



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

ALGORITMOS DE APROXIMACIÓN PARA AUMENTAR LA
VÉRTICE-CONECTIVIDAD DE UN CICLO

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA,
MENCION MATEMÁTICAS APLICADAS
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

FRANCISCO FELIPE SANHUEZA MATAMALA

PROFESOR GUÍA:
JOSÉ SOTO SAN MARTÍN

PROFESOR CO-GUÍA:
JOSÉ CORREA HAEUSSLER

MIEMBROS DE LA COMISIÓN:
IVÁN RAPAPORT ZIMERMANN
WALDO GÁLVEZ VERDUGO

Este trabajo ha sido parcialmente financiado por ANID-PFCHA/Magíster Nacional/2020
- 22201780, CMM ANID PIA AFB170001, Proyecto regular FONDECYT 1181180 y
Proyecto regular FONDECYT 1190043

SANTIAGO DE CHILE
2021

RESUMEN DE LA TESIS PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO Y AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
POR FRANCISCO FELIPE SANHUEZA MATAMALA
FECHA: 2021
PROF. GUÍA: JOSÉ SOTO SAN MARTÍN

ALGORITMOS DE APROXIMACIÓN PARA AUMENTAR LA VÉRTICE-CONECTIVIDAD DE UN CICLO

Los problemas de aumentación de conectividad son un caso particular del *Survivable Network Problem*, que ha sido muy estudiado en las últimas décadas por sus aplicaciones en el diseño de redes robustas. Estos problemas tienen como entrada un grafo G , que es k -vértice-conexo (respectivamente arista-conexo) y un conjunto de links S , tal que $G \cup S$ es $(k + 1)$ -vértice-conexo (respectivamente arista-conexo), y consisten en encontrar el conjunto $F \subseteq S$ de tamaño mínimo tal que $G \cup F$ es $(k + 1)$ -vértice-conexo. Sin embargo, la mayor parte de la investigación sobre problemas de aumentación realizada hasta ahora se ha centrado en el caso de arista-conectividad.

En esta tesis nos enfocamos en encontrar algoritmos de aproximación para aumentar la vértice-conectividad de un ciclo de n vértices C_n , y logramos construir para todo $\varepsilon > 0$ un algoritmo de búsqueda local de complejidad $n^{O(1/\varepsilon)}$ con un orden de aproximación cercano a $1,87029 + \varepsilon$. Para conseguir esto se demuestran resultados novedosos relativos a la estructura de soluciones minimales. Además, se presenta una formulación como programa lineal y una demostración de que el problema es APX-difícil.

Este trabajo se puede ver como una extensión de los resultados de Gálvez *et al.* para el caso en el que se aumenta la arista-conectividad de un ciclo [23]. Nuestros resultados corresponden al primer paso para abordar el caso más general en el que se aumenta la conectividad de un grafo 2-vértice-conexo cualquiera, cuyo mejor orden de aproximación conocido es de 2.

*«Life moves pretty fast. If you don't stop and look around once in a while you could miss it»
Ferris Bueller, Ferris Bueller's Day Off (1986)*

Dedicado a Oscar Agustín Oró Sterki (1994-2014), donde quiera que estés.

Agradecimientos

Siento que este trabajo culmina todo lo que he vivido hasta ahora, por lo que no puedo dejar de agradecer a todas las personas que me han acompañado en este proceso.

Agradezco a mi profesor guía José Soto por su paciencia, soportar mi ritmo de trabajo irregular, corregir mis borradores espantosos y dar consejos y apañe infinito para la realización de esta tesis. Gracias por aportar ideas e interesarse a fondo por el tema, a pesar de no cazar tanto sobre esto en un primer momento. Agradezco también a Waldo, por sus valiosas ideas que fuimos rebotando entre los tres para llegar a este trabajo final. Esta tesis no podría haberla escrito sin ustedes.

También agradezco a José Correa por aportar ideas y buena onda siempre que le comentaba algo y por la carta que escribió para que yo pudiera postular a la beca de magíster. No quiero dejar de mencionar a Iván Rapaport, quien aceptó de buena gana formar parte de la comisión.

Además, no puedo dejar de agradecer a ANID por la beca que me permitió estar más tranquilo mientras realizaba esta investigación. Gracias también a mi hermano Nicolás por ayudarme cuando tenía dudas de grafos, de redacción o cuando tenía que resolver sumatorias feas. Y gracias también a Ignacio Correa, Niko Tapia, Álvaro "Mariscal" Bustos, Rodolfo Gutiérrez y Diego Ramírez por resolver mis dudas de \LaTeX .

