiente igualdad

$$\sum_{e \in E(G)} |\mathcal{P}(e)| = \sum_{P \in \mathcal{P}} i |E(P)|. \quad (3.1.1)$$

Notemos que $\sum_{P \in \mathcal{P}} |E(P)| \leq \sum_{i=1}^{|E(G)|} |\mathcal{P}_i|$, y además, cada camino de un grafo G en N vértices es siempre de largo a lo más $N - 1$. En base a esto, podemos acotar

superiormente $\sum_{P \in \mathcal{P}} |E(P)|$ de la siguiente forma

$$\sum_{P \in \mathcal{P}} |E(P)| \leq 1 \times |\mathcal{P}_1| + (N - 1) \times |\mathcal{P} \setminus \mathcal{P}_1| = (N - 1)|\mathcal{P}| - (N - 2)|\mathcal{P}_1|.$$

Notemos además que $|E_1| \leq |\mathcal{P}_1|$, esto debido a que si una arista e está cubierta por un único camino P , entonces P no puede cubrir ninguna otra arista f , ya que de lo contrario no habría otro camino que cubra a e y no a f , una contradicción con el hecho de que \mathcal{P} sea un sistema separador fuerte de caminos de G . A partir de esto

$$\sum_{P \in \mathcal{P}} |E(P)| \leq (N - 1)|\mathcal{P}| - (N - 2)|E_1|. \quad (3.1.2)$$

Por otro lado, podemos acotar inferiormente $\sum_{e \in E(G)} |\mathcal{P}(e)|$ como sigue

$$\begin{aligned} \sum_{e \in E(G)} |\mathcal{P}(e)| &\geq |E_1| + 2|E_2| + 3(|E(G)| - |E_2| - |E_1|) \\ &= 3|E(G)| - |E_2| - 2|E_1|, \end{aligned}$$

y por lo tanto

$$\sum_{e \in E(G)} |\mathcal{P}(e)| \geq 3|E(G)| - |E_2| - 2|E_1|. \quad (3.1.3)$$

En lo que sigue consideraremos que $|\mathcal{P}| = \beta N$ con $\beta \in \mathbb{R}$. Luego, si utilizamos la cota superior (3.1.2) e inferior (3.1.3) mostradas anteriormente en la Ecuación (3.1.1), obtenemos que

$$3|E(G)| - |E_2| - 2|E_1| \leq (N - 1)|\beta N - (N - 2)|E_1|,$$

Para $N \geq 4$ esta inecuación se reduce a

$$3|E(G)| \leq (N - 1)\beta N + |E_2|.$$

La ecuación $3|E(G)| \leq (N - 1)\beta N + |E_2|$ es válida en general para cualquier grafo G , en particular para nuestro caso de interés, el grafo $K_{\alpha N, (1-\alpha)N}$. Dado que el grafo $K_{\alpha N, (1-\alpha)N}$ posee $\zeta \binom{N}{2} = \alpha(1 - \alpha)N^2$ aristas, la cota anterior para este caso

corresponde a

$$3\zeta \binom{N}{2} \leq \beta N(N-1) + |E_2|. \quad (3.1.4)$$

A partir de estos resultados podemos enunciar la siguiente proposición.

Proposición 3.1.7. Sea $K_{\alpha N, (1-\alpha)N}$ el grafo bipartito en N vértices, con N suficientemente grande y $\alpha \in (0, 1/2)$. Entonces

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \geq (\sqrt{6\alpha(1-\alpha) + 1} - 1)N.$$

Demostración. Dado que la cantidad de aristas que pertenecen a exactamente a dos caminos siempre es menor o igual a la cantidad de pares de caminos que podemos formar en el sistema, entonces $|E_2| \leq \binom{|\mathcal{P}|}{2} \leq \frac{\beta^2 N^2}{2}$. Aplicando esto a la Ecuación (3.1.4), y considerando un N suficientemente grande se tiene que

$$3\zeta \frac{N^2}{2} \leq \beta N^2 + \frac{\beta^2 N^2}{2}.$$

De donde obtenemos la ecuación $3\zeta \leq 2\beta + \beta^2$. Para esta ecuación cuadrática podemos completar cuadrados para obtener la ecuación

$$3\zeta + 1 \leq (\beta + 1)^2.$$

Luego, dado que $\zeta \binom{N}{2} = \alpha(1-\alpha)N^2$, se tiene que $\zeta \geq 2\alpha(1-\alpha)$, y por lo tanto

$$6\alpha(1-\alpha) + 1 \leq (\beta + 1)^2,$$

y de esta manera concluimos que

$$\sqrt{6\alpha(1-\alpha) + 1} - 1 \leq \beta.$$

Demostrando así que $\text{ssp}(K_{\alpha N, (1-\alpha)N}) \geq (\sqrt{6\alpha(1-\alpha) + 1} - 1)N$ □

Si bien hemos calculado una cota inferior para $\text{ssp}(K_{\alpha N, (1-\alpha)N})$, no es la única cota que podemos obtener por medio de este razonamiento. Nuevamente consideremos la Ecuación (3.1.1)

$$\sum_{e \in E(G)} |\mathcal{P}(e)| = \sum_{P \in \mathcal{P}} |E(P)|.$$

Dado que grafo $K_{\alpha N, (1-\alpha)N}$ posee caminos de largo a lo más $2\alpha N$ y posee $\alpha(1-\alpha)N^2$ aristas, si consideramos que $|\mathcal{P}| = \beta N$ es posible deducir de manera similar a la Ecuación (3.1.4) la siguiente ecuación

$$3\alpha(1-\alpha)N^2 \leq 2\alpha\beta N^2 + |E_2|. \quad (3.1.5)$$

A partir de esta nueva ecuación, podemos enunciar la siguiente proposición.

Proposición 3.1.8. Sea $K_{\alpha N, (1-\alpha)N}$ el grafo bipartito en N vértices con $\alpha \in (0, 1/2)$. Entonces

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \geq (1-\alpha)N.$$

Demostración. Considere un grafo auxiliar H con $V(H) = \mathcal{P}$, tal que una arista $P_1P_2 \in E(H)$, si y solo si, existe una arista $e \in G$ tal que $|\mathcal{P}(e)| = \{P_1, P_2\}$. Por definición del grafo H se tiene que $|E_2| = |E(H)|$ y $\Delta(H) \leq 2\alpha N$. Esto ultimo es debido a que si consideramos $\Delta(H) > 2\alpha N$, entonces existiría un camino P en \mathcal{P} de largo mayor a $2\alpha N$, siendo esto imposible para el grafo $K_{\alpha N, (1-\alpha)N}$. Luego, por medio de este grafo auxiliar H obtenemos que

$$|E_2| = \frac{1}{2} \sum_{v \in V(H)} d_H(v) \leq \alpha\beta N^2.$$

Si aplicamos esta cota a la Ecuación (3.1.5), obtenemos que

$$3(1-\alpha)\alpha N^2 \leq 2\alpha\beta N^2 + \alpha\beta N^2,$$

de donde se deduce que $1-\alpha \leq \beta$, y por lo tanto

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \geq (1-\alpha)N,$$

probando así lo deseado. □

Si bien ambas proposiciones permiten obtener cotas inferiores de $\text{ssp}(K_{\alpha N, (1-\alpha)N})$, podemos ser aun más precisos. Notemos que las funciones $F_1(\alpha) = \sqrt{6\alpha(1-\alpha)+1} - 1$ y $F_2(\alpha) = 1-\alpha$ como funciones en términos de α son

iguales para el valor $\alpha = 3/7$. Además se tiene que

$$\begin{cases} \sqrt{6\alpha(1-\alpha)} + 1 - 1 \leq 1 - \alpha & \text{si } 0 < \alpha \leq 3/7, \\ \sqrt{6\alpha(1-\alpha)} + 1 - 1 \geq 1 - \alpha & \text{si } 3/7 \leq \alpha \leq 1/2. \end{cases}$$

Para más detalles observe la siguiente Figura 3.1.2, donde además se presenta la cota superior $F_3(\alpha) = \lceil \frac{1-\alpha}{\alpha} \rceil \left((\sqrt{5/2} - 1 + \varepsilon)2\alpha \right)$ como función de α .

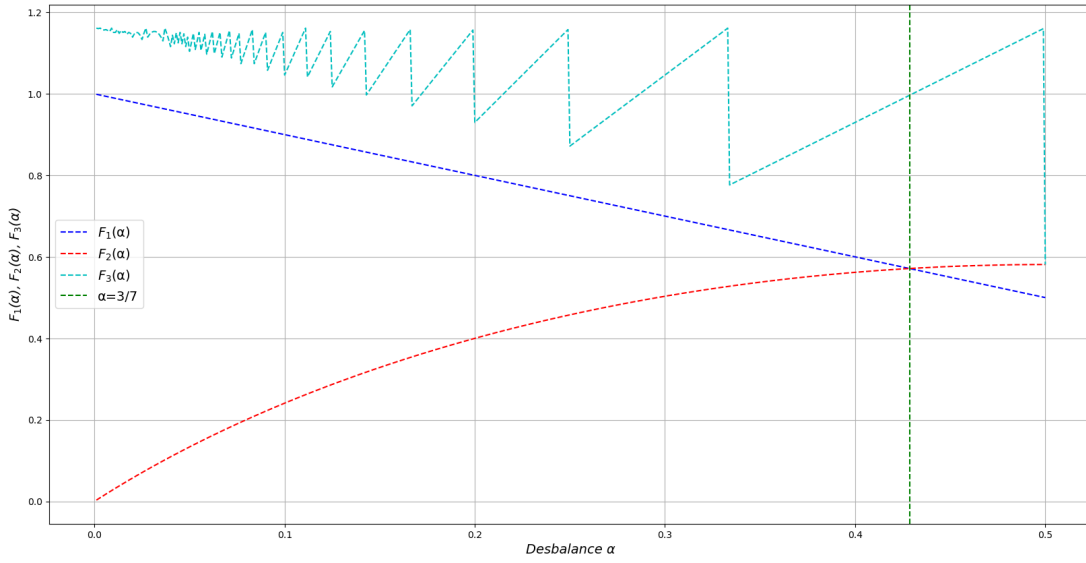


Figura 3.1.2: El gráfico presenta las funciones $F_1(\alpha) = (1 - \alpha)$, $F_2(\alpha) = \sqrt{6\alpha(1-\alpha)} + 1 - 1$ y $F_3(\alpha) = \lceil \frac{1-\alpha}{\alpha} \rceil \left((\sqrt{5/2} - 1 + \varepsilon)2\alpha \right)$ y el punto de intersección $\alpha = 3/7$ de $F_1(\alpha)$ y $F_2(\alpha)$.

Dado que $\text{ssp}(K_{\alpha N, (1-\alpha)N})$ corresponde al mínimo tamaño de un sistema separador fuerte de caminos para el grafo $K_{\alpha N, (1-\alpha)N}$, podemos concluir entonces la siguiente proposición.

Proposición 3.1.9. Sea $\alpha \in (0, 1/2)$, $N \in \mathbb{N}$ suficientemente grande y $K_{\alpha N, (1-\alpha)N}$ el grafo bipartito en N vértices. Entonces

$$\text{ssp}(K_{\alpha N, (1-\alpha)N}) \geq \begin{cases} (1 - \alpha)N & \text{si } 0 < \alpha \leq 3/7, \\ (\sqrt{6\alpha(1-\alpha)} + 1 - 1)N & \text{si } 3/7 < \alpha < 1/2. \end{cases}$$

Capítulo 4

Algoritmo determinista de sistemas separadores para el grafo $K_{n,m}$

4.1. Construcción de un sistema separador para el grafo $K_{n,m}$

El grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23] demostró que para cualquier grafo G en N vértices se tiene que $\text{ssp}(G) \leq 19N$. A continuación estudiaremos cómo se aplica su demostración para construir un sistema separador fuerte de caminos para el grafo $K_{n,m}$ con $n < m$.

Dado que estudiaremos su demostración para el grafo bipartito completo, en lo que sigue consideraremos el grafo bipartito $K_{n,m}$ con partición de vértices A, B , donde $n < m$, $|A| = n$ y $|B| = m$. Consideraremos las siguientes etiquetas para la partes $A = \{x_2, x_4, \dots, x_{2n}\}$ y $B = \{x_1, x_3, \dots, x_{2m-1}\}$.

Para un grafo bipartito completo $K_{n,m}$ tal que $n < m$ podemos deducir a qué parte de la partición pertenecen los vértices extremos de P .

Proposición 4.1.1. Sean $n, m \in \mathbb{N}$ con $n < m$ y sea $K_{n,m}$ el grafo bipartito con partición de vértices A, B , donde $|A| = n$ y $|B| = m$. Sea P un camino de largo máximo en $K_{n,m}$, entonces los extremos del camino P pertenecen a B .

Demostración. Razonaremos por contradicción. Considere un vértice $x \in A$ tal que x es un vértice extremo de P . Dado $K_{n,m}$ es un grafo completo no existe un

vértice $y \in B \setminus E(P)$, pues si esto ocurre el camino P' tal que $E(P') = E(P) \cup \{xy\}$ sería un camino más largo que el camino de largo máximo P . Dado que no existe un vértice $y \in B \setminus E(P)$ y P es un camino de largo máximo, se deduce que a lo más hay n vértices en la parte B , es decir $m \leq n$, siendo así una contradicción. \square

Dado un grafo G y vértices u, v en G , sea $P = uu' \cdots v$ un camino de u a v . Si $x \in V(P)$ es vecino de $u \neq u'$ en G y x^- es el vértice que precede a x en P , entonces $P' = P - xx^- + ux$ es un camino en G tal que $V(P') = V(P)$. Decimos que P' se ha obtenido de P mediante un *intercambio elemental* con v fijo. Un camino obtenido de P mediante una secuencia (posiblemente vacía) de intercambios elementales con v fijo se dice que es un camino *derivado* de P .

El conjunto de vértices finales de los caminos derivados de P que son distintos de v se denota por $S_v(P)$. Dado que todos los caminos derivados de P tienen el mismo conjunto de vértices que P , tenemos $S_v(P) \subseteq V(P)$.

Ahora bien para el caso del grafo $K_{n,m}$ podemos ser aun más precisos respecto al conjunto $S_v(P)$.

Proposición 4.1.2. Sea $K_{n,m}$ el grafo bipartito con partición de vértices A, B , donde $n < m$, $|A| = n$ y $|B| = m$. Sea $P = u \cdots v$ un camino de largo máximo del grafo $K_{n,m}$, entonces $S_v(P) = (B \cap V(P)) - \{v\}$.

Demostración. Dado que el vértice u tiene por vecinos a todos los vértices de A , y además todos los vértices de A pertenecen al camino, entonces sea $y \in A$ y sea $x \in B \cap V(P)$ el vértice que precede al vértice y en el camino P . Luego el camino

$$P' = P - xy + uy = xy \cdots v,$$

se ha obtenido de P mediante un intercambio elemental con v fijo. Dado que podemos repetir el proceso para cada vértice y en A y su vértice predecesor $x \in B$, concluimos que $S_v(P) = (B \cap V(P)) - \{v\}$. \square

Ahora bien, sean H y G grafos (no necesariamente disjuntos). Decimos que un conjunto de caminos \mathcal{P} en $H \cup G$ *separa* H de G si para cada par $(e, f) \in E(H) \times E(G)$ con $e \neq f$ hay un camino en \mathcal{P} que contiene e y no contiene f .

A partir de estas definiciones y proposiciones empezamos a analizar la demostración realizada por el grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23]

enfocándonos en el caso del grafo $K_{n,m}$.

Sea $P = x_1x_2x_3 \cdots x_{2n}x_{2n+1}$ un camino de largo máximo y $v = x_{2n+1}$. Sean $S := S_{x_{2n+1}}(P)$, H el subgrafo de $K_{n,m}$ inducido por las aristas con al menos un vértice en S y sea $n' = |V(H)|$. Para el caso del grafo $K_{n,m}$ se tiene por la Proposición 4.1.2 que el subgrafo H corresponde al grafo bipartito completo con partición de vértices A y $(B \cap V(P)) - \{x_{2n+1}\}$ (Observe la Figura 4.1.1). Dado

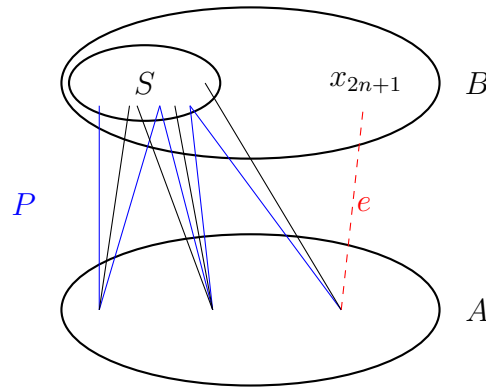


Figura 4.1.1: Ejemplo del grafo H para el grafo $K_{n,m}$.

que el largo del camino de largo máximo en el grafo $K_{n,m}$ es $2n$ se tiene que $n' = n + n = 2n$. Podemos deducir que este grafo H contiene a todas las aristas del camino P salvo su última arista e que conecta con el vértice x_{2n+1} . Además este grafo H no contiene las n aristas que conectan los vértices de la parte A con el vértice x_{2n+1} .

Por otro lado se define también el grafo $G' = K_{n,m} \setminus S$. Dado que $K_{n,m}$ corresponde a la unión arista disjunta de los grafos G' y H , se concluye que para el grafo bipartito $K_{n,m}$, el grafo G' corresponde al grafo bipartito completo con partición de vértices A y $(B \cap V(P)^c) \cup \{x_{2n+1}\}$. Podemos deducir que este grafo G' no contiene a las aristas del camino P , con excepción de su última arista $e = x_{2n}x_{2n+1}$ (Observe la Figura 4.1.2).

Dado que $K_{n,m}$ y S son distintos del vacío, construiremos una familia de caminos que separe fuertemente al grafo H . Luego, la idea que proponemos es repetir el mismo proceso pero sobre el subgrafo G' eliminando las aristas de $H_0 = H$ del grafo $K_{n,m}$ y seleccionando un nuevo camino P , repitiendo este proceso una cantidad de $l = \lceil \frac{m}{n} \rceil$ veces, hasta separar todas las aristas del grafo $K_{n,m}$ (vea la Figura 4.1.3).

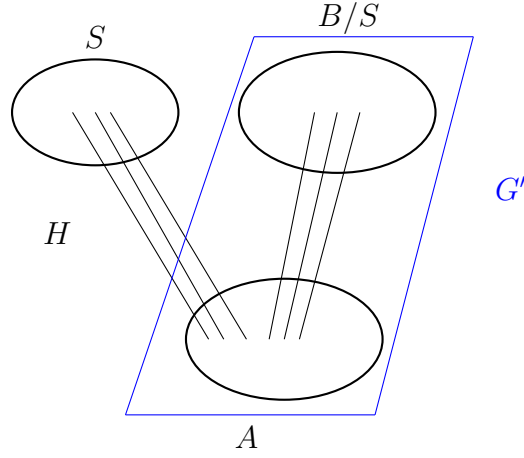


Figura 4.1.2: Ejemplo del grafo G' para el grafo $K_{n,m}$.

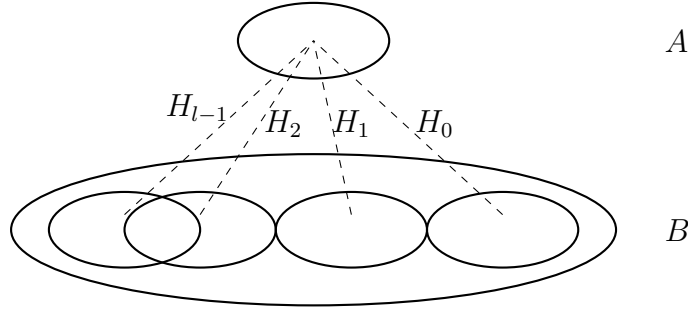


Figura 4.1.3: Ejemplo del proceso para encontrar un sistema separador fuerte para el grafo $K_{n,m}$.

En lo que sigue, construiremos una familia de caminos \mathcal{P} que separa H de $K_{n,m}$. Para construir la familia de caminos \mathcal{P} , se define el conjunto $P_S = E(P) \cap E(H)$ como el conjunto de aristas de P que tienen al menos un vértice en S . Para el grafo $K_{n,m}$ se tiene que $P_S = E(P) - \{x_{2n+1}\}$. Luego, considere el grafo $H' = H \setminus P_S$. El grafo H' corresponde a un grafo bipartito. El conjunto de vértices de H' corresponde a $V(H') = \{x_1, x_2, x_3, \dots, x_{2n}\}$ y su conjunto de aristas es

$$E(H') = \{x_i y_j : \text{para todo } i \in \{1, \dots, 2n\} \text{ tal que } (i, j) \notin \{(i, i), (i, i+1)\}\}.$$

Sea \mathcal{P}_S la familia de caminos donde cada uno de los caminos consta de una única arista en P_S , es decir, $\mathcal{P}_S = \{\{e\} : e \in P_S\}$. Dado que $P_S = E(P) - \{x_{2n+1}\}$, se tiene que $|\mathcal{P}_S| = 2n - 1$. Luego, sea \mathcal{D} una descomposición por caminos aristas disjuntos para H' . Note que $\mathcal{P}_S \cup \mathcal{D}$ separa (i) H de G' ; (ii) H' de P ; y (iii) P_S de $K_{n,m}$. A continuación construiremos esta familia \mathcal{D} tal que $|\mathcal{D}| = n$ para el

grafo H' .

Para construir la familia de caminos \mathcal{D} para el grafo H' , primero construiremos una descomposición \mathcal{D}' de caminos aristas disjuntos del grafo $K_{n,n}$.

Sea $K_{n,n}$ el grafo bipartito completo en $2n$ vértices, con $n \in \mathbb{N}$. Sea $A = \{x_0, x_1, \dots, x_{n-1}\}$ y $B = \{y_0, y_1, \dots, y_{n-1}\}$ la partición de vértices de $K_{n,n}$. Luego para cada $k \in \{0, \dots, n-1\}$ definimos el conjunto de aristas

$$F_k := \{x_i y_j \in E(K_{n,n}), \text{ tal que } j \equiv i + k \pmod{n}\}.$$

Consideremos un ejemplo de estos conjuntos F_k para el caso de $n = 4$. Observe la figura 4.1.4, en ella se representan los conjuntos F_0, F_1, F_2, F_3 para el caso de $n = 4$.

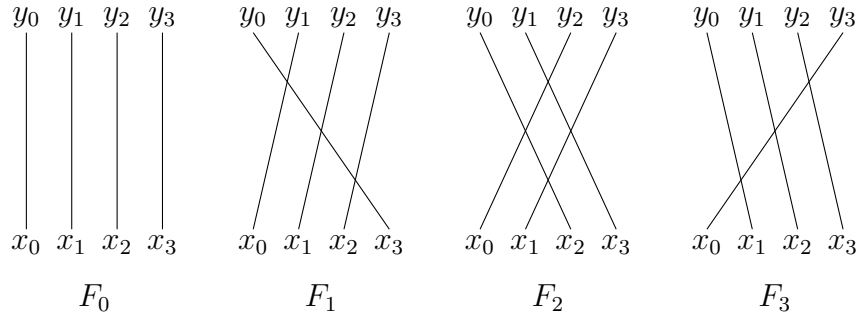


Figura 4.1.4: Ejemplos de conjuntos F_0, F_1, F_2, F_3 para el caso de $n = 4$.

Note que cada conjunto F_k contiene exactamente n aristas distintas. Dados estos conjuntos F_k podemos enunciar la siguiente proposición.

Proposición 4.1.3. Sea $n \in \mathbb{N}$ y $K_{n,n}$ el grafo bipartito completo. Entonces

$$E(K_{n,n}) = \bigsqcup_{k=0}^{n-1} F_k.$$

Demostración. Dada la definición de los conjuntos F_k es claro que para cualquier par de índices distintos i, j se tiene que $F_i \cap F_j = \emptyset$. Luego, demostraremos la igualdad por doble inclusión. Nuevamente, dada la definición de los conjuntos F_k se tiene que $\bigsqcup_{k=0}^{n-1} F_k \subseteq E(K_{n,n})$. Por otro lado, sea $e = x_i y_j \in E(K_{n,n})$. Dado que

existe un único $k \in \{0, \dots, n-1\}$ tal que $j \equiv i + k \pmod{n}$, entonces existe un único conjunto F_k tal que $e \in F_k$, y por lo tanto, $E(K_{n,n}) \subseteq \bigsqcup_{k=0}^{n-1} F_k$. \square

Para construir la familia de caminos \mathcal{D}' separaremos en casos para n par e impar y consideraremos sumas modulo n para todos los índices. Para cada $k \in \{0, 1, \dots, n-1\}$ si n es impar se define el camino

$$P_k = y_k x_k y_{k+1} x_{k-1} y_{k+2} x_{k-2} \dots y_{k-\lfloor \frac{n}{2} \rfloor} x_{k+\lfloor \frac{n}{2} \rfloor}.$$

En cambio, si n es par se define para cada $k \in \{0, 1, \dots, n-1\}$ el camino

$$P_k = y_k x_k y_{k+1} x_{k-1} y_{k+2} x_{k-2} \dots y_{k-\frac{n}{2}-1} x_{k+\frac{n}{2}-1} y_{k-\frac{n}{2}}.$$

Note que tanto para el caso par e impar los caminos P_k contienen exactamente una arista de cada conjunto F_0, \dots, F_{n-1} . Además, para cada par de índices distintos i y j los caminos P_i y P_j cumplen que $E(P_i) \cap E(P_j) \neq \emptyset$, esto es debido a que cada par de caminos utilizan aristas distintas de cada conjunto F_0, \dots, F_{n-1} . Dado que cada camino utiliza una arista distinta de las n aristas de cada conjunto F_0, \dots, F_{n-1} , por medio de la Proposición 4.1.3 podemos concluir que para toda arista $e \in E(K_{n,n})$ existe único $r \in \{0, \dots, n-1\}$ tal que $e \in P_r$. De esta manera podemos enunciar la siguiente proposición.

Proposición 4.1.4. Sea $n \in \mathbb{N}$ y $K_{n,n}$ el grafo bipartito completo. Entonces la familia $\mathcal{D}' = \{P_0, P_1, \dots, P_{n-1}\}$ es una descomposición por caminos aristas disjuntos de $K_{n,n}$.

Demostración. Dada la definición de los caminos P_k y la Proposición 4.1.3 sabemos que $\bigsqcup_{k=0}^{n-1} E(P_k) \subseteq E(K_{n,n})$. Luego, sea $e = x_i y_j \in E(K_{n,n})$ una arista cualquiera. Dado que existe un único $k \in \{0, \dots, n-1\}$ tal que $j \equiv i + k \pmod{n}$, entonces existe un único conjunto F_k tal que $e \in F_k$, y por lo tanto, un único camino $P \in \mathcal{D}'$ tal que $e \in P$. De esta manera $E(K_{n,n}) \subseteq \bigsqcup_{k=0}^{n-1} E(P_k)$, y por lo tanto, \mathcal{D}' es una descomposición por caminos aristas disjuntos de $K_{n,n}$. \square

Dado que buscamos una descomposición \mathcal{D}' de caminos aristas disjuntos de H' notemos que el grafo H corresponde a un grafo $K_{n,n}$ sin las aristas que pertenecen

al camino P . Además, para este orden de vértices el camino de largo máximo que hemos considerado corresponde a $P = y_1x_1y_2x_2 \dots x_ny_{n+1}$, y por lo tanto, se cumple que para toda arista $e \in P$, se tiene que $e \in F_0$ ó $e \in F_1$. Dada la definición de cada camino $P_k \in \mathcal{D}'$, podemos definir para cada $k \in \{0, \dots, n-1\}$ los caminos

$$Q_k = P_k - y_kx_ky_{k+1}.$$

Dado que los caminos Q_k no contienen aristas de los conjuntos F_0 y F_1 , entonces no utilizan aristas en el camino P , y por lo tanto, son caminos en el grafo H' . De esta manera al eliminar para cada camino $P \in \mathcal{D}'$ las aristas que pertenecen a los conjuntos F_0 y F_1 , obtenemos un familia de caminos $\mathcal{D} = \{Q_0, \dots, Q_{n-1}\}$ que corresponde a una descomposición por caminos aristas disjuntos de H' de tamaño n .

Dado que hemos construido la familia de caminos \mathcal{D} solo resta por crear un conjunto de caminos que separen H' de si mismo. Para realizar esta tarea, el grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23] ordenan los vértices del grafo según el orden del camino P . Luego para cada $k \in \{1, \dots, 2n'\}$ se define el conjunto $M_k = \{x_ix_j \in E(H') : i + j = k\}$, y para $k \in \{1, \dots, 3n'\}$ el conjunto $N_k = \{x_ix_j \in E(H') : i + 2j = k\}$, donde para cada par de índices i, j en ambos conjuntos se impone que $i < j$. Para cada M_k definiremos un camino P_k que utilice las aristas de M_k y algunas aristas del camino P . Análogamente definiremos para cada N_k los caminos Q_k .

Dado que para el grafo $K_{n,m}$ se tiene que $n' = 2n$, hemos de analizar los conjuntos $M_k = \{x_ix_j \in E(H') : i + j = k\}$ para cada $k \in \{1, \dots, 4n\}$, y los conjuntos $N_k = \{x_ix_j \in E(H') : i + 2j = k\}$ para $k \in \{1, \dots, 6n\}$.

Notemos que los conjuntos N_k debido al orden de los índices para los vértices de $K_{n,m}$ para cada $k < 9$ ó $k > 6n - 3$ se tiene que $N_k = \emptyset$, de esta manera la cantidad de conjuntos N_k 's es menor igual a $6n - 11$. Para entender mejor como son los caminos Q_k para algún k fijo consideremos el caso para $k = 4n + 1$, y observe la siguiente figura 4.1.5. En la Figura 4.1.5 se muestra un camino Q_{4n+1} con color azul. Una línea azul y celeste representa las aristas del camino Q_{4n+1} que pertenecen al camino P y con una línea azul solida las aristas de Q_{4n+1} tales que $x_ix_j \in N_{4n+1}$.

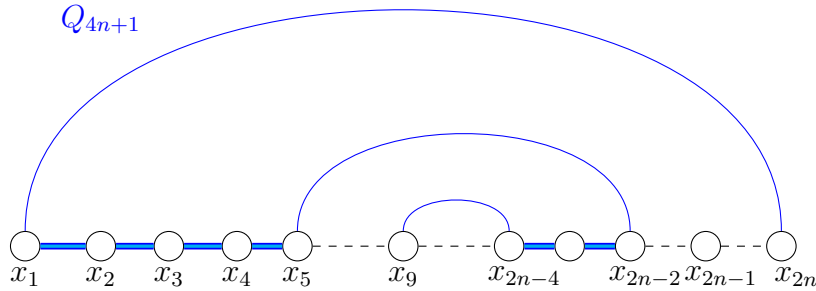


Figura 4.1.5: Ejemplo de camino Q_k para $k = 4n + 1$.

Para los conjuntos N_k si bien hemos logrado demostrar que son exactamente $6n - 12$, para nuestro análisis solo consideraremos que la cantidad de conjuntos N_k es menor a $6n$, y en consecuencia, la cantidad de caminos Q_k se encuentra acotada superiormente por $6n$.

Ahora bien, analizaremos en más detalle los conjuntos M_k . Dado el orden de vértices que hemos considerado para cada $k < 5$ ó $k > 4n - 3$ se tiene que $M_k = \emptyset$, además para cualquier arista $x_i x_j$ se cumple que $i + j \pmod 2 = 0$, de forma que para cualquier k impar se tiene que $M_k = \emptyset$. Para construir los caminos P_k 's, considere el caso $k = 2n + 1$ y observe la siguiente figura 4.1.6. En la Figura 4.1.6 se muestra un camino P_{2n+1} con color rojo. Análogamente, una línea roja y amarilla representa las aristas del camino P_{2n+1} que pertenecen al camino P y con una línea roja sólida las aristas de P_{2n+1} tales que $x_i x_j \in M_{2n+1}$.

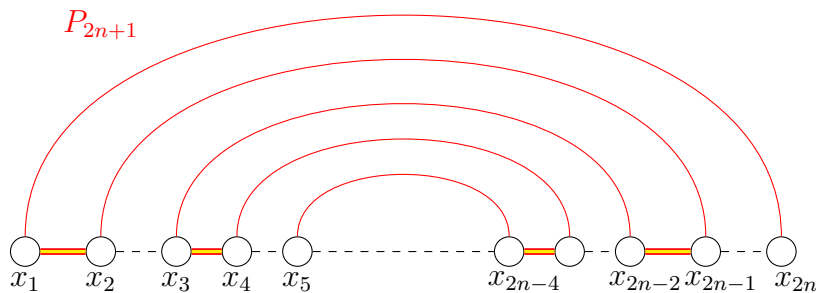


Figura 4.1.6: Ejemplo de camino P_k para $k = 2n + 1$.

El camino P_{2n+1} comienza con el par de vértices $x_{2n} x_1$ y luego va alternando entre vértices de índices impares y pares de forma que los índices de estos vértices que pertenecen a H' sumen exactamente $k = 2n + 1$. Por medio de este ejemplo

podemos encontrar para cualquier $5 \leq k \leq 2n+1$ impar un camino P_k que cumpla con lo requerido. Este camino P_k comienza con el par de vértices $x_{k-1}x_1$ y repite la idea del camino P_{2n+1} formando así el camino

$$P_k = x_{k-1}x_1x_2x_{k-2}x_{k-3}x_3x_4 \cdots x_{l_k-1}x_{l_k}$$

donde

$$x_{l_k-1}x_{l_k} = \begin{cases} x_{\frac{(k-3)}{2}}x_{\frac{(k+3)}{2}} & \text{si } k \in \{5, 9, 13, \dots, 2n-1\}, \\ x_{\frac{(k+3)}{2}}x_{\frac{(k-3)}{2}} & \text{si } k \in \{7, 11, 15, \dots, 2n+1\}. \end{cases}$$

Análogamente para cualquier $2n+1 < k \leq 4n-1$ impar podemos tomar de ejemplo el caso para $k = 2n+3$, que se encuentra representado en la siguiente figura 4.1.7.

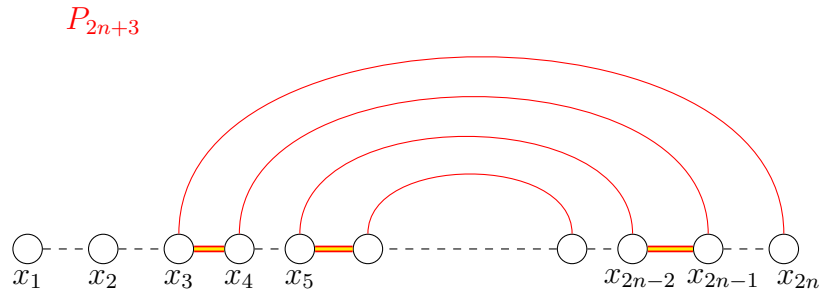


Figura 4.1.7: Ejemplo de camino P_k para $k = 2n+3$

De esta manera, para cualquier $2n+1 < k \leq 4n-1$ impar podemos construir un camino P_k que comienza con el par de $x_{2n}x_{k-(2n)}$ y repita el patrón del camino P_{2n+3} de la Figura 4.1.7. Es decir, el camino P_k para este caso viene dado por

$$P_k = x_{2n}x_{k-2n}x_{k-2n+1}x_{2n-1}x_{2n-2} \cdots x_{l_k-1}x_{l_k}$$

donde análogamente

$$x_{l_k-1}x_{l_k} = \begin{cases} x_{\frac{(k-3)}{2}}x_{\frac{(k+3)}{2}} & \text{si } k \in \{2n+3, 2n+7, \dots, 4n-3\}, \\ x_{\frac{(k+3)}{2}}x_{\frac{(k-3)}{2}} & \text{si } k \in \{2n+5, 2n+9, \dots, 4n-5\}. \end{cases}$$

Notemos que la cantidad de caminos P_k es exactamente $2n-3$, esto debido a la cantidad de impares entre 5 y $4n-3$ es exactamente $(4n-8)/2 + 1 = 2n-3$.

Finalmente los conjuntos de caminos de P_k 's y Q_k 's juntos separan cada par de aristas en H' . En efecto, sean $x_i v_j$ y $x_{i'} x_{j'}$ dos aristas distintas de H' . Cada una de estas aristas pertenece exactamente a un M_k y exactamente a un N_k . Si pertenecen a diferentes M_k , pertenecen a diferentes P_k y están separados. Del mismo modo, si pertenecen a diferentes N_k . Por lo tanto, podemos suponer que $i + j = i' + j'$ y $i + 2j = i' + 2j'$. De donde se concluye que $j = j'$ y $i = i'$. Por lo tanto, cualquier par de aristas distintas en H' están separadas por esta colección de caminos P_k 's y Q_k 's.

De esta manera la familia de caminos \mathcal{P} que corresponde a los caminos P_k 's \cup Q_k 's $\cup \mathcal{D} \cup \mathcal{P}_S$, separa fuertemente al conjunto $E(H)$ y separa H del grafo $K_{n,m}$. Este sistema separador \mathcal{P} cumple que

$$\begin{aligned} |\mathcal{P}| &= \left| \bigcup_k P_k \right| + \left| \bigcup_k Q_k \right| + |\mathcal{D}| + |\mathcal{P}_S| \\ |\mathcal{P}| &\leq 2n - 3 + 6n + n + 2n - 1 \\ |\mathcal{P}| &\leq 11n - 4. \end{aligned}$$

Dado que deseamos encontrar un sistema separador fuerte del grafo $K_{n,m}$, repetiremos este proceso una cantidad $l = \lceil \frac{m}{n} \rceil$ de veces (recuerde la Figura 4.1.3). Para realizar esta tarea, seleccionaremos una cantidad $l = \lceil \frac{m}{n} \rceil$ de caminos, donde para cada $i \in \{0, \dots, l - 2\}$ consideraremos el camino de largo máximo

$$P_i = x_{1+i(2n)} x_2 \cdots x_{2n} x_{2n+1+i(2n)}.$$

Notemos que estos caminos P_i son todos aristas disjuntos entre ellos. Además, consideramos el camino de largo máximo

$$P_l = x_{2m-2n-1} x_2 x_{2m-2n+1} x_4 \cdots x_n x_{2m-1}.$$

Para cada grafo H_i inducido por un camino P_i repetiremos la construcción del sistema separador \mathcal{P} obteniendo un sistema separador \mathcal{P}_i . Este proceso puede ser representado por el algoritmo 1.

Dado que para cualquier sistema \mathcal{P}_i se tiene que $|\mathcal{P}_i| \leq 11n - 4$ concluimos que el

Algoritmo 1 Algoritmo determinista**Input:** Grafo bipartito completo $K_{n,m}$ **Output:** Un sistema separador fuerte del grafo $K_{n,m}$.

- 1: **for** $t = 0, 1, \dots, l - 1$ **do**
- 2: Seleccionar el camino P_i en el grafo $K_{n,m}$
- 3: Definir el subgrafo H_i
- 4: Generar el sistema separador \mathcal{P}_i
- 5: **return** $\mathcal{P}_0, \dots, \mathcal{P}_{l-1}$

sistema de caminos $\bigcup_{i=0}^{l-1} \mathcal{P}_i$ generado por el Algoritmo 1 es un sistema separador fuerte de caminos de $K_{n,m}$ de tamaño menor igual a $l(11n - 4) = \lceil \frac{m}{n} \rceil (11n - 4)$.

4.2. Implementación computacional

En esta sección realizaremos la implementación computacional del Algoritmo 1. El objetivo de esta implementación es corroborar el análisis teórico realizado en la sección anterior por medio de distintas funciones en Python. Las funciones que permiten realizar esta implementación se encuentran en el Anexo 9.

4.2.1. Experimentos computacionales para la cota superior

$$\lceil \frac{m}{n} \rceil (11n - 4)$$

En esta sección presentaremos los distintos resultados experimentales obtenidos por la implementación computacional del Algoritmo 1 para distintas cantidades de vértices $N \in \{100, 250, 500, 750, 1000\}$ y distintas particiones A, B . Cabe destacar que antes de realizar estas pruebas para estas cantidades de vértices se verifico para tamaños mas pequeños que el sistema encontrado corresponde a un sistema separador fuerte.

En primer lugar compararemos el tamaño del sistema separador generado por la implementación computacional del Algoritmo 1 y la cota superior $\lceil \frac{m}{n} \rceil (11n - 4)$ para un cantidad de vértices variable. Dado que los resultados fueron similares para las distintas particiones de vértices a continuación solo mostraremos los resultados obtenidos para una cantidad de vértices $N \in \{100, 250, 500, 750, 100\}$ y $l = 2$.

La Figura 4.2.1 nos muestra que el tamaño del sistema obtenido por la

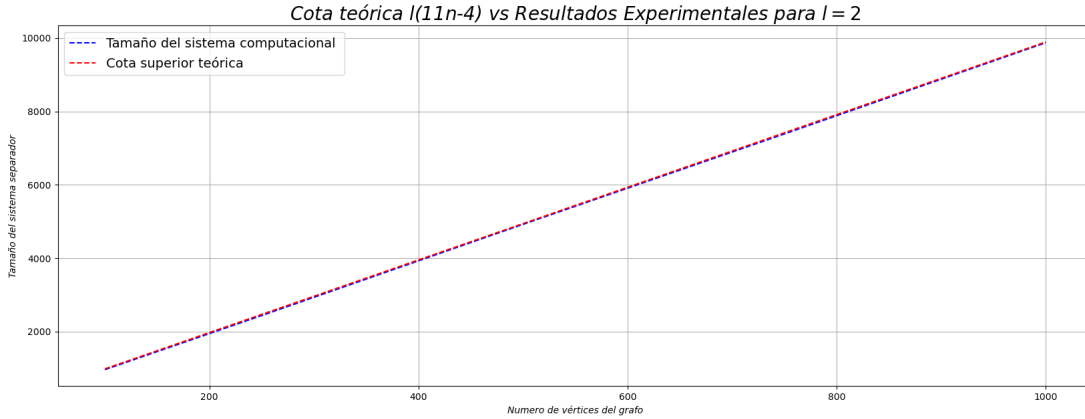


Figura 4.2.1: Cota superior teórica para el tamaño del sistema separador y el tamaño del sistema separador computacional para $l = 2$. El gráfico presenta en su eje de abscisas los distintas cantidades de vértices $N \in \{100, 250, 500, 750, 1000\}$ de los grafos $K_{n,m}$ considerando $l = 2$. En el eje de las ordenadas se presenta el tamaño del sistema separador. En rojo se marca la cota superior teórica $\lceil \frac{m}{n} \rceil (12n - 4)$ para el tamaño del sistema separador. En azul se marca el tamaño del sistema separador obtenido por la implementación computacional.

implementación computacional del Algoritmo 1 es casi idéntica a la cota superior obtenida por nuestro análisis. Estas diferencias se deben únicamente a la cota superior utilizada para el análisis de los caminos Q_k .

Ahora bien, nuevamente compararemos el tamaño del sistema separador generado por la implementación computacional del Algoritmo 1 y la cota superior $\lceil \frac{m}{n} \rceil (11n - 4)$ pero para una cantidad de vértices fija y valores de $l = \lceil \frac{m}{n} \rceil$ variable. Dado que los resultados también fueron similares para las distintas combinaciones de vértices, solo mostraremos los resultados obtenidos para un cantidad de vértices $N = 1000$ y distintos valores $l \in \{2, 3, 4, 6, 9\}$.

La Figura 4.2.2 nos muestra que el tamaño del sistema obtenido por la implementación computacional del Algoritmo 1 es también muy similar a nuestra cota superior teórica a medida de que aumenta el valor de l para una cantidad de vértices fija. Para este experimento se puede notar como el valor l no afecta directamente en el tamaño del sistema separador generado por la implementación computacional.

4.2.2. Tiempos de ejecución

A continuación presentamos los tiempos de ejecución de la implementación computacional del Algoritmo 1 para los distintos tamaños N y valores n, m utilizados en nuestros pruebas. Donde podemos notar que a pesar de que el Algoritmo 1 debe repetir el proceso de obtener el sistema \mathcal{P} una cantidad $l = \lceil \frac{m}{n} \rceil$

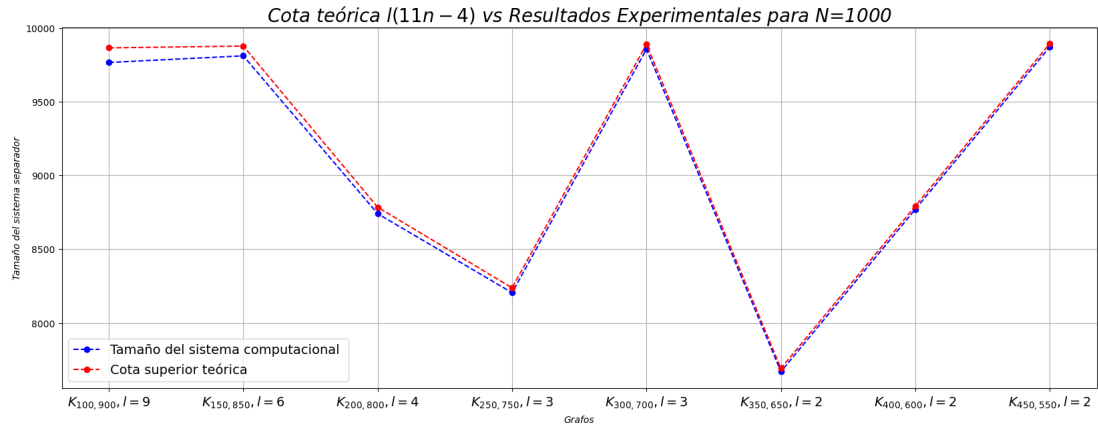


Figura 4.2.2: Cota superior teórica para el tamaño del sistema separador y el tamaño del sistema separador computacional para $N = 1000$. El gráfico presenta en su eje de abscisas los distintos grafos $K_{n,m}$ considerando algún valor $l \in \{2, 3, 4, 6, 9\}$ y una cantidad de vértices $N = 1000$. En el eje de las ordenadas se presenta el tamaño del sistema separador. En rojo se marca la cota superior teórica $\lceil \frac{m}{n} \rceil (11n - 4)$ para el tamaño del sistema separador. En azul se marca el tamaño del sistema separador obtenido por la implementación computacional.

de veces, esta cantidad $l = \lceil \frac{m}{n} \rceil$ no afecta en gran medida los tiempos de ejecución de la implementación computacional del Algoritmo 1.

Notemos que requerimos una cantidad $O(nm)$ de operaciones para generar cada uno de los caminos de las familias P_k 's y Q_k 's. Por otro lado, para encontrar cada uno de los caminos restantes de la colección se requiere de una cantidad $O(n)$ de operaciones para cada iteración. Dado que repetimos la búsqueda del sistema \mathcal{P}_i una cantidad constante l de veces podemos concluir que el Algoritmo 1 requiere una cantidad $O(n^2m) = O(nE(K_{n,m}))$ de operaciones para realizar su tarea. Este análisis no considera el costo computacional de las funciones utilizadas.

n	m	l	Media tiempo de ejecución	Desviación estándar
10	90	9	49,7 [ms]	$\pm 4,67$ [ms]
15	75	5	46 [ms]	$\pm 3,67$ [ms]
20	80	4	46,2 [ms]	$\pm 2,84$ [ms]
25	75	3	62,2 [ms]	± 33 [ms]
30	70	3	50,6 [ms]	$\pm 7,15$ [ms]
35	65	2	45,1 [ms]	$\pm 1,09$ [ms]
40	60	2	48,5 [ms]	$\pm 7,3$ [ms]
45	55	2	44,9 [ms]	± 9494 [μ s]

Cuadro 4.2.1: Tabla tiempos de ejecución para $N = 100$.

n	m	l	Media tiempo de ejecución	Desviación estándar
25	225	9	880 [ms]	$\pm 63,2$ [ms]
38	218	6	941 [ms]	$\pm 83,4$ [ms]
50	250	5	884 [ms]	$\pm 49,4$ [ms]
63	187	3	910 [ms]	$\pm 73,6$ [ms]
75	125	2	1,31 [s]	± 508 [ms]
88	162	2	1,26 [s]	± 322 [ms]
100	150	2	914 [ms]	$\pm 86,8$ [ms]
113	137	2	918 [ms]	$\pm 75,5$ [ms]

Cuadro 4.2.2: Tabla tiempos de ejecución para $N = 250$.

n	m	l	Media tiempo de ejecución	Desviación estándar
50	450	9	11,3 [s]	$\pm 3,34$ [s]
75	425	6	14 [s]	$\pm 4,12$ [s]
100	400	4	9,39 [s]	± 416 [ms]
125	375	3	10,7 [s]	$\pm 2,26$ [s]
150	350	3	9,98 [s]	$\pm 1,31$ [s]
175	325	2	9,31 [s]	± 993 [ms]
200	300	2	10 [s]	± 858 [ms]
225	275	3	11,7 [s]	$\pm 1,73$ [ms]

Cuadro 4.2.3: Tabla tiempos de ejecución para $N = 500$.

n	m	l	Media tiempo de ejecución	Desviación estándar
75	625	9	41,1 [s]	$\pm 2,27$ [s]
113	637	6	39,2 [s]	$\pm 1,73$ [s]
150	600	4	39,9 [s]	$\pm 4,31$ [m]
188	562	3	40,4 [s]	$\pm 2,84$ [s]
225	525	3	36 [s]	$\pm 6,64$ [s]
263	487	2	30,8 [s]	$\pm 2,05$ [s]
300	450	2	31 [s]	$\pm 3,68$ [s]
338	412	2	29,7 [s]	$\pm 1,07$ [s]

Cuadro 4.2.4: Tabla tiempos de ejecución para $N = 750$.

n	m	l	Media tiempo de ejecución	Desviación estándar
100	900	9	84 [s]	± 13 [s]
150	850	6	68 [s]	$\pm 1,16$ [s]
200	800	4	68 [s]	$\pm 2,11$ [ms]
250	750	3	68 [s]	± 393 [ms]
300	700	3	68 [s]	$\pm 1,02$ [s]
350	650	2	78 [s]	$\pm 7,72$ [s]
400	700	2	75 [s]	$\pm 1,57$ [s]
450	550	2	98 [s]	$\pm 15,6$ [s]

Cuadro 4.2.5: Tabla tiempos de ejecución para $N = 1000$.

Capítulo 5

Grafos auxiliares separadores

En la sección anterior logramos definir un algoritmo determinista que permite encontrar sistemas separadores fuertes para el grafo bipartito completo $K_{n,m}$. Dado que deseamos definir otros algoritmos que permitan realizar esta tarea, procederemos a definir nuevos grafos y digrafos auxiliares que permitan diseñar nuevos algoritmos que encuentren sistemas separadores.

5.1. Grafo auxiliar para sistemas separadores débiles

Dado un conjunto S y una familia de subconjuntos \mathcal{F} de S definimos el grafo auxiliar $G_{\mathcal{F}} = (V, E)$ asociado a \mathcal{F} , donde $V = S$ y para cada par de elementos distintos a, b , se tiene que $ab \in E(G)$, si y solo si, el par a, b no está separado débilmente por \mathcal{F} .

En muchos casos supondremos que los conjuntos $X \in \mathcal{F}$ son tales que $X \neq S$ y $X \neq \emptyset$, debido a que estos conjuntos no proporcionan mucha información en el contexto de los sistemas separadores. Dado un conjunto S , consideremos un subconjunto X de S , tal que $X \subset S$ y $X \neq \emptyset$. Este conjunto X siempre separa débilmente a todo par de elementos a, b , con $a \in X$ y $b \in S \setminus X$. En base a esto y a la definición del grafo auxiliar podemos enunciar el siguiente lema.

Lema 5.1.1. *Sea S un conjunto tal que $|S| = n$ y \mathcal{F} una familia de subconjuntos de S tal que para todo $X \in \mathcal{F}$, $X \subset S$, y $X \neq \emptyset$. Entonces $\mathcal{F} = \emptyset$, si y solo si,*

$$G_{\mathcal{F}} = K_n.$$

Demostración. Por definición $G_{\mathcal{F}} = K_n$, si y solo si, todo par de elementos a, b distintos de S no están separados débilmente por \mathcal{F} . Dado que todo subconjunto $X \subset S$, tal que $X \neq \emptyset$ separa débilmente al menos a un par de elementos en S , se deduce que $G_{\mathcal{F}} = K_n$, si y solo si, $\mathcal{F} = \emptyset$. \square

El siguiente lema nos permite establecer mejor cómo se relacionan los elementos de \mathcal{F} con su grafo auxiliar asociado.

Lema 5.1.2. *Sea S un conjunto, sea X tal que $X \subset S$, y $X \neq \emptyset$ entonces $E(G_{\{X\}}) = E(G_{\{X, X^c\}}) = \{\{a, b\} \in [V]^2 : a \neq b \in X \text{ ó } a \neq b \in X^c\}$.*

Demostración. Sean las familias $\{X\}$ o $\{X, X^c\}$, en cualquiera de los dos casos podemos notar que solo los pares a, b tal que $a \in X$ y $b \in X^c$ se encuentran separados débilmente por \mathcal{F} . Dado que ambas familias separan débilmente a los mismos elementos, estas familias poseen el mismo grafo auxiliar. Dado que estos pares son los únicos separados débilmente, se concluye por la definición del grafo auxiliar que $E(G_{\{X\}}) = E(G_{\{X, X^c\}}) = \{\{a, b\} \in [V]^2 : a \neq b \in X \text{ ó } a \neq b \in X^c\}$. \square

Notemos además que podemos describir $G_{\{X\}}$ con más detalles. El grafo auxiliar asociado a las familias $\{X\}$ y $\{X, X^c\}$ se encuentra particionado por dos grafos completos $K_{|X|}$ y $K_{|X^c|}$ (Observe la Figura 5.1.1), estos grafos están formados por los vértices asociados a los elementos de X y X^c respectivamente y no comparten aristas entre si, esto es debido a que todos los pares de elementos en X o en X^c no se encuentran débilmente separados por \mathcal{F} , formando así todas las aristas posibles entre los vértices asociados tanto a X como a X^c .

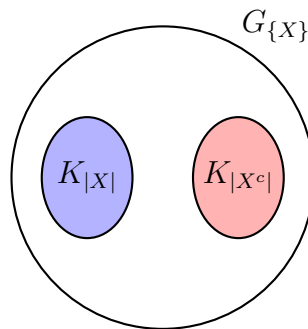


Figura 5.1.1: Ejemplo grafo auxiliar $G_{\{X\}}$.

Dada una familia \mathcal{F} , el siguiente lema nos describe una importante dinámica del grafo auxiliar $G_{\mathcal{F}}$ con los elementos de \mathcal{F} .

Lema 5.1.3. *Sea S un conjunto y \mathcal{F} una familia de subconjuntos de S , tal que para todo $X \in \mathcal{F}$, $X \subset S$, y $X \neq \emptyset$. Entonces $G_{\mathcal{F}} = \bigcap_{X \in \mathcal{F}} G_{\{X\}}$.*

Demostración. Dado que los conjuntos de vértices de $G_{\mathcal{F}}$ y $\bigcap_{X \in \mathcal{F}} G_{\{X\}}$ son los mismos por definición del grafo auxiliar, hemos de demostrar que los aristas de ambos grafos son exactamente las mismas. Sea $e = ab$ una arista de $G_{\mathcal{F}}$, dado que $e \in E(G_{\mathcal{F}})$, si y solo si, \mathcal{F} no separa débilmente a de b , entonces $e \in E(G_{\mathcal{F}})$, si y solo si, $\{X\}$ no separa débilmente a de b para cualquier $X \in \mathcal{F}$, de donde se concluye que $e \in E(G_{\{X\}})$ para todo $X \in \mathcal{F}$. \square

Dada la definición de estos grafos auxiliares podemos establecer una conexión directa entre los sistemas separadores débiles y los grafos auxiliares que hemos definido.

Lema 5.1.4. *Sea S un conjunto y \mathcal{F} una familia de subconjuntos de S . Entonces \mathcal{F} es un sistema separador débil de S , si y solo, $E(G_{\mathcal{F}}) = \emptyset$.*

Demostración. Dada la definición de $G_{\mathcal{F}}$, $E(G_{\mathcal{F}}) = \emptyset$, si y solo si, todo par de elementos distintos $a, b \in S$ están débilmente separados por \mathcal{F} , y por lo tanto, $E(G_{\mathcal{F}}) = \emptyset$, si y solo si, \mathcal{F} es un sistema separador débil de S . \square

A partir de estos grafos auxiliares podemos pensar en una estrategia para formar sistemas separadores débiles de S . Dada una familia \mathcal{F} de subconjuntos de S , seleccionamos un nuevo subconjunto X de S , luego la familia $\hat{\mathcal{F}} = \{X\} \cup \mathcal{F}$ define un nuevo grafo auxiliar $G_{\hat{\mathcal{F}}}$. Diremos que X borra aristas en el grafo auxiliar $G_{\mathcal{F}}$, si $|E(G_{\hat{\mathcal{F}}})| = |E(G_{\mathcal{F}}) \cap E(G_{\{X\}})| < |E(G_{\mathcal{F}})|$. Si consideremos que hemos seleccionado la familia $\mathcal{F} = \{X_1, \dots, X_n\}$ de subconjuntos de S , tal que $E(G_{\mathcal{F}}) = \emptyset$, entonces por el Lema 5.1.4 hemos encontrado un sistema separador débil para el conjunto S .

5.2. Digrafo auxiliar para sistemas separadores fuertes

Tal y como se mostró en la sección anterior, podemos pensar en definir una estrategia similar pero para obtener sistemas separadores fuertes por medio de algún grafo auxiliar. Es importante notar que un subconjunto X de S , no separa fuertemente a todo par de elementos a, b , con $a \in X$ y $b \in S \setminus X$, debido a esto, es conveniente definir la siguiente noción de separación.