Agradezco a la Ale por ser la mejor polilo que podría haber tenido, por acompañarme en este proceso a pesar de no entender las cosas que hacía, por darme amor y cariño, por corregir mi redacción, por las noches de películas y de *Overcooked*, y por las largas charlas que nos hacen crecer como personas. Te adoro. Y por supuesto, gracias al Beni por ser el gato más gordibonito y una luz en esta vida. Alabado seas, Benini.

Agradezco a mi Mamá, por ser la persona que más cree en mí y en lo que puedo hacer a pesar de mis fracasos. Sé que me vas a apoyar en todo lo que haga. Gracias a mi papá por su alegría y su apañe, y a Andrés por tener mucho en común conmigo y por las horas que pasamos hablando de temas que solo a nosotros nos interesan. De nuevo, gracias a Nico, por la música que me ha mostrado. Finalmente, gracias a mi tata Héctor porque la revuelve mucho, a mi abuelita Norma por su sabiduría, al tata *pelao* y a la abuelita Manira, y a mis primes y tíes del lado Sanhueza y Matamala.



Por motivos de espacio, no puedo mencionar aquí a todos los que quisiera, pero pueden leer los agradecimientos completos en el código QR que está a la derecha

Tabla de contenido

Introducción	1
1. Marco teórico	4
1.1. Definiciones	4
1.2. Revisión bibliográfica	6
2. Preliminares	11
2.1. Componentes circulares	14
2.2. Completaciones minimales	18
3. Algoritmo de búsqueda local	23
3.1. Conjuntos críticos	26
4. Análisis del algoritmo de búsqueda local	31
4.1. Refinamiento del análisis	32
5. Un algoritmo de búsqueda local refinado	40
6. Formulación como problema de programación lineal	42
6.1. Primer programa lineal	42
6.2. Segundo programa lineal	43
7. Complejidad del problema	47
Conclusiones	49
Bibliografía	50

Introducción

El objetivo de esta tesis es encontrar algoritmos de aproximación para el problema de aumentar en una unidad la vértice-conectividad de un ciclo. Es decir, dado un ciclo C_n y un conjunto de links S que son cuerdas del ciclo, tales que $C_n \cup S$ es 3-vértice-conexo, hallar el conjunto de links $F \subseteq S$ de tamaño mínimo tal que $C_n \cup F$ sea 3-vértice-conexo.

Motivación

El diseño de una red que cumpla ciertos parámetros de calidad se puede plantear como un problema matemático. La teoría de grafos y la optimización nos son útiles para diseñar redes robustas, lo cual significa que resistan cortes, fallas o ataques. Algunos ejemplos de cómo el concepto de robustez influye en el diseño de redes son los siguientes¹:

- En telecomunicaciones ha sido útil para modelar redes de distribución, como la del agua o la electricidad [2];
- En ecología ha servido para modelar grafos de interacción entre especies, donde cada enlace representa una relación trófica o simbiótica, y la robustez permite ver cuán probable es que se extinga una especie o no [48];
- Se ha usado en economía para modelar una simplificación de las interacciones entre bancos y compradores, donde la robustez de la red indica cuán estable es el sistema financiero ante una crisis en un banco [29].

Una de las métricas más usuales de robustez es el número de vértice-conectividad. Desde los años setenta se han investigado extensamente los problemas de aumento de conectividad y el diseño de redes, tanto por su interés matemático como por sus aplicaciones. El presente trabajo se suma a estos estudios centrados en la vértice-conectividad.

Otro aspecto que puede complejizar estos problemas son las restricciones presupuestarias. Podemos hallar este tipo de restricciones en el informe «Estudio y recomendaciones sobre la resiliencia de la infraestructura del internet chileno» [37], que fue realizado por el laboratorio NICLabs, a petición de la Subsecretaría de Comunicaciones. El objetivo de ese trabajo era proponer diseños de tendidos de fibra óptica resistentes a fallas en todo Chile. En ese caso el grafo inicial era el cableado de internet chileno actual y las aristas que se podían agregar tenían un costo asociado según la distancia y el tipo de cableado existente.

¹Ejemplos extraídos de la página de Wikipedia en inglés *Robustness of complex networks*. Vista el 12 de marzo de 2021.