Definición 5.2.1. Sea S un conjunto y \mathcal{F} una familia de subconjuntos de S , para cada par de elementos distintos $a, b \in S$ diremos que \mathcal{F} *semi-separa* a a de b , si existe un conjunto $X \in \mathcal{F}$, tal que $a \in X$ y $b \notin X$.

Esta definición es útil, pues dado un conjunto S y un subconjunto X tal que $X \subset S$, y $X \neq \emptyset$, se tiene que X siempre semi-separa a todos los elementos de X de los elementos de X^c .

Lema 5.2.2. *Sea S un conjunto y X un subconjunto de S , tal que $X \subset S$ y $X \neq \emptyset$. Entonces X siempre semi-separa a todos los elementos de X de los elementos de X^c .*

El Lema 5.2.3 nos permite relacionar directamente la noción de semi-separación con la noción de separación fuerte (Recuerde la definición de separación fuerte 2.0.1).

Lema 5.2.3. *Sean S un conjunto y \mathcal{F} una familia de subconjuntos de S . Sean además a, b un par de elementos distintos de S . Entonces, \mathcal{F} semi-separa a a de b y también semi-separa b de a , si y solo si, \mathcal{F} separa fuertemente al par a, b .*

Dado que la definición de semi-separación posee cierta noción de dirección, trasladaremos nuestras ideas al contexto de los digrafos. Dado un conjunto S y una familia \mathcal{F} de subconjuntos de S , definimos el dígrafo $D_{\mathcal{F}} = (V, E)$ asociado a \mathcal{F} , donde $V = S$ y el arco $(a, b) \in E$, si y solo si, a no esta semi-separado de b .

Al igual que antes, en general consideramos conjuntos $X \in \mathcal{F}$ tales que $X \neq \emptyset$ y $X \neq S$, dado que estos conjuntos tampoco son interesantes para el estudio de los sistemas separadores fuertes. Así, dada la definición del dígrafo auxiliar podemos enunciar el siguiente lema.

Lema 5.2.4. *Sea S un conjunto tal que $|S| = n$ y \mathcal{F} una familia de subconjuntos de S tal que para todo $X \in \mathcal{F}$, $X \subset S$ y $X \neq \emptyset$. Entonces, $\mathcal{F} = \emptyset$, si y solo si, $D_{\mathcal{F}}$ es el dígrafo completo en n vértices.*

Demostración. Por definición, $D_{\mathcal{F}}$ es igual al dígrafo completo en n vértices, si y solo si, todo par de elementos distintos a, b de S no están semi-separados por \mathcal{F} . Dado que todo subconjunto X distinto del vacío y de S semi-separa al menos a un par de elementos en S , se concluye que $\mathcal{F} = \emptyset$. \square

Para las familias $\{X\}$ y $\{X, X^c\}$ deseamos entender como son sus respectivos dígrafos auxiliares. Para ello, los siguientes dos lemas marcan una gran diferencia para estas dos familias entre los grafos auxiliares $G_{\mathcal{F}}$ y los dígrafos $D_{\mathcal{F}}$.

Lema 5.2.5. *Sea S un conjunto y sea $X \subset S$ tal que $X \neq \emptyset$. Entonces $E(D_{\{X\}}) = \{(a, b) \notin X \times X^c\}$, y por lo tanto*

$$E(D_{\{X\}}) = |S|^2 + |X|^2 - |S|(|X| + 1).$$

Demostración. Notemos que para la familia $\{X\}$, solo los pares ordenados (a, b) tal que $a \in X$ y $b \in X^c$ se encuentran semi-separados por \mathcal{F} , de forma que para todo $a \in X$ y $b \in X^c$, los arcos $(a, b) \notin E(D_{\{X\}})$. Dado que estos pares son los únicos semi-separados, se concluye por la definición del dígrafo auxiliar que

$$E(D_{\{X\}}) = \{(a, b) \notin X \times X^c\},$$

o equivalentemente

$$E(D_{\{X\}}) = \{(a, b) \in V \times V : a \neq b \in X \text{ ó } a \neq b \in X^c \text{ ó } (a, b) \in X^c \times X\}.$$

Ahora bien, dado que existen $2\binom{|X|}{2}$ pares (a, b) tal que $a \neq b \in X$, $2\binom{|S|-|X|}{2}$ pares

(a, b) tal que $a \neq b \in X^c$ y $(|S| - |X|)|X|$ pares $(a, b) \in X^c \times X$ se concluye que :

$$\begin{aligned} E(D_{\{X\}}) &= 2 \binom{|X|}{2} + 2 \binom{|S| - |X|}{2} + (|S| - |X|)|X| \\ &= 2 \frac{(|X||X-1|)}{2} + 2 \frac{(|S| - |X|)(|S| - |X| - 1)}{2} - (|S| - |X|)|X| \\ &= |X|^2 - |X| + (|S| - |X|)(|S| - 1) \\ &= |S|^2 + |X|^2 - |S|(|X| + 1), \end{aligned}$$

concluyendo así la demostración. \square

El grafo auxiliar asociado a la familia $\{X\}$ esta compuesto por dos digrafos completos $K_{|X|}$ y $K_{|X^c|}$ que se encuentran conectados por arcos desde los vértices de X^c hacia los vértices de X , esto es debido a que los pares ordenados $(a, b) \in X^c \times X$ no se encuentran semi-separados por la familia $\{X\}$. Los digrafos $K_{|X|}$ y $K_{|X^c|}$ están formados por los vértices asociados a los elementos en X y X^c respectivamente (Observe la Figura 5.2.1), esto es debido a que todos los pares ordenados de elementos en X o en X^c no se encuentran semi-separados por $\{X\}$, formando así todas los arcos posibles entre los vértices asociados tanto a X como a X^c .

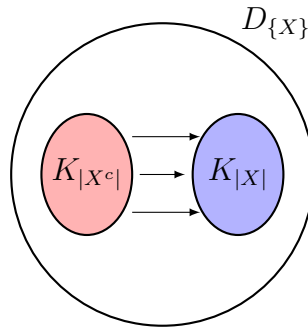


Figura 5.2.1: Ejemplo digrafo auxiliar $D_{\{X\}}$.

Lema 5.2.6. *Sea S un conjunto, sea $X \subset S$ tal que $X \neq \emptyset$. Entonces $E(D_{\{X, X^c\}}) = \{(a, b) \in V \times V : a \neq b \in X \text{ ó } a \neq b \in X^c\}$.*

Demostración. Notemos que para la familia $\{X, X^c\}$ los únicos pares semi-separados, son los pares $(a, b) \in X \times X^c$ y los pares $(a, b) \in X^c \times X$. Dado que estos pares son los únicos semi-separados se tiene que $(a, b) \notin E(D_{\{X, X^c\}})$, si y solo si, $(a, b) \in X \times X^c$ o $(a, b) \in X^c \times X$. De esta forma, por la definición del

dígrafo auxiliar se tiene que

$$E(D_{\{X, X^c\}}) = \{(a, b) \in V \times V : a \neq b \in X \text{ ó } a \neq b \in X^c\},$$

demostrando así lo deseado. \square

El siguiente lema es equivalente al Lema 5.1.3 para el caso de los digrafos auxiliares.

Lema 5.2.7. *Sea S un conjunto y \mathcal{F} una familia de subconjuntos de S , tal que todo $X \in \mathcal{F}$, $X \subset S$ y $X \neq \emptyset$. Entonces $D_{\mathcal{F}} = \bigcap_{X \in \mathcal{F}} D_{\{X\}}$.*

Demostración. Análogamente a la demostración del Lema 5.1.3. Sea $e = (a, b)$ un arco en $D_{\mathcal{F}}$. Dado que $e \in E(D_{\mathcal{F}})$, si y solo si, \mathcal{F} no semi-separa a de b , se tiene que $e \in E(D_{\mathcal{F}})$, si y solo si, $\{X\}$ no semi-separa a de b para cualquier $X \in \mathcal{F}$, de donde se concluye que $e \in E(D_{\{X\}})$ para todo $X \in \mathcal{F}$. \square

Para la noción de semi-separación detallaremos aun más a los elementos semi-separados entre si:

Lema 5.2.8. *Sea S un conjunto, sea $X \neq \emptyset$ un subconjunto de S , y sea \mathcal{F} una familia de subconjuntos de S no vacía. Entonces para cada $a \in X$ se tiene que*

$$N_{D_{\mathcal{F} \cup \{X\}}}^+(a) = N_{D_{\mathcal{F}}}^+(a) \cap X. \quad (5.2.1)$$

Demostración. Para cada $a \in X$ demostraremos la igualdad por medio de doble inclusión. Si $b \in N_{D_{\mathcal{F} \cup \{X\}}}^+(a)$ entonces $(a, b) \in E(D_{\mathcal{F} \cup \{X\}})$. Dado que $(a, b) \in E(D_{\mathcal{F} \cup \{X\}})$, en particular a no está semi-separado de b por \mathcal{F} , y por lo tanto, $b \in N_{\mathcal{F}}^+(a)$. Además, dado que $a \in X$ y a no está semi-separado de b se deduce que $b \in X$. Por otro lado, si $b \in N_{D_{\mathcal{F}}}^+(a)$ y $b \in X$ tenemos que \mathcal{F} no semi-separa a de b y X no semi-separa a de b puesto que $a, b \in X$, de esta manera $\mathcal{F} \cup \{X\}$ no semi-separa a a de b y se concluye que $(a, b) \in E(D_{\mathcal{F} \cup \{X\}})$, y por lo tanto $b \in N_{D_{\mathcal{F} \cup \{X\}}}^+(a)$. \square

Para cada conjunto $X \in S$ el Lema 5.2 nos permite conocer para cada elemento $a \in X$, que elementos en S no se encuentran semi-separados de a por la familia $\mathcal{F} \cup \{X\}$.

Dada la definición del digrafo auxiliar se tiene una relación directa entre los sistemas separadores fuertes y los digrafos auxiliares por medio del siguiente lema.

Lema 5.2.9. *Sea S un conjunto y \mathcal{F} una familia de subconjuntos de S . Entonces \mathcal{F} es un sistema separador fuerte de S , si y solo si, $E(D_{\mathcal{F}}) = \emptyset$.*

Demostración. Dado que si \mathcal{F} es tal que $E(D_{\mathcal{F}}) = \emptyset$, entonces por la definición de $D_{\mathcal{F}}$ todo par de elementos de S se encuentra semi-separado por \mathcal{F} . Luego por el Lema 5.2.3, todo par de elementos de S se encuentra fuertemente separado, y por lo tanto, \mathcal{F} es un sistema separador fuerte de S . \square

Análogamente a lo realizado en la sección anterior, dada una familia \mathcal{F} de subconjuntos de S podemos seleccionar un subconjunto X de S . La familia $\hat{\mathcal{F}} = \{X\} \cup \mathcal{F}$ define el nuevo dígrafo auxiliar $D_{\hat{\mathcal{F}}}$. Diremos que X borra arcos en el digrafo auxiliar $D_{\mathcal{F}}$ si $|E(D_{\hat{\mathcal{F}}})| = |E(D_{\mathcal{F}}) \cap E(D_{\{X\}})| < |E(D_{\mathcal{F}})|$.

Si repetimos el proceso de seleccionar subconjuntos de S y formamos una familia de subconjuntos \mathcal{F} que borre todas los arcos del digrafo auxiliar, es decir $E(D_{\mathcal{F}}) = \emptyset$, podemos asegurar por medio del Lema 5.2.9, que la familia \mathcal{F} es un sistema separador fuerte de S .

5.3. Algoritmos secuenciales

Por medio de las estrategias mencionadas en la sección anterior podemos definir algoritmos secuenciales 2 y 3 que permitan encontrar sistemas separadores débiles y fuertes de un conjunto S .

Algoritmo 2

Input: Conjunto S

Output: Un sistema separador débil \mathcal{F} de S .

- 1: $\mathcal{F} = \emptyset$
 - 2: **while** $E(G_{\mathcal{F}}) \neq \emptyset$ **do**
 - 3: Seleccionar un subconjunto X de S
 - 4: **Update** $\mathcal{F} = \mathcal{F} \cup \{X\}$
 - 5: **return** \mathcal{F}
-

Algoritmo 3

Input: Conjunto S **Output:** Un sistema separador fuerte \mathcal{F} de S .

- 1: $\mathcal{F} = \emptyset$
 - 2: **while** $E(D_{\mathcal{F}}) \neq \emptyset$ **do**
 - 3: Seleccionar un subconjunto X de S
 - 4: **Update** $\mathcal{F} = \mathcal{F} \cup \{X\}$
 - 5: **return** \mathcal{F}
-

Además, podemos usar la estructura de estos algoritmos para definir también un algoritmo secuencial 4 que encuentre sistemas separadores fuertes (o débiles sustituyendo $D_{\mathcal{P}}$ por $G_{\mathcal{P}}$) de caminos para un grafo G dado.

Algoritmo 4

Input: Grafo $G = (V, E)$ **Output:** Un sistema separador fuerte de caminos \mathcal{F} de G .

- 1: $\mathcal{P} = \emptyset$
 - 2: **while** $E(D_{\mathcal{P}}) \neq \emptyset$ **do**
 - 3: Seleccionar un camino P del grafo G
 - 4: **Update** $\mathcal{P} = \mathcal{P} \cup \{E(P)\}$
 - 5: **return** \mathcal{P}
-

Existen múltiples formas para seleccionar un camino en un grafo que pueden ser implementadas en el Algoritmo 4, en este trabajo estudiaremos que ocurre cuando estos caminos se seleccionan de manera aleatoria. Nos enfocaremos en esta manera de seleccionarlos, ya que para el grafo $K_{n,m}$ obtenemos buenos resultados teóricos (Capítulo 6) y una sencilla implementación computacional que permite corroborar los resultados teóricos por medio de resultados experimentales (Capítulo 7).

Al momento de diseñar el algoritmo, si los caminos son seleccionados de manera aleatoria podemos esperar que el desempeño del algoritmo comparta similitudes con el problema del coleccionista de cupones.

El problema del coleccionista consiste en intentar completar una colección de cupones. Para lograr esta tarea se compran sobres que contienen un cupón cada uno, donde la probabilidad de que un cupón este en un sobre es uniforme para todos los cupones. En un principio, cuando compremos sobres obtendremos en esperanza muchos nuevos cupones, pero al cabo de un tiempo, al comprar muchos sobres empezaremos cada vez más a repetir cupones dificultado así nuestro objetivo. El

problema del coleccionista de cupones ha sido ampliamente estudiado y presenta diversas variaciones, donde esta tarea puede ser aun más difícil al considerar sobres que contengan más de un cupón o cupones especiales con distintas probabilidades de aparecen en un sobre.

En nuestro caso, la colección de cupones corresponde $E(D_{\mathcal{P}})$, ya que simbolizan los cupones que deseamos coleccionar. Por otro lado, los sobres de cupones para este caso corresponden al conjunto de arcos que un camino P borra en $D_{\mathcal{P}}$ al agregarlo a \mathcal{P} . De esta manera, en cada iteración de nuestro algoritmo se selecciona un camino de manera aleatoria que simboliza la compra de un sobre, y en el momento en que hemos borrado todos los arcos del digrafo auxiliar hemos conseguido completar la colección de cupones.

Capítulo 6

Análisis del algoritmo aleatorio secuencial

Para iniciar nuestro análisis, hemos de definir un nuevo algoritmo basado en el Algoritmo 4, pero aplicado en el grafo $K_{n,m}$ y con caminos seleccionados de manera aleatoria. Para ello consideraremos de aquí en adelante que el grafo $K_{n,m}$ posee la partición de vértices $V = A \cup B$ con $|A| = n$, $|B| = m$ y $n < m$. Para seleccionar caminos aleatorios de $K_{n,m}$ enumeraremos los vértices de la parte $A = \{1, \dots, n\}$ y los vértices de la parte de $B = \{1, \dots, m\}$.

Dado un conjunto de X elementos, denotaremos S_X al conjunto de todas las permutaciones del conjunto X . Definimos además las permutaciones aleatorias $\sigma : A \rightarrow A$ y $\tau : B \rightarrow B$ de elementos de A y B respectivamente, donde denotaremos para todo $i \in A$, $\sigma(i) = \sigma_i$ y para todo $j \in B$, $\tau(j) = \tau_j$. De esta forma, si aplicamos las permutaciones aleatorias σ y τ en A y B respectivamente entonces un camino de largo $2k$ viene dado por la secuencia

$$\tau_1 \sigma_1 \tau_2 \dots \tau_k \sigma_k \tau_{k+1}.$$

Este camino lo denotaremos por $P^k(\sigma, \tau)$. Notemos además que dado que $n < m$, cuando $k = n$ el camino $P^n(\sigma, \tau)$ corresponde a un camino de largo máximo obtenido de manera aleatoria.

A partir de esta forma de seleccionar caminos aleatorios, definimos el siguiente algoritmo:

Algoritmo 5 Algoritmo aleatorio secuencial**Input:** Grafo bipartito completo $K_{n,m}$ **Output:** Un sistema separador fuerte \mathcal{P} de grafo $K_{n,m}$.

- 1: $\mathcal{P} = \emptyset$
- 2: **while** $E(D_{\mathcal{P}}) \neq \emptyset$ **do**
- 3: Seleccionar un camino $P = P^k(\sigma, \tau)$ en el grafo $K_{n,m}$
- 4: **Update** $\mathcal{P} = \mathcal{P} \cup \{E(P)\}$
- 5: **return** \mathcal{P}

Denotaremos D_t al dígrafo auxiliar asociado a \mathcal{P}_t , donde \mathcal{P}_t es el sistema separador fuerte de caminos generado por el Algoritmo 5 después de seleccionar t caminos.

6.1. Análisis teórico

En primer lugar notemos que dado un camino fijo $P = P^k(\sigma, \tau)$ con $k \in \{1, \dots, n\}$, por el Lema 5.2.2 $E(P)$ semi-separa a cada $e \in E(P)$ de todo $f \in E(P)^c$. A partir de esto podemos asegurar que la cantidad de arcos que borra cualquier camino aleatorio $P^k(\sigma, \tau)$ con $k \in \{1, \dots, n\}$ en la primera iteración es determinista.

Proposición 6.1.1. Sea $k \leq n < m$, sean D_0, D_1, \dots los dígrafos generados por el Algoritmo 5, entonces

$$|E(D_0) \setminus E(D_1)| = 2k(nm - 2k).$$

Es decir, la cantidad de arcos borrados en el dígrafo D_0 al agregar el primer camino de largo $2k$ generado por el Algoritmo 5 es igual $2k(nm - 2k)$.

Demostración. Dado que D_0 corresponde al dígrafo auxiliar cuando aún no se han seleccionado caminos, entonces el sistema separador fuerte de caminos generado por el Algoritmo 5 corresponde a $\mathcal{F} = \emptyset$. Luego, por el Lema 5.2.4 D_0 corresponde al dígrafo completo en $E(K_{n,m}) = mn$ vértices. Por otro lado, D_1 corresponde al dígrafo generado por el Algoritmo 5 cuando solo se ha agregado un camino P . Sabemos por el Lema 5.2.5 que la cantidad de arcos de este dígrafo D_1 corresponde a

$$\begin{aligned} E(D_1) &= (E(K_{n,m}))^2 + (2k)^2 - (E(K_{n,m}))(2k + 1) \\ &= m^2n^2 + 4k^2 - mn(2k + 1). \end{aligned}$$

Ahora bien, considerando el Lema 5.2.7 concluimos que la cantidad de arcos borrados en la primera iteración del Algoritmo 5 es:

$$\begin{aligned} |E(D_0) \setminus E(D_1)| &= |E(D_0)| - |E(D_1)| \\ &= mn(mn - 1) - m^2n^2 - 4k^2 + mn(2k + 1) \\ &= 2k(mn - 2k), \end{aligned}$$

demostrando así lo deseado. \square

Para un camino $P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$ con $k \in \{1, \dots, n\}$ deseamos estimar la probabilidad de que una arista fija $e \in K_{n,m}$ pertenezca al camino $P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$.

Lema 6.1.2. Sean $n < m \in \mathbb{N}$, $K_{n,m}$ el grafo bipartito completo, $\mathbf{P}^k = P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$ con $k \in \{1, \dots, n\}$ y e una arista de $K_{n,m}$, Entonces

$$\Pr[e \in E(\mathbf{P}^k)] = \frac{2k}{nm}.$$

Demostración. Consideremos que $e = xy$ y sin pérdida de generalidad que $x \in A$ y $y \in B$. Luego, condicionando sobre las posibles asignaciones de $\boldsymbol{\sigma}$ para x , obtenemos que

$$\Pr[xy \in E(\mathbf{P}^k)] = \sum_{i=1}^n \Pr[xy \in E(\mathbf{P}^k) | \sigma_i = x] \Pr[\sigma_i = x].$$

Efectuadas las permutaciones $\boldsymbol{\sigma}$ y $\boldsymbol{\tau}$ para obtener el camino $P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$, para cualquier $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, m\}$ una arista $\sigma_i \tau_j \in P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$, si y solo si, $i \leq k$ y $j \in \{i, i + 1\}$. De esta manera, $\Pr[xy \in E(\mathbf{P}^k) | \sigma_i = x]$ para todo $k < i \leq n$ es igual a 0. Por otro lado, para los σ_i con $i \leq k$ solo existen dos opciones favorables de asignaciones de $\boldsymbol{\tau}$ para el vértice y de las m posibles. A partir de esto

$$\Pr[xy \in E(\mathbf{P}^k) | \sigma_i = x] = \frac{2}{m}.$$

Dado que $\Pr[\sigma_i = x]$ es equivalente a seleccionar un elemento de manera aleatoria

de un conjunto de tamaño n , se tiene que $\Pr[\sigma_i = x] = \frac{1}{n}$, y por lo tanto

$$\Pr[xy \in E(\mathbf{P}^k)] = \sum_{i=1}^k \Pr[xy \in E(\mathbf{P}^k) | \sigma_i = x] \Pr[\sigma_i = x] = \frac{2k}{nm},$$

concluyendo así lo deseado. \square

Dado que hemos calculado la probabilidad de que una arista del grafo $K_{n,m}$ pertenezca a un camino $P^k(\sigma, \tau)$, podemos preguntarnos cuantas iteraciones son necesarias para cubrir todas las aristas del grafo $K_{n,m}$. Para ello estimaremos la cantidad esperada de aristas no cubiertas por los caminos del Algoritmo 5. Definimos para cada $t \geq 1$ la variable aleatoria \mathbf{N}_t como la cantidad de aristas no cubiertas por los caminos de la familia $\mathcal{P}_t = \{E(\mathbf{P}_1^k), \dots, E(\mathbf{P}_t^k)\}$. Definimos además \mathbf{N}_0 como la cantidad de aristas del digrafo cuando aún no se ha seleccionado ningún camino. Luego, la siguiente proposición estima la cantidad esperada de aristas no cubiertas en el grafo $K_{n,m}$ por el Algoritmo 5 en t iteraciones.

Proposición 6.1.3. Sea $k \leq n < m$, entonces

$$\mathbb{E}(\mathbf{N}_t) = \left(1 - \frac{2k}{nm}\right)^t mn, \quad (6.1.1)$$

para todo $t \geq 0$.

Demostración. En primer lugar notemos que para $t = 0$ el Algoritmo 5 no ha seleccionado ningún camino \mathbf{P}^k , por lo tanto, la cantidad de aristas no cubiertas por algoritmo es fija y viene dada por

$$\mathbb{E}(\mathbf{N}_0) = mn = \left(1 - \frac{2k}{nm}\right)^0 mn.$$

Razonaremos por inducción, supongamos que la Ecuación (6.1.1) se cumple para algún $i \in \mathbb{N}$, luego para $i + 1$ sea $X := E(\mathbf{P}_{i+1}^k) \setminus \bigcup_{1 \leq j \leq i} E(\mathbf{P}_j^k)$. Dado que X corresponde al conjunto de aristas cubiertas por el camino \mathbf{P}_{i+1}^k y que no son cubiertas por la familia \mathcal{P}_i se tiene que

$$\mathbf{N}_{i+1} = \mathbf{N}_i - |X|.$$

Por otro lado, considerando el Lema 6.1.2 se tiene que

$$\mathbb{E}(|X|) = \sum_{e \in \mathbf{N}_i} \Pr(e \in \mathbf{P}_{i+1}^k) = \frac{2k}{nm} \mathbb{E}(\mathbf{N}_i),$$

de donde obtenemos que

$$\begin{aligned} \mathbb{E}(\mathbf{N}_{i+1}) &= \mathbb{E}(\mathbf{N}_i) - \mathbb{E}(|X|) \\ &= \mathbb{E}(\mathbf{N}_i) - \frac{2k}{nm} \mathbb{E}(\mathbf{N}_i) \\ &= \left(1 - \frac{2k}{nm}\right) \mathbb{E}(\mathbf{N}_i). \end{aligned}$$

Luego, como $\mathbb{E}(\mathbf{N}_i) = \left(1 - \frac{2k}{nm}\right)^i nm$ se tiene que

$$\mathbb{E}(\mathbf{N}_{i+1}) = \left(1 - \frac{2k}{nm}\right)^{i+1} nm,$$

obteniendo así lo deseado. \square

Si bien hemos modelado la cantidad de esperada de aristas no cubiertas por el algoritmo, debemos asegurar que la variable \mathbf{N}_t es parecida a su esperanza $\mathbb{E}(\mathbf{N}_t)$. Para ello usaremos la siguiente desigualdad de concentración para permutaciones del artículo [Mau79].

Proposición 6.1.4. Sea $f : S_N \rightarrow \mathbb{R}$ una función sobre permutaciones, tal que si $\pi, \pi' \in S_N$ difieren en una trasposición, entonces $|f(\pi) - f(\pi')| \leq z$. Entonces para un $\pi \in S_n$ escogido al azar uniformemente se tiene que

$$\Pr[|f(\pi) - \mathbb{E}(f)| \geq cz] \leq 2e^{-c^2/(4n)}.$$

Para poder utilizar la Proposición 6.1.4 definiremos una manera alternativa de generar caminos de largo $2k$ en el grafo $K_{n,m}$ de manera aleatoria. Sea la permutación $\boldsymbol{\mu} : V(K_{n,m}) \rightarrow V(K_{n,m})$, donde denotaremos para cada $i \in V(K_{n,m})$, $\boldsymbol{\mu}(i) = \mu_i$. Definiremos los caminos $P^k(\boldsymbol{\mu})$ de la siguiente manera. Seleccionamos a los primeros k elementos de A , en el orden de $\boldsymbol{\mu}$ y denotamos a este orden de elementos $\boldsymbol{\sigma}$. Luego, seleccionamos a los primeros $k+1$ elementos de

B en el orden de $\boldsymbol{\mu}$ y los denotamos $\boldsymbol{\tau}$. A partir de esto

$$P^k(\boldsymbol{\mu}) := P^k(\boldsymbol{\sigma}, \boldsymbol{\tau}).$$

Considere el siguiente ejemplo de un camino $P^k(\boldsymbol{\mu})$ con $k = 2$. Sea $K_{3,4}$, donde $A = \{a_1, a_2, a_3\}$ y $B = \{b_1, b_2, b_3, b_4\}$. Sea $\boldsymbol{\mu} = \{b_1, b_3, a_2, b_2, a_1, a_3, b_4\}$, entonces para el camino $P^2(\boldsymbol{\mu})$, se tiene que $\boldsymbol{\sigma} = \{a_2, a_1, a_3\}$, $\boldsymbol{\tau} = \{b_1, b_3, b_2, b_4\}$, y por lo tanto

$$P^2(\boldsymbol{\mu}) = P^2(\boldsymbol{\sigma}, \boldsymbol{\tau}) = b_1 a_2 b_3 a_1 b_2.$$

De esta manera generando permutaciones $\boldsymbol{\mu}$ de todo el conjunto de vértices podemos generar las permutaciones $\boldsymbol{\sigma}$ y $\boldsymbol{\tau}$ que nos permiten obtener los caminos $P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$.

Definiremos ahora para cada $t \in \mathbb{N}$ las funciones $f^t : S_N \rightarrow \mathbb{N}$, donde $f^t(\boldsymbol{\mu})$ corresponde a la cantidad de aristas que restan por cubrir en el grafo $K_{n,m}$ en la iteración t del Algoritmo 5 al agregar el camino $P^k(\boldsymbol{\mu})$. Notemos que $E(f^t) = E(\mathbf{N}_t)$, puesto que \mathbf{N}_t corresponde a la variable aleatoria que modela la cantidad de aristas no cubiertas por los caminos de la familia $\mathcal{P}_t = \{E(\mathbf{P}_1^k), \dots, E(\mathbf{P}_t^k)\}$.

Ahora bien, consideremos dos permutaciones fijas pero arbitrarias $\boldsymbol{\mu}$ y $\boldsymbol{\mu}'$ que difieren en una trasposición. Notemos que para los caminos $P^k(\boldsymbol{\mu})$ y $P^k(\boldsymbol{\mu}')$, si las posiciones que difieren $\boldsymbol{\mu}$ y $\boldsymbol{\mu}'$ solo alteran a la permutación $\boldsymbol{\sigma}$, entonces estos caminos difieren en a lo más 4 aristas, análogamente si solo la permutación $\boldsymbol{\tau}$ se ve alterada también estos caminos difieren en a lo más 4 aristas. En cambio, si las posiciones que difieren $\boldsymbol{\mu}$ y $\boldsymbol{\mu}'$ alteran tanto $\boldsymbol{\sigma}$ como a $\boldsymbol{\tau}$, entonces los caminos pueden diferir en a lo más 8 aristas. De esta manera podemos concluir que para dos permutaciones $\boldsymbol{\mu}$ y $\boldsymbol{\mu}'$ que difieren en una trasposición, para todo $t \in \mathbb{N}$ se tiene que

$$|f^t(\boldsymbol{\mu}) - f^t(\boldsymbol{\mu}')| \leq 8.$$

En base a esta información y la Proposición 6.1.4, podemos asegurar que para una permutación $\boldsymbol{\mu} \in S_N$ escogida de manera aleatoria, se cumple que

$$\Pr[|f^t(\boldsymbol{\mu}) - E(f^t)| \geq 8c] \leq 2e^{-c^2/(4N)} \text{ para todo } t \in \mathbb{N},$$

o equivalentemente

$$\Pr[|f^t(\boldsymbol{\mu}) - E(\mathbf{N}_t)| \geq 8c] \leq 2e^{-c^2/(4N)} \text{ para todo } t \in \mathbb{N}. \quad (6.1.2)$$

Ahora bien, dado que buscamos que la probabilidad de que $f^t(\boldsymbol{\mu})$ y $E(\mathbf{N}_t)$ difieran sea pequeña lo largo de las iteraciones, entonces debemos imponer en la ecuación (6.2.12) que $e^{-c^2/(4N)} \ll 0$, y por lo tanto

$$c \gg 2\sqrt{N} = 2\sqrt{n+m}. \quad (6.1.3)$$

Como se menciono anteriormente, deseamos estudiar para qué iteraciones t se tiene que la diferencia $|f^t(\boldsymbol{\mu}) - E(\mathbf{N}_t)|$ es pequeña. Para nuestro análisis, esto puede ser interpretado como $E(\mathbf{N}_t) \gg 8c$, o equivalentemente

$$c \ll \frac{(1 - \frac{2k}{nm})^t}{8mn}. \quad (6.1.4)$$

Combinando las ecuaciones (6.1.3) y (6.1.4), obtenemos que

$$2\sqrt{n+m} \ll \frac{(1 - \frac{2k}{nm})^t}{8mn},$$

de donde deducimos que esto ocurre a partir de las iteraciones $t \in \mathbb{N}$ tales que

$$t \ll \frac{mn}{2k} \ln \left(\frac{mn}{16\sqrt{m+n}} \right).$$

Dado que hemos verificado que la $E(\mathbf{N}_t)$ es representativa para \mathbf{N}_t , podemos estimar cuantas iteraciones esperamos sean necesarias para cubrir todas las aristas salvo una. Denotaremos t_c a la cantidad iteraciones necesarias para que reste menos de una arista por cubrir por los caminos de la familia generada por el Algoritmo 5. Dado que $E(\mathbf{N}_t) = (1 - \frac{2k}{nm})^t nm$, obtenemos que $t_c \geq \frac{mn}{2k} \ln(mn)$.

6.2. Probabilidad de borrar un arco (xy, uv) en $D_{\mathcal{F}}$

Dado un camino $\mathbf{P}^k = P^k(\sigma, \tau)$ deseamos estimar la probabilidad de que una arista $xy \in E(\mathbf{P}^k)$ y otra arista $uv \notin E(\mathbf{P}^k)$. Es fundamental entender este evento, puesto que si éste evento ocurre entonces el camino \mathbf{P}^k semi-separa a la arista xy de uv , y por lo tanto, borra el arco (xy, uv) en el digrafo auxiliar. Sin pérdida de generalidad consideramos que $x, u \in A$, y $y, v \in B$. Definiremos para cada $k \in \{1, \dots, n\}$ el siguiente evento aleatorio

$$\mathcal{E}^k(xy, uv) : xy \in E(\mathbf{P}^k) \text{ y } uv \notin E(\mathbf{P}^k).$$

Notemos que dado un camino fijo $P^k(\sigma, \tau)$ con $k \in \{1, \dots, n-1\}$, si $\sigma_i = x$ con $i > k$ entonces el evento aleatorio anterior no ocurre, ya que por definición del camino $P^k(\sigma, \tau)$ la arista $xy \notin P^k(\sigma, \tau)$. Además, dadas las permutaciones σ y τ fijas tal que $\sigma_i = x$ y $\sigma_j = u$ el evento $\mathcal{E}^k(xy, uv)$ para todo $k \in \{1, \dots, n\}$ ocurre, si y solo si, $y \in \{\tau_i, \tau_{i+1}\}$ y $v \notin \{\tau_j, \tau_{j+1}\}$.

Por otro lado, para un evento $\mathcal{E}^k(xy, uv)$ con $k \in \{1, \dots, n\}$ existen 3 casos distintos que debemos analizar de manera independiente. Estos casos corresponden a:

- (I) Las aristas xy, uv son disjuntas, es decir $\{xy\} \cap \{uv\} = \emptyset$.
- (II) Las aristas xy, uv comparten vértice en B , es decir $\{xy\} \cap \{uv\} = \{y\}$.
- (III) Las aristas xy, uv comparten vértice en A , es decir $\{xy\} \cap \{uv\} = \{x\}$.

La siguiente proposición nos asegura que la probabilidad de que el evento $\mathcal{E}^k(xy, uv)$ ocurra es máxima cuando se seleccionan caminos de largo máximo, es decir, cuando se seleccionan caminos $P^n(\sigma, \tau)$.

Proposición 6.2.1. Sean $K_{n,m}$ un grafo bipartito completo en N vértices con $n \geq 3$, $m \geq 4$ y $n < m$. Sean xy y uv aristas distintas de $K_{n,m}$. Entonces

$$\Pr[\mathcal{E}^{k-1}(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv)],$$

para todo $k \in \{2, \dots, n\}$.

Para demostrar la Proposición 6.2.1 primero notemos que para cualquier $xy, uv \in$

$E(K_{n,m})$ y $k \in \{2, \dots, n-1\}$ se tiene que $\Pr[(\mathcal{E}^{k-1}(xy, uv))]$ es igual a

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] + \Pr[\mathcal{E}^k(xy, uv)] - \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)].$$

La ecuación anterior puede ser intuitida por la siguiente figura

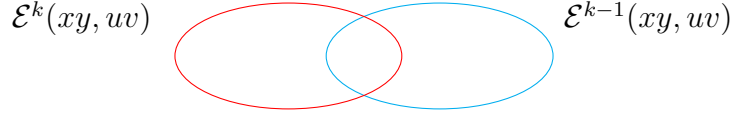


Figura 6.2.1: Intersección de los eventos $\mathcal{E}^{k-1}(xy, uv)$ y $\mathcal{E}^k(xy, uv)$.

De esta manera demostrar que $\Pr[\mathcal{E}^{k-1}(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv)]$, es equivalente a demostrar que

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)]. \quad (6.2.1)$$

Analicemos los eventos $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ y $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$. El evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ solo ocurre si se cumplen los siguientes dos puntos al mismo tiempo:

- $xy \in P^{k-1}(\sigma, \tau)$ y $uv \notin P^{k-1}(\sigma, \tau)$.
- $xy \notin P^k(\sigma, \tau)$ ó $uv \in P^k(\sigma, \tau)$.

Dado que $xy \in P^{k-1}(\sigma, \tau)$ es imposible que $xy \notin P^k(\sigma, \tau)$. Por otro lado, para que ocurra que $uv \notin P^{k-1}(\sigma, \tau)$ y además $uv \in P^k(\sigma, \tau)$, solo es posible si $uv \in \{\sigma_k \tau_k, \sigma_k \tau_{k+1}\}$. De forma que $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ solo ocurre si

$$xy \in P^k(\sigma, \tau) \quad y \quad uv \in \{\sigma_k \tau_k, \sigma_k \tau_{k+1}\}. \quad (6.2.2)$$

Ahora bien, análogamente el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$ solo ocurre si se cumple los siguientes dos puntos al mismo tiempo:

- $xy \in P^k(\sigma, \tau)$ y $uv \notin P^k(\sigma, \tau)$.
- $xy \notin P^{k-1}(\sigma, \tau)$ ó $uv \in P^{k-1}(\sigma, \tau)$.

Dado que $uv \notin P^k(\sigma, \tau)$ es imposible que $uv \in P^{k-1}(\sigma, \tau)$. Por otro lado, si

$xy \in P^k(\boldsymbol{\sigma}, \boldsymbol{\tau})$ y además $xy \notin P^{k-1}(\boldsymbol{\sigma}, \boldsymbol{\tau})$, entonces $xy \in \{\sigma_k\tau_k, \sigma_k\tau_{k+1}\}$. De forma que $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$ solo ocurre si

$$uv \notin P^k(\boldsymbol{\sigma}, \boldsymbol{\tau}) \quad \text{y} \quad xy \in \{\sigma_k\tau_k, \sigma_k\tau_{k+1}\}. \quad (6.2.3)$$

Teniendo en cuenta las condiciones (6.2.1), (6.2.2) y (6.2.3) procedemos a demostrar la Proposición 6.2.1.

Demostración. Dado que demostrar que $\Pr[\mathcal{E}^{k-1}(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv)]$, es equivalente a demostrar que

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)].$$

Procedemos a demostrar la inecuación anterior para cada uno de los 3 posibles casos para un par de aristas xy, uv fijas. Para demostrar los primeros dos casos, construiremos aplicaciones inyectivas que nos permitan asegurar que existen tanto casos favorables para ocurra el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$ como casos favorables del evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$.

(I) Las aristas xy, uv son disjuntas, es decir $\{xy\} \cap \{uv\} = \emptyset$.

Dado que el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ solo ocurre si

$$xy \in P^k(\boldsymbol{\sigma}, \boldsymbol{\tau}) \quad \text{y} \quad uv \in \{\sigma_k\tau_k, \sigma_k\tau_{k+1}\}.$$

Si consideramos que $x \neq u$ y $y \neq v$ más las condiciones para que ocurra el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, notamos que existen estos subcasos (Observe la figura 6.2.2).

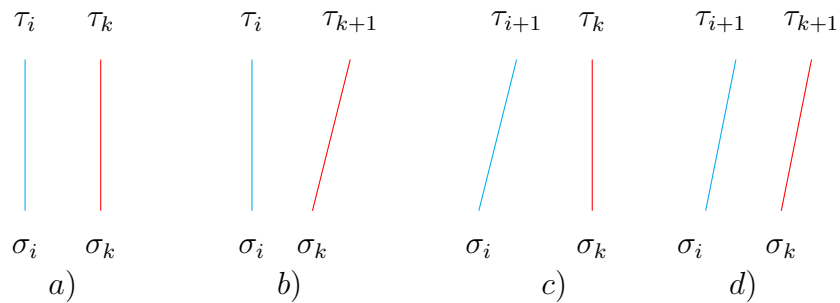


Figura 6.2.2: Ejemplos de los 4 subcasos para el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$.

- a) $xy = \sigma_i \tau_i$ para algún $i \in \{1, \dots, k-1\}$ y $uv = \sigma_k \tau_k$.
- b) $xy = \sigma_i \tau_i$ para algún $i \in \{1, \dots, k-1\}$ y $uv = \sigma_k \tau_{k+1}$.
- c) $xy = \sigma_i \tau_{i+1}$ para algún $i \in \{1, \dots, k-1\}$ y $uv = \sigma_k, \tau_k$.
- d) $xy = \sigma_i \tau_{i+1}$ para algún $i \in \{1, \dots, k-1\}$ y $uv = \sigma_k \tau_{k+1}$.

Para cada dos aristas fijas $xy, uv \in E(K_{n,m})$ y $l \in \{1, \dots, k-1\}$, definimos el siguiente conjunto de permutaciones

$$S^l(xy, uv) = \{(\sigma, \tau) \in S_A \times S_B : x = \sigma_l, y \in \{\tau_l, \tau_{l+1}\}\}.$$

Es importante notar que los pares de permutaciones (σ, τ) que generan el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, están contenidos en el conjunto $S^l(xy, uv)$.

Luego para cada $S^l(xy, uv)$, definimos la siguiente aplicación $T^l : S^l(xy, uv) \rightarrow S_A \times S_B$, donde $T^l(\tau, \sigma) = (\sigma^l, \tau^l)$ viene dada por

$$\sigma_i^l = \begin{cases} \sigma_l & \text{para } i = k, \\ \sigma_k & \text{para } i = l, \\ \sigma_i & \text{para todo } i \neq k, l. \end{cases}$$

y

$$\tau_i^l = \begin{cases} \tau_l & \text{para } i = k, \\ \tau_{l+1} & \text{para } i = k+1, \\ \tau_k & \text{para } i = l, \\ \tau_{k+1} & \text{para } i = l+1, \\ \tau_i & \text{para todo } i \neq k, k-1. \end{cases}$$

Para entender mejor la aplicación T^l observe la Figura 6.2.3. En la Figura 6.2.3 se presenta en la parte superior un par de permutaciones $(\sigma, \tau) \in S^l(xy, uv)$, donde se marcan con color azul las aristas $\sigma_l \tau_l$ y $\sigma_l \tau_{l+1}$, y con color rojo las aristas $\sigma_k \tau_k$ y $\sigma_k \tau_{k+1}$. Luego, en la parte inferior de la Figura 6.2.3 se presenta el par de permutaciones $(\sigma^l, \tau^l) = T^l(\sigma, \tau)$, donde se marcan con color azul las nuevas posiciones de las aristas $\sigma_l \tau_l$ y $\sigma_l \tau_{l+1}$, y con color rojo las nuevas posiciones de las aristas $\sigma_k \tau_k$ y $\sigma_k \tau_{k+1}$. Cabe recalcar que el resto de posiciones no se ve alterado.

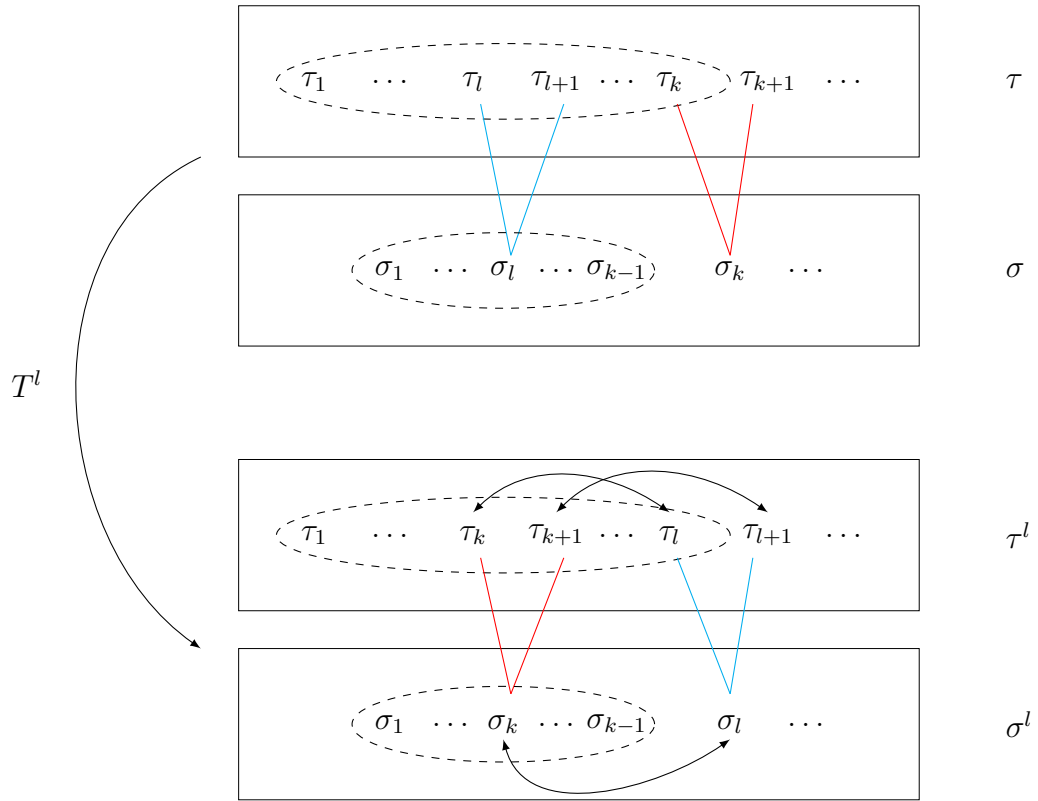


Figura 6.2.3: Par de permutaciones $(\sigma, \tau) \in S^l(xy, uv)$ antes y después de aplicar T^l .

Notemos que para cualquiera de los 4 subcasos, al aplicar T^l a un par de permutaciones (σ, τ) que genere el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, para el camino $P^k(T^l(\sigma, \tau)) = P^k(\sigma^l, \tau^l)$ se cumple que

$$xy \in P^k(\sigma^l, \tau^l) \setminus P^{k-1}(\sigma^l, \tau^l).$$

o equivalentemente

$$xy \in \{\sigma_k^l \tau_k^l, \sigma_k^l \tau_{k+1}^l\}, \tag{6.2.4}$$

y además se cumple que

$$uv \in P^{k-1}(\sigma^l, \tau^l). \tag{6.2.5}$$

Ahora bien, definimos también la aplicación $W : S_A \times S_B \rightarrow S_A \times S_B$, donde

$W(\tau, \sigma) = (\sigma^*, \tau^*)$ viene dada por

$$\sigma_i^* = \begin{cases} \sigma_{i+1 \pmod{k-1}} & \text{para todo } 1 \leq i < k-1, \\ \sigma_i & \text{para todo } k \leq i \leq n. \end{cases}$$

y

$$\tau_i^* = \begin{cases} \tau_{i-1 \pmod{k}} & \text{para todo } 1 \leq i < k-1, \\ \tau_i & \text{para todo } k \leq i \leq m. \end{cases}$$

Nuevamente, para entender mejor la aplicación W observe la Figura 6.2.4. La Figura 6.2.4 presenta en la parte superior un par de permutaciones (σ, τ) , donde con color azul se marca la arista $\sigma_i \tau_i$ y con color rojo la arista $\sigma_{k-1} \tau_k$ en el camino $P^{k-1}(\sigma, \tau)$. Luego, en la parte inferior de la Figura 6.2.4 se presenta el par de permutaciones $(\sigma^*, \tau^*) = W(\sigma, \tau)$, donde nuevamente se marcan con sus colores respectivos las nuevas posiciones de las aristas $\sigma_i \tau_i$ y $\sigma_{k-1} \tau_k$ con respecto a las nuevas permutaciones (σ^*, τ^*) . Cabe recalcar que el resto de posiciones fuera del camino $P^{k-1}(\sigma, \tau)$ no se ven alteradas.

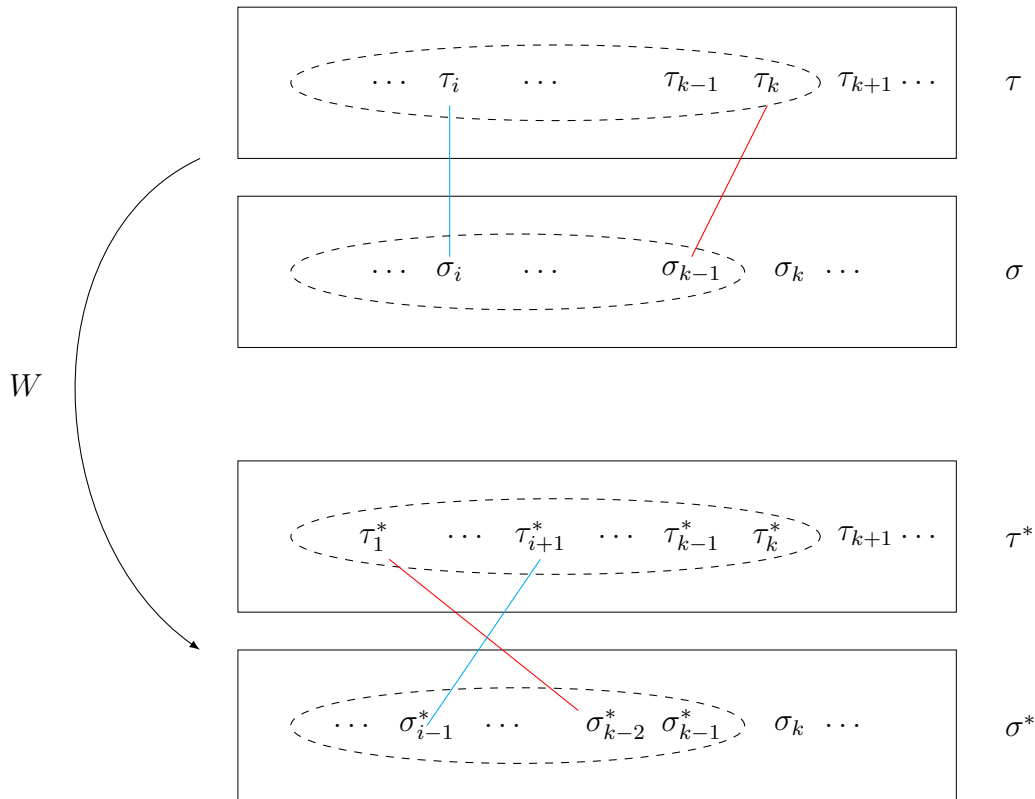


Figura 6.2.4: Par de permutaciones (σ, τ) antes y después de aplicar W .

Notemos que la aplicación W para un par de permutaciones fijas (σ, τ) traslada a todas las posiciones σ_i con $i \leq k-1$ en una posición hacia atrás de manera cíclica. Y además, la aplicación W traslada en una posición hacia adelante de manera cíclica a todas las posiciones τ_i con $i \leq k$. Por ultimo la aplicación W no altera todo el resto de posiciones de (σ, τ) .

Dado que la aplicación W desplaza las posiciones tanto de σ como de τ , para cualquiera par de permutaciones (σ, τ) tal que $uv \in P^{k-1}(xy, uv)$, se cumple que

$$uv \notin P^k(W(\sigma, \tau)). \quad (6.2.6)$$

Por medio de cualquier aplicación H^l con $l \in \{1, \dots, n-1\}$ y la aplicación W , podemos definir la aplicación $W \circ T^l : S^l(xy, uv) \rightarrow S_A \times S_B$, donde $W \circ T^l(\sigma, \tau) = W(T^l(\sigma, \tau))$

Para cada par de permutaciones $(\sigma, \tau) \in S^l(xy, uv)$ con $l \in \{1, \dots, k-1\}$ que genere el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ podemos aplicar $W \circ T^l$ de forma que $H^l(\sigma, \tau) = (\sigma^l, \tau^l)$ y $W(\sigma^l, \tau^l) = (\sigma^*, \tau^*)$.

Debido a los puntos (6.2.4) y (6.2.5) sabemos que

$$xy \in \{\sigma_k^l \tau_k^l, \sigma_k^l \tau_{k+1}^l\} \quad y \quad uv \in P^{k-1}(\sigma^l, \tau^l).$$

Luego, sabemos por el punto (6.2.6) que para el par de permutaciones (σ^l, τ^l) se cumple

$$uv \notin P^k(W(\sigma^l, \tau^l)). \quad (6.2.7)$$

Dadas las definiciones de las aplicaciones de H^l y W se tiene finalmente que

$$uv \notin P^k(\sigma^*, \tau^*) \quad y \quad xy \in \{\sigma_k^* \tau_k^*, \sigma_k^* \tau_{k+1}^*\}. \quad (6.2.8)$$

La aplicación $W \circ T^l$ es inyectiva, siempre y cuando $k \geq 3, n \geq 3$ y $m \geq 4$. Además para un par de aristas fijas xy, uv y una permutación (σ, τ) que genera el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, la aplicación $W \circ T^l$ genera un par de permutaciones $W(T^l(\sigma, \tau)) = (\sigma^*, \tau^*)$ que cumple las condiciones (Recuerde el punto (6.2.3)) para que se genere el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$. De forma que si condicionamos sobre las posibles permutaciones (σ, τ) que generan el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$,

podemos concluir por medio de la aplicación inyectiva $W \circ T^l$ que existen al menos tantos casos favorables para el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$ como casos favorables para el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$. En otras palabras, se tiene que

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)].$$

(II) Las aristas xy, uv comparten vértice en B , es decir $\{xy\} \cap \{uv\} = \{y\}$.

Nuevamente, recordemos que el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ solo ocurre si se cumple los siguientes dos puntos al mismo tiempo:

$$xy \in P^k(\sigma, \tau) \quad \text{y} \quad uv \in \{\sigma_k \tau_k, \sigma_k \tau_{k+1}\}.$$

Si consideramos $v = y$ más las condiciones para que ocurra el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, entonces se tiene que

$$xy = \sigma_{k-1} \tau_k \quad \text{y} \quad uv = \sigma_k \tau_k.$$

Análogamente al primer caso, definimos la siguiente biyección $H' : S_A \rightarrow S_A$, donde $H'(\sigma) = \sigma'$ viene dada por

$$\sigma'_i = \begin{cases} \sigma_k & \text{para } i = k-1, \\ \sigma_{k-1} & \text{para } i = k, \\ \sigma_i & \text{para todo } i \neq k, k-1. \end{cases}$$

De esta manera la biyección H' desplaza la posición de σ_{k-1} en una posición hacia adelante, y σ_k en una posición hacia atrás.

Procedemos a condicionar los eventos $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ y $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$ sobre los posibles caminos que los definen. Podemos notar que para este caso que al aplicar $H'(\sigma)$ a la permutación σ de un camino $P^k(\sigma, \tau)$ que permita que suceda el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, encontramos un camino $P^k(H'(\sigma), \tau)$ que genera un caso favorable para el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$.

Esto ultimo es debido a que si consideramos el camino $P^k(H'(\sigma), \tau)$

entonces se cumple que $xy = \sigma'_k \tau_k$, y por lo tanto $xy \notin P^{k-1}(H'(\sigma), \tau)$. Además, dado que para el camino $P^k(H'(\sigma), \tau)$ también se cumple que $uv = \sigma'_{k-1} \tau_k$, y en consecuencia, $uv \in P^k(H'(\sigma), \tau)$. De forma que se cumplen las dos condiciones para que ocurra el evento $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$.

Dado que $n \geq 2$ y $k \geq 2$ se tiene que la aplicación H' es biyectiva y concluimos que al menos existen tanto casos favorables de $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ como casos favorables de $\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)$, y por lo tanto

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)].$$

(III) Las aristas xy, uv comparten vértice en A , es decir $\{xy\} \cap \{uv\} = \{x\}$.

Dado que el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ solo ocurre si

$$xy \in P^k(\sigma, \tau) \quad \text{y} \quad uv \in \{\sigma_k \tau_k, \sigma_k \tau_{k+1}\}.$$

Si consideramos $x = u$ más las condiciones para que ocurra el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$, notamos que es imposible que ocurra el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$ debido a la definición de los caminos $P^{k-1}(\sigma, \tau)$ y $P^k(\sigma, \tau)$. Puesto que si las aristas xy, uv comparten vértice en A y $xy \in P^{k-1}(\sigma, \tau)$, entonces $uv \in P^{k-1}(\sigma, \tau)$, siendo así una contradicción a las condiciones para que ocurra el evento $\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)$. De esta manera para este caso se tiene que

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] = 0,$$

y por lo tanto

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)].$$

Concluyendo así que para cualquier par de aristas fijas xy, uv , $n \geq 3$, $m \geq 4$ y $k \geq 2$ se cumple que

$$\Pr[\mathcal{E}^{k-1}(xy, uv) \setminus \mathcal{E}^k(xy, uv)] \leq \Pr[\mathcal{E}^k(xy, uv) \setminus \mathcal{E}^{k-1}(xy, uv)],$$

demostrando así lo deseado. □

La Proposición 6.2.2 para todo $n \geq 3$, $m \geq 4$ y $m > n$ nos asegura que la probabilidad de borrar un arco en el dígrafo auxiliar asociado al grafo $K_{n,m}$ es máxima al considerar caminos de largo máximo $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$. En base a esto, centraremos nuestro análisis del Algoritmo 5 de aquí en adelante únicamente en caminos $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$. Cabe recalcar además que la Proposición 6.2.2 nos proporciona información para la gran mayoría de valores de n, m , siendo el único caso realmente importante que resta por analizar y que no consideraremos en este análisis, cuando $n = 2$ y $m \geq 4$.

Dado que enfocaremos el análisis de nuestro algoritmo cuando se seleccionan caminos $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$, deseamos entender con qué probabilidad borramos un arco en el dígrafo auxiliar al agregar el camino $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$, es decir, estimar $\Pr[\mathcal{E}^n(xy, uv)]$ para un par de aristas xy, uv fijas pero arbitrarias.

Proposición 6.2.2. Sean $K_{n,m}$ un grafo bipartito completo con $n < m$, sean xy y uv aristas distintas de $K_{n,m}$ y un camino aleatorio $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$. Entonces

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{2}{m} - O\left(\frac{1}{m^2}\right) \leq \frac{2}{m}.$$

Demostración. Para el $\mathcal{E}^n(xy, uv)$ podemos condicionar sobre las posibles permutaciones σ en S_A , luego

$$\Pr[\mathcal{E}^n(xy, uv)] = \sum_{\sigma \in S_A} \Pr[\mathcal{E}^n(xy, uv) | \boldsymbol{\sigma} = \sigma] \Pr[\boldsymbol{\sigma} = \sigma].$$

Ahora bien, analizaremos los eventos $\mathcal{E}^n(xy, uv)$ para los 3 casos posibles:

- (I) Las aristas xy, uv son disjuntas, es decir $\{xy\} \cap \{uv\} = \emptyset$.

Si $\{xy\} \cap \{uv\} = \emptyset$, entonces las aristas xy, uv no comparten vértices, y por lo tanto $x \neq u$ y $y \neq v$. Dado que condicionar sobre las permutaciones de S_A es equivalente a condicionar sobre las posibles asignaciones de $\boldsymbol{\sigma}$ para x y u , obtenemos que

$$\Pr[\mathcal{E}^n(xy, uv)] = \sum_{(i,j)} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}] \Pr[\{\sigma_i, \sigma_j\} = \{x, u\}].$$

Dado que la $\Pr[\{\sigma_i, \sigma_j\} = \{x, u\}]$ es equivalente a seleccionar un par (i, j) del total de pares ordenados posibles $2\binom{n}{2}$, entonces $\Pr[\{\sigma_i, \sigma_j\} = \{x, u\}] = 1/(2\binom{n}{2})$, y por lo tanto

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{1}{2\binom{n}{2}} \sum_{(i,j)} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}].$$

Dividiremos el análisis para los eventos $\mathcal{E}^n(xy, uv)$ en los índices consecutivos y no consecutivos,

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{1}{2\binom{n}{2}} (P_{consecutivos} + P_{no\ consecutivos}), \quad (6.2.9)$$

donde

$$P_{consecutivos} = \sum_{\substack{(i,j) \\ consecutivos}} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}], \quad (6.2.10)$$

y

$$P_{no\ consecutivos} = \sum_{\substack{(i,j) \\ no\ consecutivos}} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}]. \quad (6.2.11)$$

Para el caso de índices consecutivos para una permutación fija σ tal que $\sigma_i = x$ y $\sigma_j = u$ observe la Figura 6.2.5. En ella se muestran los casos favorables de asignación de τ para y con color azul y con color rojo los casos de asignación de τ para v que debemos evitar para que ocurra el evento $\mathcal{E}^n(xy, uv)$.

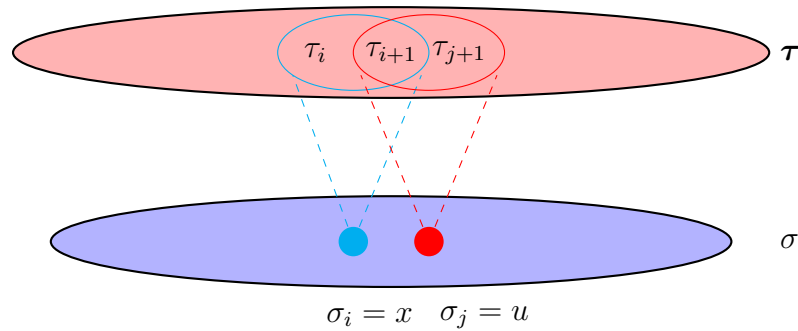


Figura 6.2.5: Las aristas no comparten vértices y además i, j son consecutivos.

Existen dos asignaciones de τ favorables para y , $\tau_i = y$ o $\tau_{i+1} = y$. Si se fija

$\tau_i = y$ restan $m - 3$ posibles asignaciones favorables para v , en cambio si se fija $\tau_{i+1} = y$ restan $m - 2$ posibles asignaciones favorables para v . Fijando alguna combinación de asignaciones favorables para v e y , existen $(m - 2)!$ permutaciones de las $(m - 2)$ asignaciones restantes. Dado que existen $m!$ permutaciones en S_B y dado que para dos naturales $i \leq n$ y $j \leq n$ existen $2(n - 1)$ pares ordenados (i, j) consecutivos se concluye que

$$P_{consecutivos} = 2(n - 1) \left(\frac{(m - 3)(m - 2)! + (m - 2)(m - 2)!}{m!} \right),$$

siendo así

$$P_{consecutivos} = \frac{2(n - 1)(2m - 5)(m - 2)!}{m!}.$$

Ahora bien para calcular $P_{no\ consecutivos}$ considere un permutación fija σ tal que $\sigma_i = x$ y $\sigma_j = u$ donde los índices i, j no son consecutivos. Luego observe la Figura 6.2.6, en ella se muestran los casos favorables de asignación de τ para y con color azul y con color rojo los casos de asignación de τ para v que debemos evitar para que ocurra el evento $\mathcal{E}^n(xy, uv)$.

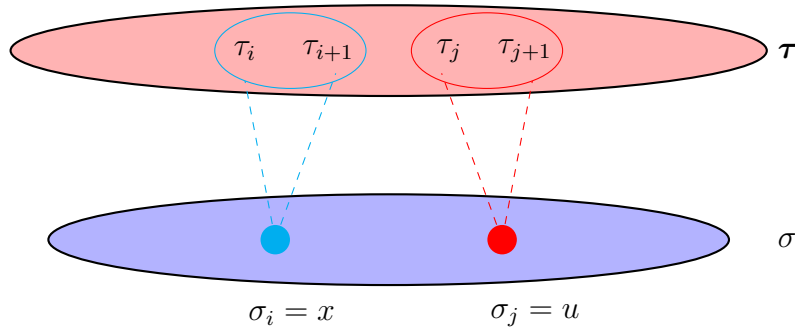


Figura 6.2.6: Las aristas no comparten vértices y además i, j no son consecutivos.

Existen dos asignaciones de τ favorables para y , $\tau_i = y$ o $\tau_{i+1} = y$. Si se fija $\tau_i = y$ o $\tau_{i+1} = y$ entonces restan $m - 3$ posibles asignaciones favorables para v . Fijando alguna combinación de asignaciones favorables para v e y , existen $(m - 3)!$ permutaciones de las $(m - 3)$ asignaciones restantes. Dado que existen $m!$ permutaciones en S_B y dado que para dos naturales $i \leq n$ y $j \leq n$ existen $(n - 1)(n - 2)$ pares ordenados (i, j) no consecutivos se

concluye que

$$P_{no\ consecutivos} = 2(n-1)(n-2) \left(\frac{(m-3)(m-2)!}{m!} \right).$$

De esta manera, recordado que las ecuaciones (6.2.9),(6.2.10),(6.2.11) obtenemos que para el caso que las aristas xy y uv no compartan vértices entre si, se tiene que

$$\begin{aligned} \Pr[\mathcal{E}^n(xy, uv)] &= \frac{1}{2\binom{n}{2}} (P_{consecutivos} + P_{no\ consecutivos}) \\ &= \frac{2(n-1)(m-2)!}{2\binom{n}{2}m!} \left((2m-5) + (n-2)(m-3) \right). \end{aligned}$$

Luego, para n y m suficientemente grandes se tiene que

$$\begin{aligned} \Pr[\mathcal{E}^n(xy, uv)] &= \frac{2(n-1)(m-2)!}{2m!\binom{n}{2}} \left((2m-5) + (n-2)(m-3) \right) \\ &= \frac{\left(2n^2m - nm - 6n^2 - 16n - 2m + 4 \right)}{(m(m-1)(n(n-1)))} \\ &= \frac{2n}{(m-1)(n-1)} + \frac{\left(-nm - 6n^2 - 16n - 2m + 4 \right)}{(m(m-1)n(n-1))}, \end{aligned}$$

de donde obtenemos que

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{2}{m} - O\left(\frac{1}{m^2}\right) \leq \frac{2}{m}.$$

(II) Las aristas xy, uv comparten vértice en B , es decir $\{xy\} \cap \{uv\} = \{y\}$.

Si las aristas xy, uv comparten vértice en B , entonces las aristas xy y uv son tales que $y = v$. Nuevamente dado que condicionar sobre las permutaciones de S_A es equivalente a condicionar sobre las posibles asignaciones de σ para x y u , obtenemos que

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{1}{2\binom{n}{2}} \sum_{(i,j)} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}].$$

Nuevamente dividimos en casos de índices consecutivos y no consecutivos,

volviendo a formular la ecuación

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{1}{2\binom{n}{2}} (P_{consecutivos} + P_{no\ consecutivos}),$$

donde nuevamente

$$P_{consecutivos} = \sum_{\substack{(i,j) \\ consecutivos}} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}],$$

y

$$P_{no\ consecutivos} = \sum_{\substack{(i,j) \\ no\ consecutivos}} \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i, \sigma_j\} = \{x, u\}].$$

Para una permutación fija σ tal que $\sigma_i = x$ y $\sigma_j = u$ donde los índices i, j son consecutivos observe la Figura 6.2.7. En ella se muestran los casos favorables de asignación de τ para y con color azul y los casos de asignación de τ para v que debemos evitar para que ocurra el evento $\mathcal{E}^n(xy, uv)$.

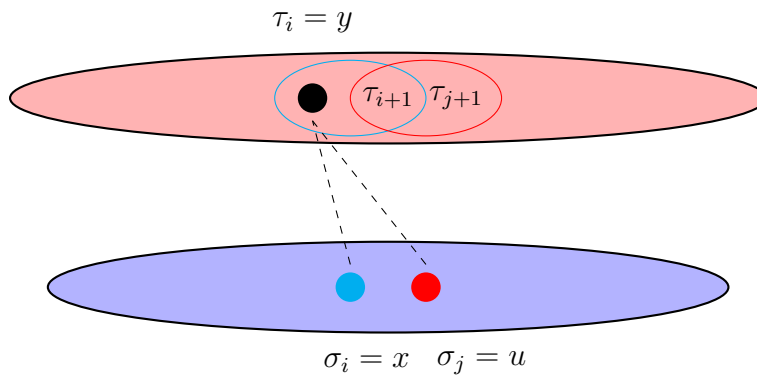


Figura 6.2.7: Las aristas comparten un vértice $y = v$ y además i, j son consecutivos.

Dado que las aristas comparten el vértice y solo existe una asignación favorable $\tau_i = y$ marcada en negro en la Figura 6.2.7. De esta manera fijando $\tau_i = y$ existen $(m - 1)!$ permutaciones de las $(m - 1)$ asignaciones restantes. Dado que existen $m!$ permutaciones en S_B y dado que para dos naturales $i \leq n$ y $j \leq n$ existen $2(n - 1)$ pares ordenados (i, j) consecutivos se concluye

que

$$P_{consecutivos} = 2(n-1) \left(\frac{(m-1)(m-2)!}{m!} \right).$$

Por otro lado, para una permutación fija σ tal que $\sigma_i = x$ y $\sigma_j = u$ donde los índices i, j no son consecutivos observe la Figura 6.2.8.

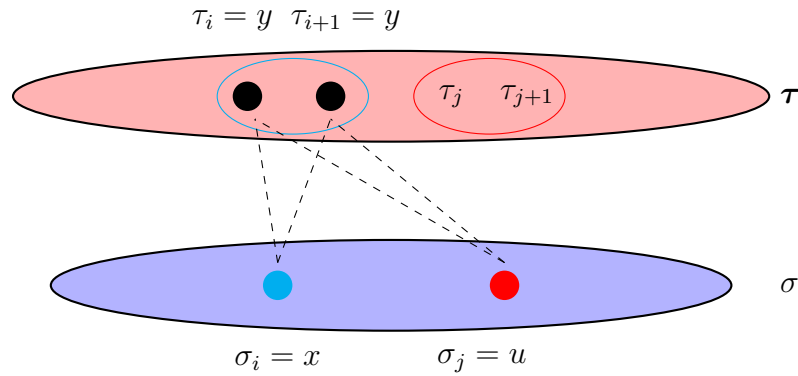


Figura 6.2.8: Las aristas comparten un vértice $y = v$ y además i, j no son consecutivos.

Dado que las aristas comparten el vértice y , para los pares ordenados (i, j) no consecutivos existen dos asignaciones favorables $\tau_i = y$ y $\tau_{i+1} = y$ marcadas en negro en la Figura 6.2.8. De esta manera fijando $\tau_i = y$ o $\tau_{i+1} = y$ existen $(m-1)!$ permutaciones de las $(m-1)$ asignaciones restantes. Dado que existen $m!$ permutaciones en S_B y dado que para dos naturales $i \leq n$ y $j \leq n$ existen $2(n-1)(n-1)$ pares ordenados (i, j) no consecutivos se concluye que

$$P_{no\ consecutivos} = 2(n-2)(n-1) \left(\frac{(m-1)(m-2)!}{m!} \right).$$

De esta manera, para el caso que las aristas xy, uv compartan vértice en B , se tiene que $\Pr[\mathcal{E}^n(xy, uv)]$ viene dada por

$$\frac{1}{2 \binom{n}{2}} \left(2(n-1) \left(\frac{(m-1)(m-2)!}{m!} \right) + 2(n-2)(n-1) \left(\frac{(m-1)(m-2)!}{m!} \right) \right).$$

Reordenando la expresión anterior concluimos que para este caso

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{(m-2)!(2n-1)(m-1)}{2\binom{n}{2}m!}.$$

Luego, para n y m suficientemente grandes se tiene que

$$\begin{aligned} \Pr[\mathcal{E}^n(xy, uv)] &= \frac{(m-2)!(2n-1)(m-1)}{2\binom{n}{2}m!} \\ &= \frac{\left(2n^2m - 4nm - 2n^2 + 4n - 2\right)}{(m(m-1)(n(n-1)))} \\ &= \frac{2n}{(m-1)(n-1)} + \frac{\left(-4nm - 2n^2 + 4n - 2\right)}{(m(m-1)n(n-1))}, \end{aligned}$$

de donde obtenemos que

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{2}{m} - O\left(\frac{1}{m^2}\right) \leq \frac{2}{m}.$$

(III) Las aristas xy, uv comparten vértice en A , es decir $\{xy\} \cap \{uv\} = \{x\}$.

Si las aristas xy, uv comparten vértice en A , entonces se tiene que $x = v$. Para este caso procedemos a condicionar sobre las posibles asignaciones de σ para x , donde obtenemos que

$$\Pr[\mathcal{E}^n(xy, uv)] = \sum_{i=1}^n \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i\} = \{x\}] \Pr[\{\sigma_i\} = \{x\}].$$

Dado que $\Pr[\{\sigma_i\} = \{x\}]$ es equivalente a seleccionar aleatoriamente un elemento de un conjunto de tamaño n , se tiene que $\Pr[\{\sigma_i\} = \{x\}] = 1/n$, y por lo tanto

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{1}{n} \sum_{i=1}^n \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i\} = \{x\}].$$

Consideremos una permutación fija σ tal que $\sigma_i = x$, observe la Figura 6.2.9.

Solo las asignaciones $\tau_i = y, \tau_{i+1} = y$ marcadas con azul en la Figura 6.2.9 son opciones favorables para que ocurra el evento $\mathcal{E}^n(xy, uv)$. Si se cumple

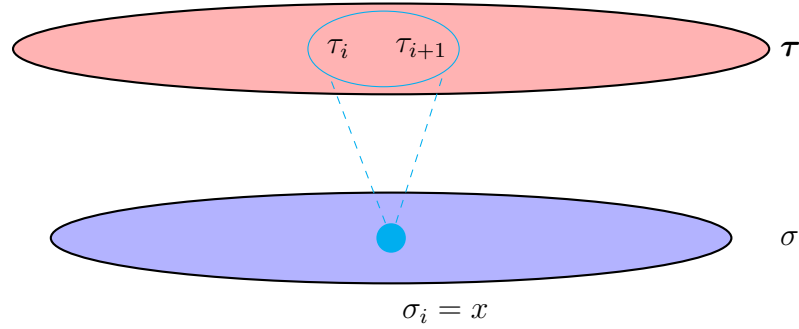


Figura 6.2.9: Las aristas comparten el vértice $x = u$.

que $\tau_i = y$, entonces se debe cumplir que $\tau_{i+1} \neq v$. Por otro lado, si se tiene que $\tau_{i+1} = y$, entonces $\tau_i \neq v$. De esta manera, fijando $\tau_i = y$ o $\tau_{i+1} = y$ existen $m - 2$ asignaciones favorables para v . Si fijamos alguna combinación de asignaciones favorables para v e y existen $(m - 2)!$ permutaciones de las $(m - 2)$ asignaciones restantes. Dado que existen $m!$ permutaciones en S_B obtenemos que

$$\begin{aligned} \Pr[\mathcal{E}^n(xy, uv)] &= \frac{1}{n} \sum_{i=1}^n \Pr[\mathcal{E}^n(xy, uv) | \{\sigma_i\} = \{x\}] \\ &= \frac{1}{n} \sum_{i=1}^n \frac{2(m-2)(m-2)!}{m!} \\ &= \frac{2(m-1)(m-2)!}{nm!}. \end{aligned}$$

Luego, para n y m suficientemente grandes se tiene que

$$\begin{aligned} \Pr[\mathcal{E}^n(xy, uv)] &= \frac{2(m-1)(m-2)!n}{nm!} \\ &= \frac{2m}{m(m-1)} - \frac{2}{m(m-1)} \\ \Pr[\mathcal{E}^n(xy, uv)] &= \frac{2}{m} - O\left(\frac{1}{m^2}\right) \leq \frac{2}{m}. \end{aligned}$$

Concluyendo así que

$$\Pr[\mathcal{E}^n(xy, uv)] = \frac{2}{m} - O\left(\frac{1}{m^2}\right) \leq \frac{2}{m}.$$

para todo par de aristas fijas xy y uv . □

Considerando que seleccionamos caminos de largo máximo $P^n(\sigma, \tau)$, podemos estimar cuantas iteraciones como máximo esperamos que sean necesarias para borrar todos los arcos en el dígrafo auxiliar, y por lo tanto, obtener un sistema separador fuerte de caminos para el grafo $K_{n,m}$.

Definimos para cada $t \geq 1$ la variable aleatoria \mathbf{M}_t como la cantidad de arcos del dígrafo $D_{\mathcal{P}}$ asociada a la familia $\mathcal{P}_t = \{E(\mathbf{P}_1^n), \dots, E(\mathbf{P}_t^n)\}$ generada por el Algoritmo 5 en t iteraciones (Note que a diferencia de la variable \mathbf{N}_t , la variable \mathbf{M}_t modela la cantidad de arcos que restan por borrar en el dígrafo auxiliar). Siguiendo esta definición, definimos \mathbf{M}_1 como la cantidad de arcos del dígrafo auxiliar al seleccionar solo un camino $P^n(\sigma, \tau)$. Sabemos por la Proposición 6.1.1 que $\mathbf{M}_1 = 2 \binom{mn}{2} - 2n(mn - 2n)$.

Definimos además para todo $t > 1$ la variable aleatoria X_t que modela el conjunto de arcos borrados en D_t cuando el algoritmo agrega un camino $P^n(\sigma, \tau)$ al sistema \mathcal{P}_t . Teniendo en cuenta estas variables y la Proposición 6.2.2 podemos deducir que para cualquier $t > 1$ se tiene que:

$$\mathbb{E}(|X_t|) = \sum_{(e,f) \in D_t} \Pr[(e, f) \in X_t] = \sum_{(e,f) \in D_t} \Pr[\mathcal{E}^n(e, f)] \leq \frac{2}{m} \mathbb{E}(\mathbf{M}_t).$$

En lo que sigue supondremos que la anterior desigualdad se alcanza con igualdad para ciertos t , esta suposición sera demostrada más adelante para ciertos valores t . Notemos además que si $\mathbb{E}(|X_t|) = \frac{2}{m} \mathbb{E}(\mathbf{M}_t)$, entonces para los $t \in \mathbb{N}$ tal que $\mathbb{E}(\mathbf{M}_t) \geq m/2$ aún en términos de la esperanza se borra al menos un arco en el dígrafo auxiliar.

La siguiente proposición estima la cantidad de aristas restantes en el dígrafo D_t .

Proposición 6.2.3. Sea $n < m$, entonces

$$\mathbb{E}(\mathbf{M}_t) = \left(1 - \frac{2}{m}\right)^{t-1} \mathbf{M}_1,$$

para todo $t \geq 2$.

Demostración. En primer lugar notemos que para $t = 2$ el Algoritmo 5 solo a seleccionado dos caminos $\mathbf{P}_1^n, \mathbf{P}_2^n$ de largo $2n$, y por lo tanto, la cantidad esperada

de arcos borrados corresponde a

$$\begin{aligned} E(\mathbf{M}_2) &= E(\mathbf{M}_1) - E(X_1) \\ &= E(\mathbf{M}_1) - \sum_{(e,f) \in D_1} \Pr[(e, f) \in X_1] \\ E(\mathbf{M}_2) &= E(\mathbf{M}_1) - \sum_{(e,f) \in D_1} \Pr[\mathcal{E}^n(e, f)]. \end{aligned}$$

Luego, si consideramos que la ecuación de la Proposición 6.2.2 se alcanza con igualdad obtenemos que

$$\begin{aligned} E(\mathbf{M}_2) &= E(\mathbf{M}_1) - \sum_{(e,f) \in D_1} \Pr[\mathcal{E}^n(e, f)] \\ &= E(\mathbf{M}_1) - \frac{2}{m} E(\mathbf{M}_1) \\ E(\mathbf{M}_2) &= \left(1 - \frac{2}{m}\right) \mathbf{M}_1, \end{aligned}$$

esto ultimo debido a que $\mathbf{M}_1 = 2 \binom{mn}{2} - 2n(mn - 2n)$, y por lo tanto $E(\mathbf{M}_1) = \mathbf{M}_1$.

Ahora bien, razonaremos por inducción. Consideremos que la Ecuación 6.2.3 se cumple para algún $i \in \mathbb{N}$. Luego, dado que X_i corresponde al conjunto de arcos borrados en D_i cuando el Algoritmo 5 agrega un camino $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$ se tiene que

$$\begin{aligned} E(\mathbf{M}_{i+1}) &= E(\mathbf{M}_i) - E(X_i) \\ &= E(\mathbf{M}_i) - \sum_{(e,f) \in D_i} \Pr[(e, f) \in X_i] \\ &= E(\mathbf{M}_i) - \sum_{(e,f) \in D_i} \Pr[\mathcal{E}^n(e, f)] \\ &= E(\mathbf{M}_i) - \frac{2}{m} E(\mathbf{M}_i) \\ E(\mathbf{M}_{i+1}) &= \left(1 - \frac{2}{m}\right) E(\mathbf{M}_i). \end{aligned}$$

Luego, como $E(\mathbf{M}_i) = \left(1 - \frac{2}{m}\right)^{i-1} \mathbf{M}_1$ se concluye finalmente que

$$E(\mathbf{M}_{i+1}) = \left(1 - \frac{2}{m}\right)^i \mathbf{M}_1,$$

donde $\mathbf{M}_1 = 2 \binom{mn}{2} - 2n(mn - 2n)$. □

Al igual que la variable \mathbf{N}_t , deseamos analizar si $E(\mathbf{M}_t)$ es representativa con respecto a la variable \mathbf{M}_t . Para poder utilizar nuevamente la Proposición 6.1.4 utilizaremos nuestra definición alternativa de caminos aleatorios de largo k en $K_{n,m}$

$$P^k(\boldsymbol{\mu}) := P^k(\boldsymbol{\sigma}, \boldsymbol{\tau}).$$

Definiremos para cada $t \in \mathbb{N}$ las funciones $g^t : S_N \rightarrow \mathbb{N}$, donde $g^t(\boldsymbol{\mu})$ corresponde a la cantidad de arcos en el grafo $K_{n,m}$ en la iteración t del Algoritmo al agregar el camino $P^k(\boldsymbol{\mu})$. Notemos que $E(g^t) = E(\mathbf{M}_t)$, puesto que \mathbf{M}_t corresponde a la variable aleatoria que modela la cantidad de arcos en el digrafo D_t generado por la familia $\mathcal{P}_t = \{E(\mathbf{P}_1^n), \dots, E(\mathbf{P}_t^n)\}$. Estudiaremos solo que ocurre cuando $t \geq t_c$, es decir, que ocurre con $E(\mathbf{M}_t)$ cuando ya sean cubierto casi todas las aristas en el grafo $K_{n,m}$ por algún camino $P \in \mathcal{P}_t$. Para las iteraciones $t \leq t_c$, la variable $E(\mathbf{M}_t)$ será analizada en nuestros experimentos computacionales en el Capítulo 7.

Ahora bien, consideremos dos permutaciones fijas pero arbitrarias μ y μ' que difieren en una trasposición. Tal y como se discutió para la variable \mathbf{N}_t , los caminos $P^n(\mu)$ y $P^n(\mu')$ difieren en lo más 8 aristas. Dado que nos enfocamos en la iteraciones $t \geq t_c$, entonces las aristas están ya cubiertas en casi en su totalidad por caminos en \mathcal{P}_t , y por lo tanto existen a lo más para cada arista $e \in D_t$ un total de $2n$ arcos en $D_t^+(e)$. Luego, sin pérdida de generalidad, como cada arista e tal que $e \in P^k(\mu)$ y $e \notin P^k(\mu')$ podría borrar a lo más $2n$ arcos en D_t se deduce que

$$|g^t(\mu) - g^t(\mu')| \leq 16n.$$

En base a esta información y la Proposición 6.1.4, podemos asegurar que para una permutación $\boldsymbol{\mu} \in S_N$ escogida de manera aleatoria se cumple que

$$\Pr[|g^t(\boldsymbol{\mu}) - E(g^t)| \geq 16nc] \leq 2e^{-c^2/(4N)} \quad \text{para todo } t \in \mathbb{N},$$

o equivalentemente

$$\Pr[|g^t(\boldsymbol{\mu}) - E(\mathbf{M}_t)| \geq 16nc] \leq 2e^{-c^2/(4N)} \quad \text{para todo } t \in \mathbb{N}. \quad (6.2.12)$$

Ahora bien, nuevamente dado que buscamos que la probabilidad de que $g^t(\boldsymbol{\mu})$ y $E(\mathbf{M}_t)$ difieran sea pequeña lo largo de las iteraciones, entonces debemos imponer en la ecuación (6.2.12) que $e^{-c^2/(4N)} \ll 0$, y por lo tanto

$$c \gg 2\sqrt{N} = 2\sqrt{n+m}. \quad (6.2.13)$$

Como se mencionó anteriormente, deseamos estudiar para que valores de t se tiene que la diferencia $|g^t(\boldsymbol{\mu}) - E(\mathbf{M}_t)|$ es pequeña. Para este estudio, esto puede ser interpretado como $E(\mathbf{M}_t) \gg 16nc$, o equivalentemente

$$c \ll \frac{(1 - \frac{2}{m})^t}{16mn^2}. \quad (6.2.14)$$

Combinando las ecuaciones (6.2.13) y (6.2.14), obtenemos

$$2\sqrt{n+m} \ll \frac{(1 - \frac{2}{m})^t}{16mn^2},$$

de donde deducimos que esto ocurre a partir las iteraciones $t \in \mathbb{N}$ tales que

$$t \ll \frac{m}{2} \ln \left(\frac{mn^2}{32\sqrt{m+n}} \right).$$

Dado que para $k = n$ se tiene que $t_c \geq \frac{m}{2} \ln(mn)$, entonces para n y m suficientemente grandes se tiene que $\frac{m}{2} \ln(mn) \leq \frac{m}{2} \ln \left(\frac{mn^2}{32\sqrt{m+n}} \right)$. A partir de esto, hemos concluidos que $E(\mathbf{M}_t)$ es representativa con respecto a \mathbf{M}_t para la mayoría de iteraciones posteriores a la iteración t_c .

Ahora bien, podemos estimar la cantidad esperada de iteraciones t necesarias para que $E(\mathbf{M}_t) < m/2$, y por lo tanto borrar en términos de la esperanza menos de un arco por iteración. Denotaremos t_b a la variable aleatoria que modela la cantidad de iteraciones necesarias para que en términos de esperanza se borre menos de un arco al agregar un camino $P^n(\boldsymbol{\sigma}, \boldsymbol{\tau})$. Si despejamos la ecuación:

$$\left(1 - \frac{2}{m}\right)^{t-1} \mathbf{M}_1 < \frac{m}{2},$$

entonces

$$t_b > \frac{m}{2} \ln \left(\frac{(mn(mn-1) - 2n^2m - 4n^2) e^{2/m}}{2m} \right).$$

En nuestro análisis anteriormente hemos calculado que la cantidad esperada de iteraciones para cubrir todas las aristas salvo una es mayor igual a $t_c = \frac{m}{2} \ln(mn)$ para el caso que se seleccionen caminos de largo $2n$. Dado que $\frac{m}{2} \ln(mn) < t_b$, podemos considerar t_b como una mejor cota inferior para la cantidad esperada de iteraciones del Algoritmo 5. Para facilitar el análisis, notemos que $t_b \geq \frac{m}{2} \ln(n^2m) > t_c$. Para mayor claridad consideraremos en nuestro análisis la cota inferior $\frac{m}{2} \ln(n^2m)$ para la cantidad esperada de iteraciones del Algoritmo 5.

Sea t^* la primera iteración tal que $E(\mathbf{M}_{t^*}) < m/2$. Para las iteraciones $t^* \geq t$ se tiene que el Algoritmo 5 borra en términos de la esperanza, menos de un arco en el digrafo auxiliar al agregar un camino. La siguiente proposición permite estimar la cantidad esperada de iteraciones que son necesarias para borrar todos los arcos restantes después de la iteración t^* , y en consecuencia, detener el Algoritmo 5.

Proposición 6.2.4. Sea $K_{n,m}$ el grafo bipartito completo con $n < m$ suficientemente grandes, entonces la cantidad de iteraciones esperadas restantes para que el Algoritmo 5 borre los $E(\mathbf{M}_{t^*})$ arcos restantes es menor igual a $\frac{m}{2} (\ln(\frac{m}{2}) + 1)$.

Demostración. Para cada $i \in \{1, \dots, E(\mathbf{M}_{t^*})\}$ definimos las variables aleatorias T_i como la cantidad de caminos seleccionados para que el digrafo auxiliar reduzca sus arcos de i a $i-1$. Estas variables aleatorias son geométricas con parámetro $p_i = \frac{2}{m}i$ cada una respectivamente ya que la cantidad de arcos restantes es i y la probabilidad de borrar un arco para un m suficientemente grande es igual a $\frac{2}{m}$.

Así, podemos calcular el valor esperado para cada T_i , siendo este igual a $\frac{m}{2}i$. A partir de esto, la cantidad esperada de iteraciones necesarias para borrar los \mathbf{M}_{t^*} arcos restantes en el digrafo auxiliar corresponde a

$$\sum_{i=1}^{E(\mathbf{M}_{t^*})} E(T_i) = \frac{m}{2} \sum_{i=1}^{E(\mathbf{M}_{t^*})} \frac{1}{i} \leq \frac{m}{2} \ln(E(\mathbf{M}_{t^*}) + 1),$$

luego como $E(\mathbf{M}_{t^*}) < m/2$, se tiene que

$$\sum_{i=1} E(T_i) \leq \frac{m}{2} \left(\ln \left(\frac{m}{2} \right) + 1 \right),$$

concluyendo así lo deseado. \square

6.3. Cota superior para la cantidad esperada de iteraciones

En nuestro análisis hemos estimado que la cantidad esperada de iteraciones del Algoritmo 5 se encuentra acotada inferiormente por $\frac{m}{2} \ln(n^2 m)$. Ahora bien, podemos obtener también una cota superior para la cantidad esperada de iteraciones de la siguiente forma.

Sean $D = D_0 \supseteq D_1 \supseteq D_2 \supseteq \dots \supseteq D_t \supseteq \dots$ los digrafos generados por el Algoritmo 5 al seleccionar los caminos $P_1, P_2, \dots, P_t, \dots$. Luego, para cada arco $(e, f) \in E(D)$ y para cada $i \in \mathbb{N}$ definimos la siguiente variable aleatoria

$$\mathbf{Y}_i^{(e,f)} = \begin{cases} 1 & \text{Si } e \in E(P_i), f \notin E(P_i), \\ 0 & \text{para cualquier otro caso.} \end{cases}$$

Para cada arco (e, f) las variables aleatorias $\mathbf{Y}_i^{(e,f)}$ corresponden a variables indicatrices que identifican los caminos P seleccionados por el Algoritmo 5, tal que $e \in E(P), f \notin E(P)$, y por lo tanto los caminos P que semi-separan e de f . Además, para cada arco (e, f) las variables $\mathbf{Y}_i^{(e,f)}$, con $i \in \mathbb{N}$, son variables Bernulli independientes entre sí. Todas estas variables Bernulli por la Proposición 6.2.2 sabemos que poseen parámetro $p_{(e,f)} = \frac{2}{m} - O\left(\frac{1}{m^2}\right)$.

Definimos además para cada arco $(e, f) \in D$ las variables aleatorias

$$\mathcal{T}^{(e,f)} = \min\{i \geq 1 : \mathbf{Y}_i^{(e,f)} = 1\}.$$

Para cada arco (e, f) la variable aleatoria $\mathcal{T}^{(e,f)}$ corresponde a la primera iteración i tal que la arista e se ha semi-separado de f , y por lo tanto, la iteración i tal que el arco (e, f) se ha borrado del digrafo auxiliar D . Además, dado que las variables $\mathbf{Y}_i^{(e,f)}$, con $i \in \mathbb{N}$ son variables Bernulli independientes entre si con

parámetro $p_{(e,f)}$, entonces para cualquier arco $(e, f) \in E(D)$ se tiene que $\mathcal{T}^{(e,f)}$ es una variable geométrica de parámetro $p_{(e,f)}$.

Dado que deseamos estimar la cantidad esperada máxima de iteraciones para que el Algoritmo 5 se detenga, definimos la variable aleatoria

$$\mathcal{T} = \max_{(e,f) \in D} \mathcal{T}^{(e,f)}.$$

Podemos deducir que la variable aleatoria \mathcal{T} corresponde a la cantidad de iteraciones necesarias para borrar la última arista en digrafo D , y por lo tanto, detener el Algoritmo 5.

A partir de estas variables aleatorias, se tiene la siguiente proposición

Proposición 6.3.1. Sea $n < m$ y $k \in \mathbb{N}$, entonces

$$\Pr(\mathcal{T} > k) \leq e^{(2 \ln(mn) - (\frac{2}{m} - O(\frac{1}{m^2})))k}.$$

Demostración. Para estimar $\Pr(\mathcal{T} > k)$ considere que

$$\Pr(\mathcal{T} > k) = \Pr(\exists (e, f) \in E(D) : \mathcal{T}^{(e,f)} > k) \leq \sum_{(e,f) \in D} \Pr(\mathcal{T}^{(e,f)} > k).$$

Luego, dado que para cada arco $(e, f) \in D$ se tiene que $\Pr(\mathcal{T}^{(e,f)} > k)$ es equivalente a fallar k veces en una distribución geométrica de parámetro $p_{(e,f)} = \frac{2}{m} - O(\frac{1}{m^2})$, se concluye que $\Pr(\mathcal{T}^{(e,f)} > k) = (1 - (\frac{2}{m} - O(\frac{1}{m^2})))^k$. De esta forma, se tiene que

$$\begin{aligned} \Pr(\mathcal{T} > k) &\leq |E(D)| \left(1 - \left(\frac{2}{m} - O\left(\frac{1}{m^2}\right)\right)\right)^k \\ &= 2 \binom{mn}{2} \left(1 - \left(\frac{2}{m} - O\left(\frac{1}{m^2}\right)\right)\right)^k \\ &\leq (mn)^2 e^{(-\frac{2}{m} - O(\frac{1}{m^2}))k} \\ &= e^{(2 \ln(mn) - (\frac{2}{m} - O(\frac{1}{m^2})))k}. \end{aligned}$$

Por lo tanto

$$\Pr(\mathcal{T} > k) \leq e^{(2\ln(mn) - (\frac{2}{m} - O(\frac{1}{m^2})))k},$$

concluyendo así lo deseado. \square

Por medio de la Proposición 6.3.1 podemos estimar a partir de qué iteración $k \in \mathbb{N}$ se tiene que $\Pr(\mathcal{T} > k) \ll 1$. Si exigimos que $\Pr(\mathcal{T} > k) \ll 1$ entonces se tiene que

$$2\ln(mn) - \left(\frac{2}{m} - O\left(\frac{1}{m^2}\right)\right)k \ll 0,$$

y por lo tanto

$$k \gg (1 + o(1)) m \ln(mn).$$

De esta forma, a partir de una cantidad $(1 + o(1)) m \ln(mn)$ de iteraciones el Algoritmo 5 se detiene con alta probabilidad. A partir de esto, hemos concluido que la cantidad esperada de iteraciones para que el algoritmo se detenga esta acotada superiormente por $(1 + o(1)) m \ln(mn)$. Además, dado que anteriormente hemos estimado que la cantidad esperada de iteraciones se encuentra acotada inferiormente por $\frac{m}{2} \ln(n^2 m)$ podemos concluir que la cantidad esperada de iteraciones se encuentra acotada por una cantidad $\Theta(m \ln(mn))$ de iteraciones.

Capítulo 7

Implementación computacional del algoritmo secuencial

7.1. Implementación computacional

En esta sección presentaremos la implementación computacional realizada en el lenguaje Python del Algoritmo 5 seleccionado caminos de largo máximo:

Input: Grafo bipartito completo $K_{n,m}$

Output: Un sistema separador fuerte \mathcal{P} para el grafo $K_{n,m}$.

- 1: $\mathcal{P} = \emptyset$
 - 2: **while** $E(D_{\mathcal{P}}) \neq \emptyset$ **do**
 - 3: Seleccionar un camino $P = P^n(\sigma, \tau)$ en el grafo $K_{n,m}$
 - 4: **Update** $\mathcal{P} = \mathcal{P} \cup \{E(P)\}$
 - 5: **return** \mathcal{P}
-

El objetivo de esta implementación es obtener resultados experimentales que corroboren los resultados teóricos obtenidos en la sección anterior. Las ideas y funciones que permiten la correcta implementación del Algoritmo 5 se encuentran en el Anexo 9.

7.2. Resultados experimentales:

En esta sección denotaremos al grafo bipartito completo $K_{n,m}$ de N vértices, con partes $|A| = n$ y $|B| = m$ por $K_{\alpha N, (1-\alpha)N}$, donde $|A| = \alpha N$ y $|B| = (1 - \alpha)N$,

de forma que $\alpha \in (0, 1/2)$. Al parámetro α lo llamaremos el *desbalance* del grafo $K_{n,m}$.

A continuación presentaremos los distintos resultados experimentales obtenidos por la implementación computacional del Algoritmo 5 para distintas cantidades de vértices $N \in \{100, 250, 500, 750, 1000\}$ y desbalances $\alpha \in \{0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45\}$.

7.2.1. Medias experimentales de la cantidad de iteraciones t del Algoritmo aleatorio secuencial

En el Capítulo 6 estimamos que la esperanza de la cantidad de iteraciones del Algoritmo 5 se encuentra acotada por una cantidad $\Theta(m \ln(mn))$ de iteraciones.

El siguiente gráfico presenta los resultados experimentales obtenidos para la media de cantidad de iteraciones de la implantación computacional después de 250 intentos y los compara con nuestras cotas obtenidas para una cantidad de vértices $N = 1000$ y desbalance $\alpha = 0,45$. Solo presentamos este gráfico debido a la similitud de los resultados para las demás pruebas con distintos tamaños y desbalances. Posteriormente mostraremos tablas con todos los resultados importantes de estas pruebas.

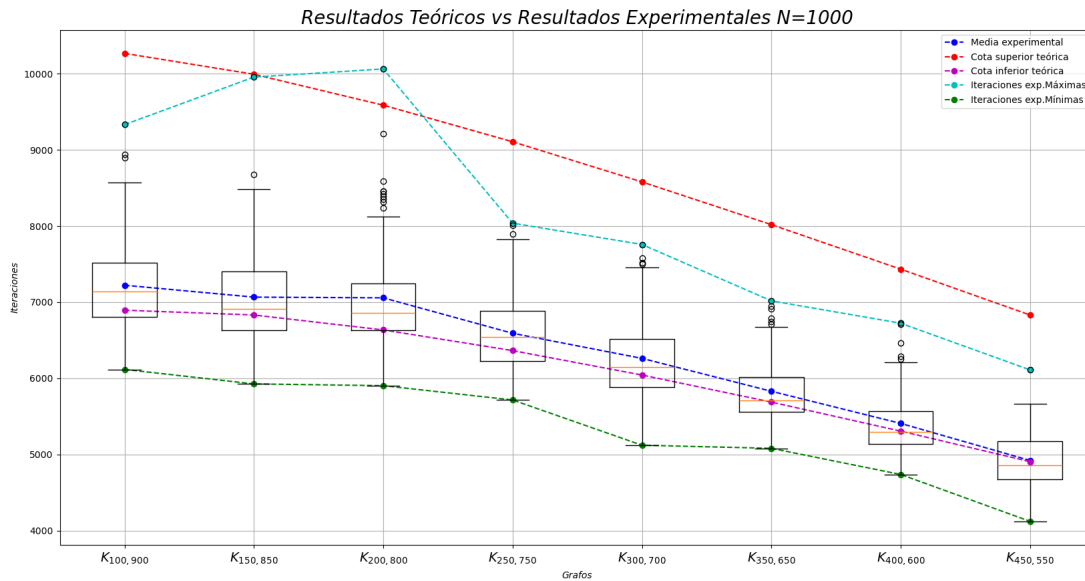


Figura 7.2.1: Media experimental de iteraciones y esperanza teórica de iteraciones del algoritmo para $N = 1000$: El gráfico presenta en su eje de abscisas los distintos grafos $K_{n,m}$ en 1000 vértices para los desbalances $\alpha \in \{0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45\}$. En el eje de las ordenadas se presenta la cantidad de iteraciones del algoritmo. En rojo se marca la cota superior teórica para la esperanza de iteraciones del algoritmo para cada uno de los grafos $K_{n,m}$. En morado se marca la cota inferior teórica para la esperanza de iteraciones del algoritmo para cada uno de los grafos $K_{n,m}$. En azul se marca la media de iteraciones obtenida por la implementación computacional para cada uno de los grafos $K_{n,m}$. En verde se marca la mínima cantidad de iteraciones obtenida para cada grafo $K_{n,m}$ y en celeste se marca la máxima cantidad obtenida de iteraciones para cada grafo $K_{n,m}$. Además, para cada grafo se presenta un gráfico de caja que marca la mediana con una línea naranja, cada cuartil con una línea horizontal negra y con círculos negros sus puntos atípicos.

Dado que las distintas pruebas comparten el patrón, podemos concluir que para todos los casos que hemos investigado la media experimental de iteraciones del Algoritmo 5 es representativa, puesto que sus respectivos gráficos de cajas nos aseguran que la media para cada caso se encuentra cercana a la mediana, y además, los datos se encuentran centrados cerca de la media con una tendencia hacia los valores más pequeños. Siendo así los casos de iteraciones máximas casos atípicos dentro del experimento.

En nuestros experimentos podemos apreciar que la esperanza de iteraciones del algoritmo se encuentra muy cercana a la cota inferior $\frac{m}{2} \ln(mn^2)$. Además, para cada cantidad de vértices N a medida que el desbalance α crece el algoritmo en términos de la media requiere menos iteraciones para finalizar su tarea.

A continuación presentaremos en más detalle las cantidades obtenidas para la media en cada uno de los experimentos. Para cada cantidad de vértices N , la primera tabla muestran los resultados experimentales de nuestro algoritmo y la segunda tabla compara la media experimental de estos resultados con la cota inferior y superior encontradas para cada desbalance α .

$\alpha \backslash$ Media	Experimental	Des. Estándar	Máximo	Mínimo
0.10	489,04	$\pm 54,12$	627	358
0.15	500,23	$\pm 63,78$	705	398
0.20	482,12	$\pm 59,62$	637	374
0.25	473,64	$\pm 58,38$	678	369
0.30	443,64	$\pm 50,02$	617	361
0.35	422,75	$\pm 41,73$	554	339
0.40	396,83	$\pm 37,39$	608	324
0.45	361,29	$\pm 38,38$	498	284

Cuadro 7.2.1: Tabla de valores de la media experimental de iteraciones con $N = 100$.

$\alpha \backslash$ Media	Experimental	Cota superior	Cota inferior
0.10	489,04	612,22	378,492
0.15	500,23	607,81	389,516
0.20	482,12	590,22	387,20
0.25	473,64	565,23	377,321
0.30	443,64	535,48	362,518
0.35	422,75	502,43	344,239
0.40	396,83	466,99	323,373
0.45	361,29	429,77	300,514

Cuadro 7.2.2: Tabla comparativa para la media experimental de iteraciones con $N = 100$.

$\alpha \backslash$ Media	Experimental	Des. Estándar	Máximo	Mínimo
0.10	1441,85	$\pm 148,263$	1852	1202
0.15	1433,14	$\pm 121,489$	1771	1149
0.20	1409,43	$\pm 141,582$	1904	1150
0.25	1366,94	$\pm 139,734$	2053	1167
0.30	1274,51	$\pm 119,593$	1594	1083
0.35	1208,42	$\pm 107,685$	1530	1006
0.40	1123,3	$\pm 103,372$	1474	947
0.45	1040,41	$\pm 88,45$	1330	875

Cuadro 7.2.3: Tabla de valores de la media experimental de iteraciones con $N = 250$.

$\alpha \backslash$ Media	Experimental	Cota superior	Cota inferior
0.10	1441.85	1942,86	1255.56
0.15	1433.14	1906.76	1265.48
0.20	1409.431	1842.06	1242.91
0.25	1366.94	1760.35	1202.96
0.30	1274.51	1659.39	1146.82
0.35	1208.42	1549.51	1081.27
0.40	1123.3	1442.37	1014.58
0.45	1040.41	1331.11	943.30

Cuadro 7.2.4: Tabla comparativa para la media experimental de iteraciones con $N = 250$.

$\alpha \backslash$ Media	Experimental	Des. Estándar	Máximo	Mínimo
0.10	3245.91	± 351.78	4493	2669
0.15	3220.09	± 267.52	4249	2706
0.20	3120.31	± 248.27	3956	2665
0.25	3062.2	± 285.68	3937	2565
0.30	2868.83	± 236.12	3408	2431
0.35	2677.55	± 262.77	3480	2292
0.40	2516.28	± 263.65	3579	2140
0.45	2342.25	± 224.06	3137	2043

Cuadro 7.2.5: Tabla de valores de la media experimental de iteraciones con $N = 500$.

$\alpha \backslash$ Media	Experimental	Cota superior	Cota inferior
0.10	3245.91	4509.57	2979.023
0.15	3220.09	4407.07	2973.73
0.20	3120.31	4238.65	2901.72
0.25	3062.2	4033.21	2791.94
0.30	2868.83	3803.99	2657.55
0.35	2677.55	3558.29	2505.79
0.40	2516.28	3300.62	2341.09
0.45	2342.25	3034.03	2166.42

Cuadro 7.2.6: Tabla comparativa para la media experimental de iteraciones con $N = 500$.

$\alpha \backslash$ Media	Experimental	Des. Estándar	Máximo	Mínimo
0.10	5139.02	± 454.52	6900	4414
0.15	5100.27	± 389.76	6354	4451
0.20	4957.28	± 370.05	6062	4312
0.25	4753.81	± 369	5772	3954
0.30	4543.67	± 381.94	5851	3807
0.35	4217.87	± 380.15	6442	3590
0.40	3898.51	± 267.74	4774	3373
0.45	3649.80	± 290.85	5030	3133

Cuadro 7.2.7: Tabla de valores de la media experimental de iteraciones con $N = 750$.

$\alpha \backslash$ Media	Experimental	Cota superior	Cota inferior
0.10	5139.02	7311.73	4879.07
0.15	5100.27	7130.82	4849.49
0.20	4957.28	6844.53	4717.51
0.25	4753.81	6501.18	4527.25
0.30	4543.67	6131.73	4305.64
0.35	4217.87	5738.21	4058.66
0.40	3898.51	5315.86	3785.32
0.45	3649.80	4892.60	3505.62

Cuadro 7.2.8: Tabla comparativa para la media experimental de iteraciones con $N = 750$.

$\alpha \backslash$ Media	Experimental	Des. Estándar	Máximo	Mínimo
0.10	7265	± 686.89	10114	6052
0.15	7181.40	± 664.78	10072	5967
0.20	6950.25	± 563.82	8966	6029
0.25	6593.57	± 505.03	9065	5809
0.30	6281.32	± 482.89	8206	5547
0.35	5873.64	± 524.33	7686	4898
0.40	5464.57	± 422.43	6669	4738
0.45	4989.14	± 376.36	6100	4422

Cuadro 7.2.9: Tabla de valores de la media experimental de iteraciones con $N = 1000$.

$\alpha \backslash$ Media	Experimental	Cota superior	Cota inferior
0.10	7265	10266.80	6893.81
0.15	7181.40	9992.49	6831.17
0.20	6950.25	9586.34	6635.23
0.25	6593.57	9106.1	6363.69
0.30	6281.32	8578.40	6042.92
0.35	5873.64	8017.68	5687.39
0.40	5464.57	7433.03	5306.01
0.45	4989.14	6830.54	4904.69

Cuadro 7.2.10: Tabla comparativa para la media experimental de iteraciones con $N = 1000$.

Los resultados experimentales que hemos obtenidos para una cantidad de vértices $N \in \{100, 250, 500, 750, 1000\}$ y desbalance $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45\}$ nos permiten confirmar que la cantidad esperada de iteraciones se encuentra acotada por una cantidad $\Theta(m \ln(mn))$ de iteraciones, siendo solo algunos casos atípicos superiores a nuestras cotas.

Dentro de nuestro estudio también nos interesa analizar como se ajustan estas cotas a media que aumenta la cantidad de vértices N . Para concluir respecto a este último punto, analizaremos que ocurre para un desbalance fijo α y una cantidad de vértices N variable. Nuevamente, presentaremos solo un gráfico para el caso $\alpha = 0.45$, esto debido a que para los otros valores de desbalance α las gráficas presentan un comportamiento similar.

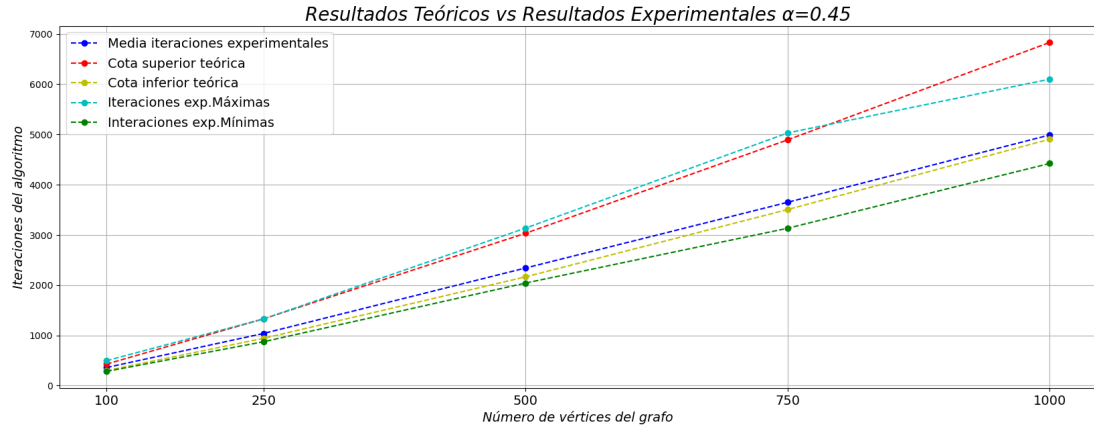


Figura 7.2.2: Media y esperanza teórica de iteraciones para $\alpha = 0.45$: El gráfico presenta en su eje de abscisas los distintos tamaños $N \in \{100, 250, 500, 750, 1000\}$ de los grafos $K_{n,m}$ considerando un desbalance fijo $\alpha = 0.45$. En el eje de las ordenadas se presenta la cantidad de iteraciones del algoritmos. En rojo se marca la cota superior teórica para la esperanza de iteraciones del algoritmo para cada uno de los grafos $K_{n,m}$. En amarillo se marca la cota inferior teórica para la esperanza de iteraciones del algoritmo para cada uno de los grafos $K_{n,m}$. En azul se marca la media de iteraciones obtenida por la implementación computacional para cada uno de los grafos $K_{n,m}$. En verde se marca la mínima cantidad de iteraciones obtenida para cada grafo $K_{n,m}$ y en celeste se marca la máxima cantidad obtenida de iteraciones para cada grafo $K_{n,m}$.

Notamos que la diferencia entre nuestras estimaciones para la cota superior $m \ln(mn)$ de la cantidad esperada de iteraciones para el Algoritmo 5 y los resultados experimentales crece a medida que aumenta la cantidad de vértices. Creemos que esto es debido a las cotas superiores y aproximaciones utilizadas en nuestro análisis, por lo tanto, esperamos que este comportamiento mejore o como mínimo se mantenga al aumentar la cantidad de vértices. Por otro lado, la cota inferior que hemos encontrado parece ser muy precisa, ajustándose cada vez más a la media experimental de iteraciones al aumentar la cantidad de vértices N .

Podemos analizar nuestra media de iteraciones experimental con respecto a los resultados conocidos del problema del coleccionista de cupones. Considerando una colección de $mn(mn - 1)$ cupones y sobres que contienen $2n(nm - 2n)$ cupones distintos con probabilidad uniforme de aparecer cada uno. A continuación presentamos la figura 7.2.3 que compara la media de selecciones necesarias para el problema del coleccionista de cupones con la cota superior $m \ln(mn)$.

Como podemos apreciar la media de iteraciones del problema original del coleccionista de cupones se ajusta mucho mejor a la cota teórica $m \ln(mn)$, siendo levemente superior, esto se debe a que la cota superior es exactamente $(o(1) + m) \ln(mn)$. A partir de esto, dado que la media de iteraciones del Algoritmo 5 es menor a $m \ln(mn)$, podemos conjeturar que la selección de caminos en el Algoritmo 5 disminuye la cantidad de iteraciones en comparación al problema del

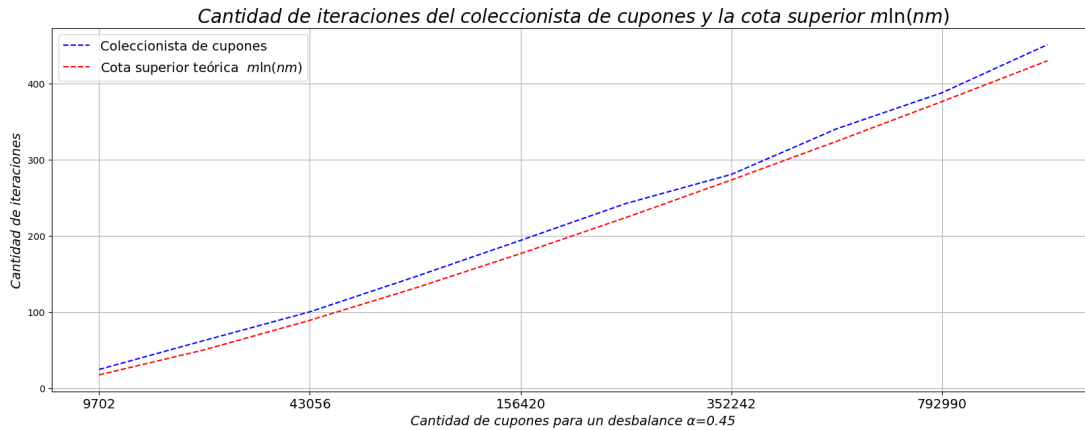


Figura 7.2.3: Gráfica de la media de selecciones necesarias para el problema del coleccionista de cupones y la cota superior $m \ln(mn)$. En azul se presenta la media de selecciones necesarias para el problema del coleccionista de cupones y en rojo la cota superior $m \ln(mn)$.

coleccionista de cupones original.

En conclusión, nuestra cota superior funciona, pero no podemos concluir que sea la mejor posible. Por otro lado, la cota inferior ajusta con mucha precisión el comportamiento del Algoritmo 5. Además, si bien el Algoritmo 5 presenta similitudes con el problema del coleccionista de cupones, podemos notar una considerable mejora en la media de iteraciones respecto a este conocido problema, que deducimos se debe a que los arcos borrados son seleccionados previamente por medio de un camino aleatorio.

7.2.2. Experimentos para la esperanza de aristas no cubiertas $E(N_t)$

La Proposición 6.1.3 nos permitió modelar la esperanza de la cantidad de aristas restantes por cubrir $E(N_t) = (1 - \frac{2}{m})^t mn$ en el grafo después de $t \geq 1$ iteraciones. A continuación compararemos la cantidad de aristas restantes por cubrir $E(N_t)$ en el grafo con los resultados experimentales obtenidos en 250 intentos para la cantidad de aristas restantes por iteración.

Mostraremos nuestros resultados experimentales para una cantidad de vértices $N = 1000$ y $\alpha = 0.45$. Cabe destacar que también se obtuvieron resultados para los casos con $N \in \{100, 250, 500, 750\}$ y $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$, donde para las mismas pruebas se obtuvieron resultados similares, y por lo tanto, se presentan solo los resultados para la mayor cantidad de vértices y aristas que hemos estudiado.

Para $N = 1000$ y desbalance $\alpha = 0.45$ se tiene lo siguiente:

- El primer gráfico muestra la cantidad de aristas restantes por cubrir $E(\mathbf{N}_t) = (1 - \frac{2}{m})^t mn$ y la media de arcos borrados obtenidos en nuestros experimentos.
- El segundo gráfico presenta el error entre la esperanza teórica y la media experimental de aristas restantes por cubrir.
- Ambos gráficos presentan en su eje de abscisas la cantidad de iteraciones del algoritmo.
- El primer gráfico presenta en su eje de ordenadas la cantidad de aristas restantes por cubrir en el grafo en escala logarítmica. Además, se presenta con color azul la media experimental de aristas restantes por cubrir y en color rojo la cantidad teórica estimada.
- El segundo gráfico presenta en su eje de ordenadas la cantidad de error y con color celeste el error entre la media experimental y la esperanza teórica de aristas restantes por cubrir.
- Para ambos gráficos la línea punteada verde indica la cota inferior para la cantidad de iteraciones estimada t_c necesarias para que reste menos de una arista por cubrir por la familia de caminos generada por el algoritmo.

Gráficos para la esperanza teórica $E(\mathbf{N}_t)$ y la media experimental de la cantidad de aristas restantes por cubrir para $N = 1000$ y $\alpha = 0.45$.

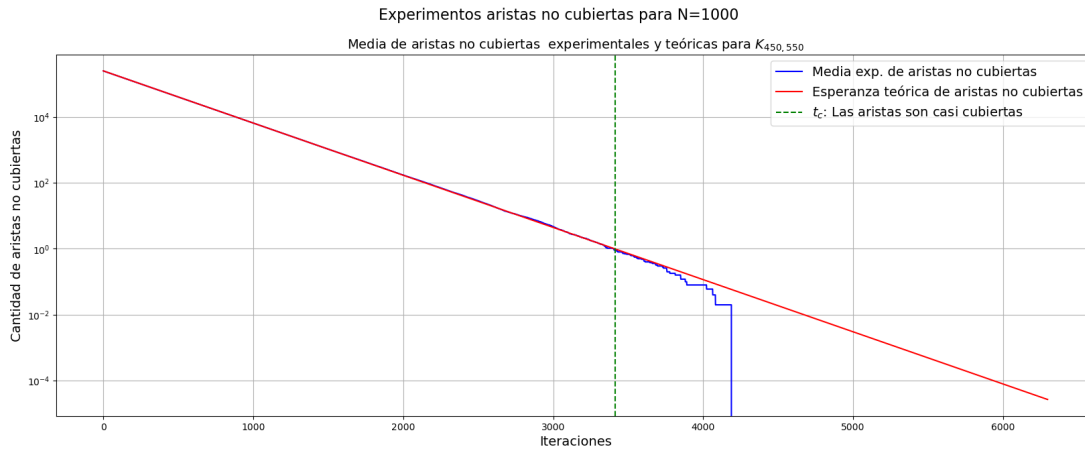


Figura 7.2.4: Media y esperanza de aristas restantes por cubrir para $N = 1000$ y $\alpha = 0.45$: El gráfico muestra la cantidad de aristas restantes por cubrir $E(N_t) = (1 - \frac{2}{m})^t mn$ y la media de arcos borrados obtenidos en nuestros resultados experimentales para una cantidad de vértices $N = 1000$ y desbalance $\alpha = 0.45$. El gráfico presenta en su eje de abscisas la cantidad de iteraciones del algoritmo y en su eje de ordenadas la cantidad de aristas restantes por cubrir en el grafo en escala logarítmica. El gráfico presenta con color azul la media experimental de aristas restantes por cubrir y en color rojo la cantidad teórica estimada. Además, la línea punteada verde indica la cota inferior para la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino.

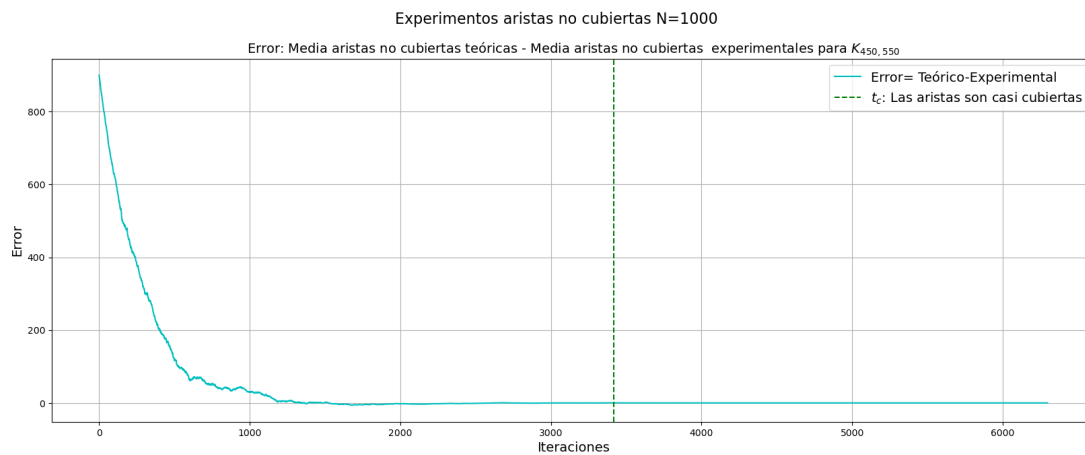


Figura 7.2.5: Error de aristas restantes por cubrir para $N = 1000$ y $\alpha = 0.45$: El gráfico presenta el error entre la esperanza teórica de aristas restantes por cubrir y la media experimental de aristas restantes por cubrir obtenida para $N = 1000$ y $\alpha = 0.45$. El gráfico presenta en su eje de abscisas la cantidad de iteraciones del algoritmo y en su eje de ordenadas la cantidad de error entre la media y la esperanza. El gráfico presenta con color celeste el error entre la media y la esperanza teórica de aristas restantes por cubrir. Además, la línea punteada verde indica la cota inferior para la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino.

Podemos apreciar que nuestra estimación para la cantidad de aristas no cubiertas por algoritmo es muy precisa, donde el error desaparece al cabo de una pequeña cantidad de iteraciones. Además para todos los casos que estudiamos en estas pruebas, la estimación de t_c cumple con lo deseado, estimando el instante en que resta solo una arista por cubrir salvo cierto error. Las medias experimentales presenta un abrupta caída debido a que el eje de las ordenadas para el primer gráfico se encuentra en escala logarítmica, y en general en términos de la media para ciertas cantidades de iteraciones en casi la totalidad de intentos restan exactamente 0 aristas por cubrir el grafo original.

Para más detalles, presentamos para una cantidad de vértices $N = 1000$ y cada desbalance α las siguientes tablas que muestran la cantidad teórica y experimental de aristas que restan por cubrir al cabo de t_c iteraciones

$\alpha \setminus E(N_{t_c})$	Teórica	Experimental	Desviación estándar
0.10	0.99	0.92	± 0.99
0.15	0.99	1.10	± 1.08
0.20	0.98	0.87	± 0.90
0.25	0.98	1.04	± 1.01
0.30	0.98	1.03	± 0.98
0.35	0.98	1.04	± 0.93
0.40	0.97	1.09	± 1.04
0.45	0.97	0.96	± 1.04

Cuadro 7.2.11: Cantidad de aristas restantes por cubrir iteración t_c para $N = 1000$.

7.2.3. Experimentos para la esperanza de arcos borrados

$$\frac{2}{m} \mathbf{E}(M_t)$$

En el Capítulo 6 por medio de la Proposición 6.2.3 modelamos la esperanza de la cantidad de arcos restantes $E(\mathbf{M}_t)$ en el dígrafo después de $t \geq 2$ iteraciones. Equivalentemente podemos estimar la cantidad esperada de arcos borrados en una iteración $t \geq 1$, siendo esta cantidad igual a

$$\frac{2}{m} E(\mathbf{M}_t) = \frac{2}{m} \left(1 - \frac{2}{m}\right)^{t-1} \left(2 \binom{mn}{2} - 2n(mn - 2n)\right).$$

A continuación compararemos esta cantidad $\frac{2}{m} E(\mathbf{M}_t)$ con los resultados experimentales obtenidos en 250 intentos para la cantidad de arcos borrados por iteración.

Mostraremos únicamente nuestros resultados experimentales para una cantidad de vértices $N = 1000$ y $\alpha = 0.45$ debido a las mismas razones del experimento anterior.

Para $N = 1000$ y $\alpha = 0.45$ se tiene lo siguiente:

- El primer gráfico muestra la cantidad de arcos borrados $\frac{2}{m} E(\mathbf{M}_t)$ y la media experimental de arcos borrados obtenida en nuestros experimentos.
- El segundo gráfico presenta el error entre la esperanza teórica de arcos borrados y la media de arcos borrados experimental obtenida.
- Ambos gráficos presentan en su eje de abscisas la cantidad de iteraciones del algoritmo.

- El primer gráfico presenta en su eje de ordenadas la cantidad de arcos borrados en escala logarítmica. Además, el primer gráfico presenta con color azul la media de arcos borrados por la implementación del algoritmo y en color rojo la cantidad teórica estimada.
- El segundo gráfico presenta en su eje de ordenadas la cantidad de error y con color celeste el error entre la media experimental de arcos borrados y la esperanza de arcos borrados teórica.
- Para ambos gráficos la línea punteada verde indica la cantidad de iteraciones estimada t_c . Así mismo, la línea punteada morada de cada gráfico indica la cantidad de iteraciones estimada t_b necesarias para que el algoritmo borre en esperanza menos de un arco por iteración.

Gráficos para la esperanza teórica de la cantidad de arcos borrados $\frac{2}{m}E(\mathbf{M}_t)$ y la media experimental de la cantidad de arcos borrados para $N = 1000$ y $\alpha = 0.45$.

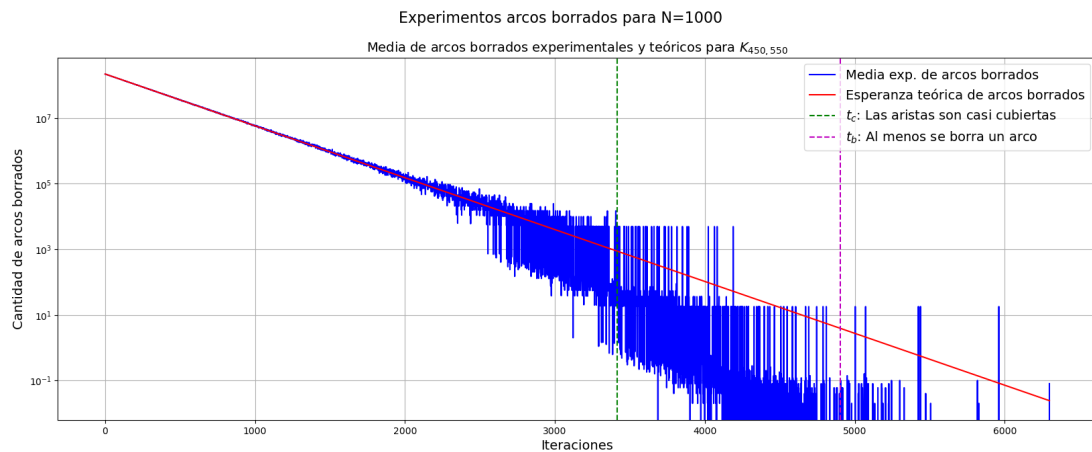


Figura 7.2.6: Media y esperanza teórica de arcos borrados $N = 1000$ y $\alpha = 0.45$: El gráfico muestra la cantidad de arcos borrados $\frac{2}{m}E(\mathbf{M}_t)$ y la media de arcos borrados obtenidos en nuestros resultados experimentales para una cantidad de vértices $N = 1000$ y un desbalance $\alpha = 0.40$. El gráfico presenta en su eje de abscisas la cantidad de iteraciones del algoritmo y en su eje de ordenadas la cantidad de arcos borrados en escala logarítmica. El gráfico presenta con color azul la media experimental de arcos borrados obtenida por la implementación del algoritmo y en color rojo la cantidad teórica estimada. Además, la línea punteada verde indica la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino. Así mismo, la línea punteada morada de cada gráfico indica la cantidad de iteraciones estimada t_b necesarias para que el algoritmo borre en esperanza menos de un arco por iteración.

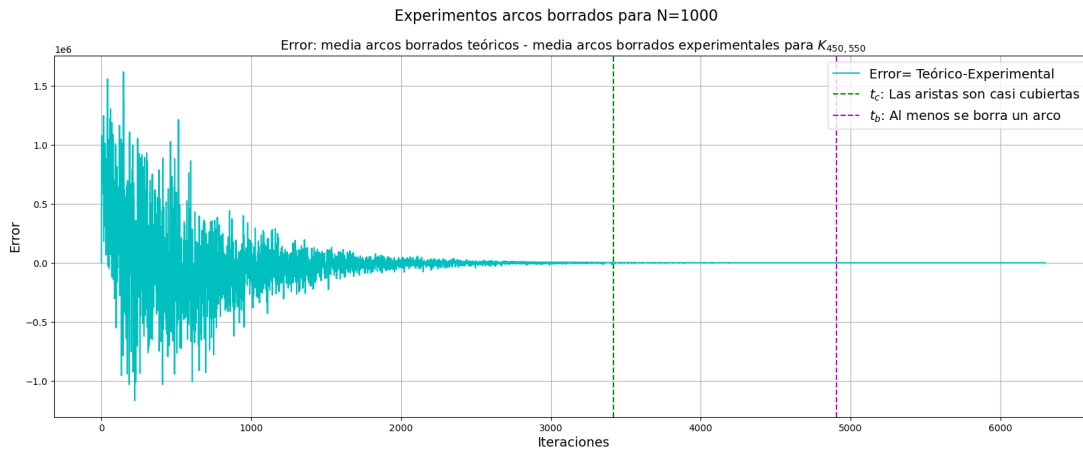


Figura 7.2.7: Error de arcos borrados para $N = 1000$ y $\alpha = 0.45$: El gráfico presenta el error entre la esperanza teórica y la media experimental de arcos borrados obtenida para $N = 1000$ y $\alpha = 0.45$. El gráfico presenta en su eje de abscisas la cantidad de iteraciones del algoritmo y en su eje de ordenadas la cantidad de error entre la media y la esperanza mencionadas anteriormente. El gráfico presenta con color celeste el error entre la media y la esperanza teórica de arcos borrados. Además, la línea punteada verde indica la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino. Así mismo, la línea punteada morada de cada gráfico indica la cantidad de iteraciones estimada t_b necesarias para que el algoritmo borre en esperanza menos de un arco por iteración.

A diferencia de nuestros experimentos para la media de iteraciones de la cantidad de aristas restantes por cubrir, la cantidad de arcos borrados experimentales es mucho más variable, presentando en cada iteración t menor a t_c distintos resultados. Podemos atribuir a este comportamiento a una propiedad de *auto-regulación* de nuestro Algoritmo 5. Suponga que en alguna iteración t la cantidad real de arcos borrados en términos de la media es menor a la estimada por nuestro análisis teórico. En consecuencia, en la siguiente iteración dado que se han borrado menos arcos de los estimados, la cantidad de arcos en D_{t+1} será mayor a lo estimado, y por lo tanto, esperamos borrar más arcos en la siguiente iteración $t+1$ de lo considerado por nuestro análisis. Esto explicaría la variabilidad y el comportamiento en zig-zag mostrados en todas nuestras figuras para este experimento.

Destacamos que en general para iteraciones mayores a la iteración t_c que estima cuando las aristas del grafo son cubiertas casi en su totalidad, esta variabilidad de la media de arcos borrados disminuye significativamente, según como se había previsto anteriormente en nuestro análisis teórico. A continuación presentamos dos tablas para evaluar cuantos arcos se borran en las iteraciones t_c y t_b .

Las tablas nos indican que para algunos desbalances α existe una gran desviación estándar de la media para la iteración t_c , siendo así, la media experimental poco representativa para estos casos. Este comportamiento de grandes desviaciones

$\alpha \setminus E(\mathbf{M}_{t_c}) \frac{2}{m}$	Teórica	Experimental	Desviación estándar
0.10	197.98	4.04	± 25.56
0.15	296.04	1272	± 12646.23
0.20	394.21	12.14	± 68.15
0.25	492.04	25.19	± 108.38
0.30	589.91	2112.25	± 20833.45
0.35	688.60	2393.17	± 22563.29
0.40	784.97	16.62	± 111.640
0.45	880.73	36.39	± 175.82

Cuadro 7.2.12: Cantidad de arcos borrados iteración t_c para $N = 1000$.

$\alpha \setminus E(\mathbf{M}_{t_b}) \frac{2}{m}$	Teórica	Experimental	Desviación estándar
0.10	0.96	0.00	± 0.00
0.15	0.96	0.00	± 0.00
0.20	0.95	0.00	± 0.00
0.25	0.95	0.00	± 0.00
0.30	0.95	0.00	± 0.00
0.35	0.94	0.00	± 0.00
0.40	0.94	0.00	± 0.00
0.45	0.93	0.00	± 0.00

Cuadro 7.2.13: Cantidad de arcos borrados iteración t_b para $N = 1000$.

estándar dependiendo del desbalance α , no fueron una constante en los demás experimentos para los casos $N \in \{100, 250, 500, 750\}$, y por lo tanto, los consideramos como eventos atípicos y no dependientes de la cantidad de vértices N . Para los demás desbalances α con desviaciones estándar mas pequeñas, apreciamos que en general borramos menos arcos de los estimados.

Podemos notar que en la iteración t_b tal y como conjeturamos ya no se borran arcos en términos de la media, más aun, gran parte de los intentos del Algoritmo 5 ya sean detenido para esta iteración t_b . Dado que para esta iteración gran parte de los intentos se han detenido presentamos una media de arcos borrados y una desviación estándar para esta iteración t_b computacionalmente idénticas a 0.

Para poder estimar a cuanto equivale el error que hemos obtenido entre nuestros experimentos y los resultados teóricos para la cantidad de arcos borrados antes de la iteración t_c , presentamos el siguiente gráfico de proporción. Este gráfico presenta en términos de porcentaje la proporción de error que hemos obtenido con respecto a la cantidad máxima de arcos borrados posibles en una iteración $2n(mn - 2n)$ para una cantidad de vértices $N = 1000$ y un desbalance $\alpha = 0.45$.

Para $N = 1000$ y $\alpha = 0.45$ se tiene lo siguiente:

- El gráfico presenta en su eje de abscisas la cantidad de iteraciones realizadas por la implementación del algoritmo.
- El gráfico en el eje de las ordenadas presenta la cantidad de proporción medida en porcentaje de la cantidad máxima posible de arcos borrados $2n(mn - 2n)$.
- El gráfico presenta con color celeste la proporción medida en porcentaje entre el error obtenido por nuestras pruebas y la cantidad máxima posible de arcos borrados $2n(mn - 2n)$.
- En el gráfico se marca con una línea punteada verde la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino. Así mismo, se marca con una línea punteada morada la cantidad de iteraciones estimada t_b necesarias para que el algoritmo borre en esperanza menos de un arco por iteración.

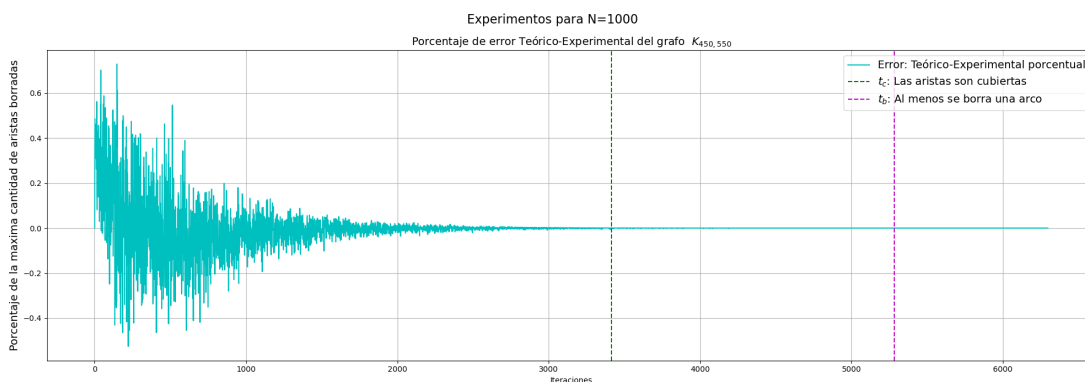


Figura 7.2.8: Error proporcional de arcos borrados experimentales y esperanza de arcos borrados para $N = 1000$ y $\alpha = 0.45$. El gráfico presenta en su eje de abscisas la cantidad de iteraciones realizadas por la implementación del algoritmo para para $N = 1000$ y $\alpha = 0.45$. En el eje de las ordenadas presenta la proporción medida en porcentaje de la cantidad máxima posible de arcos borrados $2n(mn - 2n)$. El gráfico presenta con color azul la proporción mediada en porcentaje entre el error obtenido por nuestras pruebas y la cantidad máxima antes mencionada. Además, la línea punteada verde marca la cantidad de iteraciones estimada t_c necesarias para que todas las aristas salvo una sean cubiertas por un camino. Así mismo, se marca con una línea punteada morada la cantidad de iteraciones estimada t_b necesarias para que el algoritmo borre en esperanza menos de un arco por iteración.

Para todos los casos obtuvimos un error de a lo más del 1% de la cantidad máxima de aristas borradas posibles (esto corresponde a alrededor de un 0.01% de la cantidad total de arcos del digrafo). Notemos que estos gráficos nos permiten apreciar claramente una gran variabilidad de nuestros resultados para las iteraciones menores a t_c , teniendo iteraciones donde nuestra estimación teórica es menor a la experimental o otras iteraciones donde la estimación teórica es mayor a la experimental.

7.2.4. Tiempos de ejecución

Nuestra implementación computacional logra corroborar los resultados teóricos obtenidos en el Capítulo 6 pero sus tiempos de ejecución para un solo intento en la práctica son relativamente largos. Si bien nuestra implementación computacional no busca ser una herramienta real en el área aplicada, mostraremos los tiempos de ejecución de un solo intento para los distintos tamaños N y desbalances α utilizados en nuestras pruebas.

Desbalance α	Media tiempo de ejecución	Desviación estándar
0.10	40.4 [ms]	± 3.9 [ms]
0.15	52 [ms]	± 5.49 [ms]
0.20	67.7 [ms]	± 4.14 [ms]
0.25	85.6 [ms]	± 4.8 [ms]
0.30	103 [ms]	± 6.29 [ms]
0.35	119 [ms]	± 3.73 [ms]
0.40	141 [ms]	± 2.94 [ms]
0.45	161 [ms]	± 8.44 [ms]

Cuadro 7.2.14: Tabla tiempos de ejecución para $N = 100$.

Desbalance α	Media tiempo de ejecución	Desviación estándar
0.10	245 [ms]	± 8.43 [ms]
0.15	416 [ms]	± 25.7 [ms]
0.20	613 [ms]	± 12.6 [ms]
0.25	809 [ms]	± 12 [ms]
0.30	1.07 [s]	± 38.7 [ms]
0.35	1.3 [s]	± 21.3 [ms]
0.40	1.5 [s]	± 18.7 [ms]
0.45	1.71 [s]	± 31.6 [ms]

Cuadro 7.2.15: Tabla tiempos de ejecución para $N = 250$.

Desbalance α	Media tiempo de ejecución	Desviación estándar
0.10	1.33 [s]	± 54.7 [ms]
0.15	2.56 [s]	± 96.9 [ms]
0.20	4.01 [s]	± 108 [ms]
0.25	5.6 [s]	± 249 [ms]
0.30	6.96 [s]	± 81.9 [ms]
0.35	10.3 [s]	± 138 [ms]
0.40	12.2 [s]	± 51.2 [ms]
0.45	13.9 [s]	± 57.1 [ms]

Cuadro 7.2.16: Tabla tiempos de ejecución para $N = 500$.

Desbalance α	Media tiempo de ejecución	Desviación estándar
0.10	3.73 [s]	± 155 [ms]
0.15	6.9 [s]	± 55.5 [ms]
0.20	11.3 [s]	± 74.9 [ms]
0.25	19.5 [s]	± 199 [ms]
0.30	26.3 [s]	± 162 [ms]
0.35	33.1 [s]	± 173 [ms]
0.40	39.8 [s]	± 255 [ms]
0.45	46.2 [s]	± 213 [ms]

Cuadro 7.2.17: Tabla tiempos de ejecución para $N = 750$.

Desbalance α	Media tiempo de ejecución	Desviación estándar
0.10	7.83 [s]	± 91.2 [ms]
0.15	15.5 [s]	± 269 [ms]
0.20	31.1 [s]	± 447 [ms]
0.25	45 [s]	± 313 [ms]
0.30	60 [s]	± 301 [ms]
0.35	77 [s]	± 378 [ms]
0.40	94 [s]	± 302 [ms]
0.45	111 [s]	± 853 [ms]

Cuadro 7.2.18: Tabla tiempos de ejecución para $N = 1000$.

Se puede apreciar que el tiempo de ejecución de la implementación computacional del Algoritmo 5 se ve afectada directamente por la cantidad de aristas del grafo $K_{n,m}$, esto es debido a que el algoritmo constantemente debe actualizar las vecindades del digrafo auxiliar de cada arista en el camino aleatorio seleccionado. El Algoritmo 5 selecciona un camino P en una cantidad $O(n)$ de operaciones. Para cada arista $e \in P$ se borran los arcos correspondientes en una cantidad de

operaciones $O(n^2m^2)$, si bien este análisis es correcto, por medio de un análisis más detallado en la actualización del digrafo podemos conjeturar que la cantidad de operaciones necesarias para borrar los arcos correspondientes son $O(n^2 \log(n))$, de esta forma la cantidad de operaciones por iteración del algoritmo aleatorio es $O(n^2 \log(n))$.

Capítulo 8

Conclusión

En nuestra investigación logramos encontrar cotas inferiores y superiores para $\text{ssp}(K_{n,m})$. Las cotas superiores e inferiores que hemos encontrado dan indicios de la dependencia de $\text{ssp}(K_{n,m})$ con respecto a la diferencia de tamaños entre las partes A y B que componen la partición de los vértices de $K_{n,m}$. Dicho esto, dado que no podemos justificar que estas cotas sean las mejores posibles, aún resta por responder la siguiente pregunta.

Pregunta 1: Sean n, m números naturales. ¿Cuál es el valor de $\text{ssp}(K_{n,m})$?

En relación a esta pregunta, C. B. Wickes [Wic24a] ha estudiado el valor de $\text{ssp}(K_{n,m})$ y el grupo de G. Fernandes, G. O. Mota, y N. Sanhueza-Matamala [Fal+14] a encontrado el valor de $\text{ssp}(K_{n,m})$ salvo un error $o(1)$ para el caso $n = m$.

Luego, como resultado principal logramos definir grafos y digrafos auxiliares que permiten el análisis de los sistemas separadores de una manera alternativa. Por medio de estos digrafos, realizamos un análisis teórico de un algoritmo que encuentra sistemas separadores fuertes de caminos para el grafo $K_{n,m}$ en un cantidad esperada $\Theta(n \ln(mn))$ de iteraciones. Una posterior implementación computacional nos permitió verificar gran parte de los resultados teóricos y plantear nuevas preguntas.

Ahora bien, el Algoritmo 5 fue analizado y posteriormente implementado computacionalmente para el caso del grafo $K_{n,m}$, pero los elementos que componen este algoritmo en general son aplicables a cualquier grafo G . La única problemática real para realizar esta tarea es la selección de un camino aleatorio en un

grafo cualquiera G , que permita un correcto análisis y una implementación razonablemente eficiente. Un candidato a grafo G para realizar un análisis teórico similar y una posterior implementación computacional corresponde a los grafos expansores, ya que la tarea de seleccionar un camino de manera aleatoria en los grafos expansores ha sido ampliamente estudiada y posee buenas propiedades [AS16, Chapter 9.2].

Es importante mencionar que nuestra cota superior de iteraciones estimadas para para el Algoritmo 5 en las pruebas computacionales demostró ser muy superior a la media real obtenida. Esto lo atribuimos a la propiedad de autoregulamiento que conjeturamos que posee el Algoritmo 5, siendo poco representativa la cantidad de arcos borrados $E(\mathbf{M}_t)$ para las iteraciones $t \leq t_c$, y además, a la cotas utilizadas para la estimación de la cota superior realizada en el análisis teórico. Dada esta problemática planteamos la siguiente pregunta

Pregunta 2: *¿Existe una mejor cota superior que $m \ln(mn)$ para la cantidad esperada de iteraciones necesarias para que el Algoritmo 5 aplicado al grafo $K_{n,m}$ se detenga? De ser así, ¿Cual es el valor de esta cota?*

En relación a esta pregunta, nosotros planteamos que solo es posible mejorar esta cota en alguna constante $0 < C < 1$, es decir, la cota superior de la cantidad de iteraciones para que se detenga el Algoritmo 5 es $Cm \ln(mn)$. La media de iteraciones del Algoritmo 5 presentó valores muchos menores a la media experimental del coleccionista de cupones considerando una colección de $mn(mn - 1)$ cupones y sobres que contienen $2n(nm - 2n)$ cupones distintos con probabilidad uniforme de aparecer cada uno. Conjeturamos que este comportamiento se atribuye a que los arcos borrados por el Algoritmo 5 son seleccionados por medio un camino aleatorio previamente seleccionado, reduciendo así la cantidad esperada de iteraciones con respecto al problema original.

Cabe mencionar que en los resultados experimentales jamás se superó, en términos de la media, una cantidad mayor a $19N$, donde N es la cantidad de vértices del grafo $K_{n,m}$. Esta cantidad corresponde a la cota superior de $\text{ssp}(G)$ encontrada por el grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23] para cualquier grafo G en N vértices. Este comportamiento es atribuible a que aún estamos experimentando con cantidades de vértices muy pequeñas para observar cantidades de iteraciones del $m \ln(mn)$.

Enfocándonos en el trabajo algorítmico y computacional realizado, logramos verificar y estudiar la posibilidad de una implementación de algunas estrategias para encontrar sistemas separadores de tamaño menor igual a $\lceil \frac{m}{n} \rceil (11n - 3)$ de manera determinista para el grafo $K_{n,m}$. Para lograr esta tarea hemos utilizado la demostración planteada en el trabajo del grupo de Bonamy, Botler, Dross, Naia, y Skokan [Bon+23] obteniendo resultados satisfactorios. Como trabajo posterior restaría analizar si existen mejores implementaciones en algún otro lenguaje, como por ejemplo C++, que mejore el desempeño que hemos obtenido en esta investigación.

Al igual que para la implementación mencionada anteriormente, sería interesante analizar otra implementación computacional para el Algoritmo 5 en otro lenguaje de programación distinto de Python, que mejore su desempeño. Dentro de las posibles mejoras a nuestro Algoritmo 5, planteamos implementar un proceso de etapas, esto debido a que el desempeño de nuestro algoritmo está muy ligado al momento en que se cubren las aristas del grafo $K_{n,m}$. En base a esto, se podría implementar de manera sencilla una primera etapa que solo busque cubrir todas las aristas por algún camino y luego ejecutar el Algoritmo 5 a partir de ese punto. Además, debido al comportamiento similar a una variable geométrica en la cantidad de arcos restantes en el dígrafo D_t , para la primera iteración t tal que $E(\mathbf{M}_t) \leq \frac{m}{2}$ (Siendo así una cantidad $o(m)$ de arcos), se podría implementar una etapa final donde todos los arcos del dígrafo D_t que aún no se han borrado se agreguen a \mathcal{P}_t como caminos. Esta última etapa es sencilla, puesto que se lleva un conteo de las aristas asociadas a estos arcos en el dígrafo auxiliar, y por lo tanto, se tienen fácilmente identificadas las aristas que restan por semi-separar.

Capítulo 9

Anexo

9.1. Algoritmo determinista

9.1.1. Funciones preliminares

Los siguientes paquetes de Python fueron utilizados para la definición de nuestras funciones:

```
1 import os
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 import math
6 import random
7 from statistics import mean
8 import os, time
```

La Funcion_aristas genera las aristas del subgrafo H' para un grafo $K_{n,m}$ cualquiera. Recuerde que $H' = H \setminus P_S$, donde H corresponde al subgrafo inducido por todas las aristas con algún vértice en el conjunto S y P_S corresponde a todas las aristas salvo la ultima del camino $P = x_1x_2 \cdots x_{2n}x_{2n+1}$.

Listing 9.1: Función aristas H'

```
1 def funcion_aristas(n):
```

```

2     #n: Numero de vertices de la parte A de la particion de
        vertices
3     B=list(range(1,2*n+1,2)) # Particiones A y B de los vertices
        del grafo H
4     A=list(range(2,2*n+2,2))
5     Aristas = []
6     for x in A: #Se generan solo las aristas que pertenecen al
        grafo H'
7         B_2=B.copy()
8         lista1=[x] #
9         if x == 2*n:
10            B_2.remove(x-1)
11        else:
12            B_2.remove(x+1)
13            B_2.remove(x-1)
14            resultado = list(zip(B_2, lista1 * len(B_2)))
15            Aristas=Aristas+[list(a) for a in resultado]
16    return Aristas

```

Después de definir la función que genera las aristas del grafo H' procedemos a definir la Función Caminos_maximos que genera los caminos de largo máximo P_1, \dots, P_l , donde $l = \lceil \frac{m}{n} \rceil$.

Listing 9.2: Función Caminos maximos

```

1     def caminos_maximos(n,m,prop):
2         #n: Numero de vertices de la parte A de la particion de
            vertices
3         #m: Numero de vertices de la parte B de la particion de
            vertices
4         #prop: Corresponde al cajon superior de m/n
5         caminos=[list(range(1+2*n*i,2*n+1+2*n*i+1))
6         for i in list(range (0,prop-1)))]# Se agregan los caminos del 1
            al l-1
7         camino=list(range(2*m-1-2*n-1,2*m)) # Se agrega el camino
            numero l, este camino dependiendo de caso puede repetir

```

```

    vertices
8   caminos.append(camino)
9   return (caminos)

```

La funciones Caminos M_k y Caminos N_k por medio de la lista de aristas generada por la Funcion_aristas genera los conjuntos N_k y M_k , para posteriormente ordenar las vértices y entregar los caminos P_k y Q_k como secuencias de vértices.

Listing 9.3: Función Caminos M_k

```

1  def caminos_M_k (aristas,n):
2  # aristas: Lista de aristas del subgrafo H'
3  # n: Numero de vertices de la parte A de la particion de vertices
4  dicc={} # Diccionario que guardara los caminos M_k
5  for clave in range(5,4*n-1,2):
6      dicc[clave]=[]
7  for k in range(5,4*n-1,2):
8      for a in aristas:
9          if a[0]+a[1]==k:
10             if a[0]<a[1]: # Se guarda en el orden correcto cada
11                 arista e que pertence a algun conjunto M_k
12                 dicc[k].append(a)
13             else:
14                 dicc[k].append([a[1],a[0]])
15  return(dicc)

```

Listing 9.4: Función Caminos N_k

```

1  def caminos_N_k (aristas,n):
2  # aristas: Lista de aristas del subgrafo H'
3  # n: Numero de vertices de la parte A de la particion de vertices
4  dicc={} # Diccionario que guardara los caminos N_k
5  for clave in range(9,6*n-3):
6      dicc[clave]=[]
7  for k in range(9,6*n-3):
8      for a in aristas:

```

```

9         if a[0]<a[1]:
10             if a[0]+2*a[1]==k: # Se guarda en el orden correcto
                cada arista e que pertenece a algun conjunto N_k
11                 dicc[k].append(a)
12         else:
13             if a[1]+2*a[0]==k:
14                 dicc[k].append([a[1],a[0]])
15     return(dicc)

```

Dado que ya hemos definimos las funciones para obtener los caminos P_k 's y Q_k 's, procederemos a definir la función `Descomposicion` que nos permitirá obtener la familia de caminos aristas disjuntos \mathcal{D} que descompone al grafo H' .

Listing 9.5: Función `Descomposicion`

```

1  def descomposicion(n):
2  # n: Numero de vertices de la parte A de la particion de vertices
3      caminos=[] # Lista que guardara los caminos de la
                descomposicion
4  for k in range(0,n):
5      camino = []
6      if n%2==1: # Si es impar se genera la secuencia
                correspondinte de vertices para cada camino Q_i
7          for i in range(1,int((n)/2)+1):
8              camino.append(((k+i)%n)*2+1)
9              camino.append(((k-i)%n)*2+2)
10     else: # Si es par se genera la secuencia correspondinte de
                vertices para cada camino Q_i
11         for i in range(1,int((n)/2)):
12             camino.append(((k+i)%n)*2+1)
13             camino.append(((k-i)%n)*2+2)
14
15         camino.append(((k+int((n)/2))%n)*2+1)
16     caminos.append(camino)
17     return caminos

```

Las funciones que hemos definido ya nos permiten obtener el sistema separador \mathcal{P} que separa fuertemente a H' . Después de obtener el sistema separador \mathcal{P} usaremos las siguientes funciones para modificar los índices de los caminos P'_k s, Q'_k s y obtener caminos análogos pero que pertenecen a los subgrafos H_2, H_3, \dots, H_{l-1} .

Listing 9.6: Función

```

1 def indices_caminos(n,caminos,k):
2     # n: Numero de vertices de la parte A de la particion de
      vertices
3     # caminos : caminos P_K o Q_k generados anteriormente
4     # k: indica que para que sugrafo H_i con i desde 2 a l-1 se
      estan generando los nuevos caminos
5     aumentaB=2*n # Modificacion en los indices de la parte B
6     for key in caminos.keys():
7         for i in range(len(caminos[key])):
8             if caminos[key][i][0] %2==0: # Se respeta el orden i<j
9                 caminos[key][i][1]=caminos[key][i][1]+k*aumentaB
10            else:
11                caminos[key][i][0]=caminos[key][i][0]+k*aumentaB
12    return(caminos)

```

Listing 9.7: Función

```

1 def indices_caminos_l(n,m,caminos):
2     # n: Numero de vertices de la parte A de la particion de
      vertices
3     #m: Numero de vertices de la parte B de la particion de
      vertices
4     # caminos: Caminos P_K o Q_k generados anteriormente
5     for key in caminos.keys(): # Genera los caminos Q_K y N_k para
      el sugrafo H_l
6         for i in range(len(caminos[key])):
7             if caminos[key][i][0] %2==0:
8                 caminos[key][i][1]=caminos[key][i][1]+2*(m-n-1)
9             else:

```

```

10         caminos[key][i][0]=caminos[key][i][0]+2*(m-n-1)
11     return(caminos)

```

De la misma manera modificaremos por medio de las siguientes funciones los índices de los caminos de la descomposición D para encontrar caminos análogos en cada subgrafo H_2, H_3, \dots, H_l .

Listing 9.8: Función

```

1 def indices_caminos_D(n,caminos,k):
2     # n: Numero de vertices de la parte A de la particion de
      vertices
3     # caminos : caminos de la descomposcion D
4     # k: indica que para que sugrafo H_i con i desde 2 a l-1 se
      estan generando los nuevos caminos
5     aumentaB=2*n # Modifacacion en los indices de la parte B
6     for camino in caminos:
7         for i in list(range(0,int(n/2),2)):
8             camino[i]= prop*aumentaB+ camino[i]
9     return(caminos)

```

Listing 9.9: Función

```

1 def indices_caminos_D_l(n,m,caminos):
2     # n: Numero de vertices de la parte A de la particion de
      vertices
3     #m: Numero de vertices de la parte B de la particion de
      vertices
4     # caminos : caminos de la descomposcion D
5     for camino in caminos:
6         for i in list(range(0,int(n/2),2)):
7             camino[i]= 2*m-1-(2*n+1-camino[i])
8     return(caminos)

```

9.1.2. Función Algoritmo determinista

Dadas estas funciones preliminares, definimos la Función Inductiva que implementa computacionalmente el Algoritmo 1.

Listing 9.10: Función inductiva

```

1
2     prop= math.ceil(m/n) # valor del cajon superior de m/n
3     tamaño = 0
4     caminos_de_largo_maximo = caminos_maximos(n,m,prop) # Se
        generan los caminos maximales necesarios
5     aristas = funcion_aristas(n) # Se generan aristas del grafo H
6     des_caminos_disjuntos = descomposicion(n) # Descomposicion D
        del grafo $H$
7     dicc_M_k = caminos_M_k(aristas,n) # Caminos P_k's del grafo $H$
8     dicc_N_k = caminos_N_k(aristas,n) # Caminos Q_k's del grafo $H$
        }tamaño+=len(des_caminos_disjuntos)+len(dicc_m_k)+len(
        dicc_N_k)+len(caminos_de_largo_maximo)+len(caminos_P) # Se
        actualiza el tamaño de sistema
9     # Se desplazan los indices correspondientes para encontrar los
        caminos deseados para cada subgrafo H_2,H_3,...,H_{l-1}
10    descomposicion_general=[indices_caminos_D(n,
        des_caminos_disjuntos,i) for i in range(1,prop-1)]
11    caminos_M_k_index=[indices_caminos(n,dicc_M_k,i) for i in range
        (1,prop-1)]
12    caminos_N_k_index=[indices_caminos(n,dicc_N_k,i) for i in range
        (1,prop-1)]
13    tamaño+=(len(des_caminos_disjuntos)+len(dicc_m_k)+len(dicc_N_k)
        )*(prop-2) # Se actualiza el tamaño de sistema
14    descomposicion_general.append(des_caminos_disjuntos)
15    caminos_M_k_index.append(dicc_m_k)
16    caminos_N_k_index.append(dicc_N_k)
17
18    if m!=n+1: # Si la diferencia entre los tamaño es mayor a 1, es
        necesario generar el subgrafo $H_l$

```

```

19     caminos_m_k_index.append(indices_caminos_l(n,m,dicc_M_k))
20     caminos_N_k_index.append(indices_caminos_l(n,m,dicc_N_k))
21     descomposicion_general.append(indices_caminos_l_D(n,m,
22         des_caminos_disjuntos))
23     tamaño+=len(des_caminos_disjuntos)+len(dicc_M_k)+len(
24         dicc_N_k) # Se actualiza el tamaño de sistema

return ([descomposicion_general,caminos_M_k_index,
        caminos_N_k_index],tamaño+(prop)*(2*n)) # Se retorna el
        sistema separador y su respectivo tamaño

```

9.2. Algoritmo aleatorio

9.2.1. Funciones preliminares:

Los siguientes paquetes de Python fueron utilizados para la definición de nuestras funciones:

```

1 import os
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 import math
6 import random
7 from statistics import mean
8 import os, time

```

La Función `vertices_K_n_m` posee dos parámetros; la cantidad de vértices y el desbalance deseado. Esta función genera dos listas que representan las dos partes que forman la partición de vértices del grafo $K_{n,m}$. Estas dos listas enumeran los vértices de 1 a $|A|$ para la parte A y de 1 a $|B|$ para la parte B (Lineas 10 y 11 del código). Finalmente, la función retorna la cantidad de vértices del grafo, la cantidad de vértices de las partes A y B , las listas de vértices de las partes A y B , y el desbalance.

Listing 9.11: Función vertices $K_{n,m}$

```

1 def vertices_K_n_m(N,alpha=0):
2     # N: Numero de vertices del grafo
3     # alpha: Desbalance del grafo
4
5     if alpha==0: # Si el desbalance no se escoge, entonces se
6         selecciona automaticamente
7         alpha = random.uniform(0.1, 0.5) # Desbalance escogido de
8             manera aleatoria
9
10    Numeros_de_vertices_A = round(alpha * N) # Numero de vertices
11        en la parte A
12    Numeros_de_vertices_B = round((1 - alpha) *N) # Numero de
13        vertices en la parte B
14
15    Parte_menor = list(range(1, Numeros_de_vertices_A+1)) # Lista
16        de la parte A de la particion del grafo bipartito completo
17    Parte_mayor = list(range(1, Numeros_de_vertices_B+1)) # Lista
18        parte B de la particion del grafo bipartito completo
19
20    return N,Numeros_de_vertices_A,Numeros_de_vertices_B,
21        Parte_menor,Parte_mayor,alpha

```

Dado el grafo $K_{n,m}$ con $n \leq m$ y partes $A = \{1, \dots, n\}$ y $B = \{1, \dots, m\}$, procedemos a asignar el siguiente orden para las aristas de $K_{n,m}$. Para cada arista $ij \in E(K_{n,m})$ se le asigna la etiqueta con el numero $(i - 1)(|B| + j - 1)$. Para más detalles considere el siguiente ejemplo: Sea el grafo $K_{2,3}$ de partes $A = \{1, 2\}$ y $B = \{1, 2, 3\}$, entonces las aristas del grafo $K_{2,3}$ corresponden a $\{11, 12, 21, 22, 23\}$ y bajo nuestra asignación poseen las siguientes etiquetas $\{0, 1, 2, 3, 4\}$. Teniendo en cuenta esta asignación, presentaremos la función que genera de manera aleatoria caminos de largo máximo en el grafo $K_{n,m}$.

Listing 9.12: Función camino aleatorio

```

1 def camino_aleatorio(vertices):
2     # vertices: Valores obtenidos por la funcion vertices_K_n_m(
3     # vertices[1]: Cantidad de vertices de la parte A; |A|

```

```

4     # vertices[2]: Cantidad de vertices de la parte B; |B|
5     # vertices[3]: Lista de los vertices de la parte A
6     # vertices[4]: Lista de los vertices de la parte B
7     aristas_camino = [] # Lista vacia que guarda las aristas
      ordenadas del camino
8     parte_a = np.random.choice(vertices[3], vertices[1], replace=
      False) # Permutacion de la lista de la parte A
9     parte_b = np.random.choice(vertices[4], vertices[1] + 1,
      replace=False) # Permutacion de la lista de la parte B
10
11    for i in range(len(vertices[3])): #Las aristas se guardan de a
      pares segun nuestra asignacion en una lista
12        arista_camino.append(int((parte_a[i] - 1) * vertices[2] +
      parte_b[i] - 1))
13        arista_camino.append(int((parte_a[i] - 1) * vertices[2] +
      parte_b[i + 1] - 1))
14
15    return camino_arista

```

La Función `camino_aleatorio` recibe los valores proporcionados por la Función `vértices` $K_{n,m}$ y los utiliza para generar caminos aleatorios de largo máximo. La Función `camino_aleatorio` aplica permutaciones aleatorias a las listas generadas por la Función `vértices` $K_{n,m}$ (líneas 9 y 10), luego, las aristas son seleccionadas por medio de las permutaciones tal y como se presento en el Capitulo 6. Finalmente son guardadas a pares (esto es solo debido a detalles de eficiencia) según la asignación definida anteriormente (líneas 12-14) en una lista llamada `aristas_camino`.

Dado que al agregar un camino podemos semi-separar algunos elementos de $E(K_{n,m})$, debemos actualizar constantemente los arcos del digrafo auxiliar asociado, borrando de ser necesario los arcos correspondientes. Para realizar esta acción, aplicaremos el Lema 5.2, y de esta manera, calcular para cada $e \in E(K_{n,m})$ las aristas que no se encuentran semi-separadas de e por el sistema de caminos generado hasta ese momento por el Algoritmo 5. Dado que realizaremos esta acción múltiples veces definiremos la siguiente función de intersección.

Listing 9.13: Función intersección

```

1  def interseccion(vecinos_no_semiseparadas, arista_del_camino):
2      #arista_del_camino; Arista del camino aleatorio
3      #vecinos_no_semiseparadas: Vecinos de la arista que aun no se
        han semiseparado por el algoritmo
4      set_1 = set(vecinos_no_semiseparadas)
5      set_2 = set(arista_del_camino)
6
7      return list(set_1.intersection(set_2))

```

Dado un camino P obteniendo por medio de la Función `camino_arista`, para cada arista $e \in P$ aplicamos la Función `intersección`. La Función `intersección` recibe una arista y una lista `vecinos_no_semiseparadas` asociada a la arista. La lista `vecinos_no_semiseparadas` corresponde a la lista de aristas que se encuentran aún no semi-separadas de e por el sistema generado por el Algoritmo 5. Luego esta función transforma las listas en arreglos tipo `set` para realizar una intersección entre ellos, se utilizó este tipo de arreglo debido a su eficiencia.

9.2.2. Función Algoritmo aleatorio

Para la implementación del Algoritmo 5 primero ingresamos un camino aleatorio $P^n(\sigma, \tau)$, luego, las aristas $e \in P^n(\sigma, \tau)$ son marcadas como cubiertas por este camino (Línea 23) y generamos un diccionario con las aristas que no se encuentran semi-separadas de e (Línea 24). Luego, el algoritmo se detiene cuando todas las aristas de $K_{n,m}$ fueron cubiertas (Línea 42) y en todos sus diccionarios respectivos solo restan ellas en su interior (Línea 36). De esta manera, todas las aristas se encuentran semi-separadas y por el Lema 5.2.3 se concluye que la familia de caminos encontrada por el Algoritmo 5 es un sistema separador fuerte de caminos de $E(K_{n,m})$.

Listing 9.14: Función Algoritmo secuencial

```

1  def algoritmo_secuencial(vertices, intentos_de_búsqueda):
2      # vertices: Valores obtenidos por la función vertices_K_n_m
3      # vertices[1]: Cantidad de vertices de la parte A; |A|

```

```
4 # vertices[2]: Cantidad de vertices de la parte A; |B|
5 # vertices[3]: Lista de los vertices de la parte A
6 # vertices[4]: Lista de los vertices de la parte B
7 #intentos_de_búsqueda: Cantidad maxima de intentos para
  encontrar un sistema separador fuerte
8 lista_inicial_de_aristas = (np.array(np.zeros(x[1]*x[2]), dtype
  =bool)) # Lista que marca las aristas cubiertas por algun
  camino
9 lista_inicial_de_aristas_semi_separadas = (np.array(np.zeros(x
  [1]*x[2]), dtype=bool)) # Lista que marca las aristas
  semi_separadas
10 diccionario_vecinos = {} # Diccionario que guarda los vecinos
  restantes por semi_separar de cada arista
11 arista_no_semi_separadas= x[1]*x[2] # cantidad de aristas no
  semi-separadas
12
13 for i in range(intentos):
14     camino_por_agregar = caminos_aleatorios_dig(x) # Se escoge
  el camino aleatorio por medio de la funcion
  camino_aleatorio
15
16     for arista in camino_por_agregar: # Actualizamos las
  vecindades de cada arista en el camino
17
18         if not lista_inicial_de_aristas[arista]: # Primera
  aparicion de la arista "e" en el algoritmo
19             lista_inicial_de_aristas[arista] = True # Se marca
  la arista "e"
20             diccionario_vecinos[arista] = np.array(
  camino_por_agregar) # Se agrega el camino como
  vecinos no semi-separados de "e"
21
22         else: # La arista "e" ya ha aparecido anteriormente en
  el algoritmo
23             if not lista_inicial_de_aristas_semi_separadas[
```

```
arista]: # Se pregunta si la arista no esta ya
semiseparada del resto
24 valor_inicial = len(diccionario_vecinos[arista])
    # Cantidad de vecinos aun por semi-separar
25 diccionario_vecinos[arista] = intersection(np.
    array(camino_por_agregar),
    diccionario_vecinos[arista])
26 # se intersecta los vecinos restantes por semi-
    separar con el camino
27
28 if len(diccionario_vecinos[arista]) == 1: # Solo
    resta ella misma en su lista
29 lista_inicial_de_aristas_semi_separadas[
    arista] = True # Se marca la arista como
    semi-separada
30 arista_no_semiseparadas -= 1 # Se actualiza el
    contador de semi_separadas
31 diccionario_vecinos.pop(arista) #Dado que ya
    esta semi_separada borramos su
    informacion del diccionario
32
33 if arista_no_semiseparadas == 0: #si todas
    las aristas estan semi_separadas entonces
    encontramos un sistema separador fuerte
34 return x[0], x[1], x[2], i + 1
35
36 return 'no_separa_en_t+1_adiciones' # Advertencia: la cantidad
    de intentos de busqueda no fueron suficientes.
```

Bibliografía

- [ARK09] Satyajeet S. Ahuja, Srinivasan Ramasubramanian y Marwan Krunz. «Single-link failure detection in all-optical networks using monitoring cycles and paths». En: *IEEE/ACM Transactions on Networking* 17.4 (2009), págs. 1080-1093.
- [Arr+23] Francisco Arrepol, Patricio Asenjo, Raúl Astete, Víctor Cartes, Anahí Gajardo, Valeria Henríquez, Catalina Opazo, Nicolás Sanhueza-Matamala y Christopher Thraves Caro. «Separating path systems in trees». En: *arXiv preprint arXiv:2306.00843* (2023).
- [AS16] Noga Alon y Joel H. Spencer. *The probabilistic method*. Fourth. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, 2016, págs. xiv+375.
- [Bal+16] József Balogh, Béla Csaba, Ryan R. Martin y András Pluhár. «On the path separation number of graphs». English. En: *Discrete Appl. Math.* 213 (2016), págs. 26-33.
- [Bon+23] Marthe Bonamy, Fábio Botler, François Dross, Tássio Naia y Jozef Skokan. «Separating the edges of a graph by a linear number of paths». En: *Advances in Combinatorics* (2023).
- [Bon14] J. Adrian Bondy. «Beautiful conjectures in graph theory.» En: *Eur. J. Comb.* 37 (2014), págs. 4-23.
- [BS07] Béla Bollobás y Alex Scott. «On separating systems». En: *European Journal of Combinatorics* 28.4 (2007), págs. 1068-1071.
- [CS23] Guilherme Oliveira Mota Cristina G. Fernandes y Nicolás Sanhueza-Matamala. «Separating path systems in complete graphs». En: *arXiv preprint arXiv:2312.14879* (2023).
- [De +03] K.M.J. De Bontridder, B.V. Halldórsson, M.M. Halldórsson, C.A.J. Hurkens, J.K. Lenstra, R. Ravi y L. Stougie. «Approximation algorithms for the test cover problem». En: *Mathematical Programming* 98.1-3 (2003), págs. 477-491.
- [Dic69] T.J. Dickson. «On a problem concerning separating systems of a finite set». En: *Journal of Combinatorial Theory* 7.3 (1969), págs. 191-196.
- [Fal+14] Victor Falgas-Ravry, Teeradej Kittipassorn, Dániel Korándi, Shoham Letzter y Bhargav Narayanan. «Separating path systems». En: *Journal of Combinatorics* 5.1-3 (2014), págs. 335-354.

- [Han64] Georges Hansel. «Nombre minimal de contacts de fermeture nécessaires pour réaliser une fonction booléenne symétrique de n variables». En: *CR Acad. Sci. Paris* 258.25 (1964), págs. 6037-6040.
- [Har+07] N.J.A. Harvey, Mihai Patrascu, Yonggang Wen, Sergey Yekhanin y Vincent W.S Chan. «Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs». En: (2007), págs. 697-705.
- [HKL03] Iiro Honkala, Mark G Karpovsky y Simon Litsyn. «Cycles identifying vertices and edges in binary hypercubes and 2-dimensional tori». En: *Discrete Applied Mathematics* 129.2-3 (2003), págs. 409-419.
- [Kat66] Gyula Katona. «On separating systems of a finite set». En: *Journal of Combinatorial Theory* 1.2 (1966), págs. 174-194.
- [KMT01] André Kündgen, Dhruv Mubayi y Prasad Tetali. «Minimal completely separating systems of k -sets». En: *Journal of Combinatorial Theory, Series A* 93.1 (2001), págs. 192-198.
- [Let24] Shoham Letzter. «Separating path systems of almost linear size». En: *Transactions of the American Mathematical Society* 337 (2024), págs. 5583-5615.
- [Lov68] L. Lovász. *On covering of graphs, Theory of graphs (Proc. Colloq., Tihany, 1966)*. 1968.
- [Mat13] Foucaud Florent y Matjaž Kovše. «Identifying path covers in graphs». En: *Journal of Discrete Algorithms* 23 (2013), págs. 21-34.
- [Mau79] Bernard Maurey. «Construction de suites symétriques». En: *CR Acad. Sci. Paris Sér. AB* 288.14 (1979), A679-A681.
- [Rén61] A. Rényi. «On random generating elements of a finite Boolean algebra». En: *Acta Sci. Math. Szeged* 22.75-81 (1961), pág. 4.
- [Ros03] Petri Rosendahl. «On the identification of vertices using cycles». En: *the electronic journal of combinatorics* (2003), R7-R7.
- [RR96] Colin Ramsay y Ian T. Roberts. «Minimal completely separating systems of sets». En: *Australasian Journal of Combinatorics* 13 (1996), págs. 129-150.
- [Spe28] Emanuel Sperner. «Ein satz über untermengen einer endlichen menge». En: *Mathematische Zeitschrift* 27.1 (1928), págs. 544-548.
- [Spe70] Joel Spencer. «Minimal completely separating systems». En: *Journal of Combinatorial Theory* 8.4 (1970), págs. 446-447.
- [TRH12] János Tapolcai, Lajos Rónyai y Pin-Han Ho. «Link fault localization using bi-directional M-trails in all-optical mesh networks». En: *IEEE transactions on communications* 61.1 (2012), págs. 291-300.
- [Weg79] Ingo Wegener. «On separating systems whose elements are sets of at most k elements». En: *Discrete Mathematics* 28.2 (1979), págs. 219-222.
- [Wic24a] Belinda Wickes. «Covering and Separation for Permutations and Graphs». En: *Queen Mary University of London Theses* (2024).
- [Wic24b] Belinda Wickes. «Separating path systems for the complete graph». En: *Discrete Mathematics* 347.3 (2024), pág. 113784.

- [ZK98] Lev Zakrevski y Mark Karpovsky. «Fault-tolerant message routing for multiprocessors». En: Springer. 1998.