Como generalmente ocurre, los problemas abordados en el estudio anterior se modelaron de este modo: dado un grafo k -vértice-conexo y un conjunto de links que al añadirlos al grafo lo hacen $(k + 1)$ -vértice-conexo, se busca el conjunto de aristas de menor peso que deje $(k + 1)$ -vértice-conexo al grafo.

Un subcaso importante es cuando los links que se pueden agregar son todos del mismo peso; se demostró recientemente que este subcaso se puede resolver en tiempo polinomial si no hay restricciones en la elección de links [49]. No obstante, cuando hay restricciones (i.e. hay links con peso infinito) el problema es NP-difícil en muchos casos.

En este trabajo estudiaremos cómo aumentar la vértice-conectividad de 2 a 3 en un ciclo, por ser este el grafo 2-conexo más simple. El mejor algoritmo de aproximación para el caso de un grafo 2-vértice-conexo tiene un orden de aproximación de 2 [3], por lo que es interesante abordarlo. Esta tesis puede verse como una extensión del trabajo de Gálvez *et al.* [23] en el que se buscó aumentar la arista-conectividad. Sin embargo, para nuestro caso necesitamos herramientas más sofisticadas que las empleadas por Gálvez *et al.*, quienes emplearon extensamente la contracción de links (i.e. contraer los extremos de un link a un nuevo super-vértice). Estos autores hallaron que, en el caso de arista-conectividad, una solución es factible si y solo si iterativamente la contracción de todos los links transforma el ciclo en un solo vértice. Esta noción no se extiende al caso de vértice-conectividad, por lo que es indispensable crear métodos alternativos para verificar la factibilidad de una solución, desarrollar formulaciones lineales más fuertes y ajustar los parámetros utilizados con más cuidado. Los resultados que presentaremos se elaboraron en conjunto con Waldo Gálvez y José Soto.

Objetivos

Objetivo general

- Encontrar mejores algoritmos de aproximación para el problema de aumentar en uno la vértice-conectividad de un ciclo.

Objetivos específicos

- Estudiar la complejidad del problema de aumentar la vértice-conectividad de un ciclo.
- Estudiar la formulación del problema de programación lineal definido por el problema de aumentar la vértice-conectividad de un ciclo.
- Diseñar mejores algoritmos de aproximación que permitan resolver el problema de aumentar la vértice-conectividad de un ciclo.

Resultados obtenidos

- Diseño de un algoritmo de búsqueda local que utiliza propiedades de soluciones minimales del problema, y cuyo factor de aproximación es $1,87029 + \varepsilon$ para todo

$\varepsilon > 0$.

- Demostración de que el problema es NP-difícil y APX-difícil.
- Formulación del problema como un PL entero con un *gap* de integralidad no restrictivo en instancias clásicas.

Organización de la tesis

La presente tesis está dividida en 7 capítulos.

En el capítulo 1 se muestran las definiciones básicas que se requieren para entender la tesis y se hace un repaso al estado del arte del problema en cuestión. Los capítulos del 2 al 5 corresponden al núcleo de la tesis y describen cómo se usan algoritmos de búsqueda local para lograr el orden de aproximación. Más específicamente, en el capítulo 2 se demuestran resultados preliminares que usaremos para construir nuestro algoritmo. En el capítulo 3 se expone nuestro enfoque algorítmico, para ello, se construye un algoritmo de búsqueda local. En el capítulo 4 se muestran 2 análisis del algoritmo y se demuestra que es posible mejorar el orden de aproximación de 2. El capítulo 5 presenta una extensión del algoritmo para mejorar el orden obtenido en el capítulo anterior.

Si bien los dos últimos capítulos también atienden el problema principal de esta tesis, lo abordan empleando herramientas fuera de los algoritmos combinatoriales y que no se estudian extensamente en el presente trabajo, pero que podrían ser materia para investigaciones posteriores. Así, en el capítulo 6 examinamos dos formulaciones de este problema como un programa lineal entero, para determinar las características de este programa y su *gap* de integralidad, con el objetivo de encontrar cotas inferiores. El capítulo 7 analiza la clase de complejidad del problema a partir del trabajo realizado por Gálvez *et al.* y demuestra que en efecto el problema es NP-difícil y, más aún, que es APX-difícil.

Finalmente, se presentan las conclusiones y las posibles opciones de trabajo futuro.

Capítulo 1

Marco teórico

En este capítulo, se define qué es un grafo, la vértice-conectividad, un algoritmo de aproximación, la relajación de un problema lineal y la clase de complejidad APX, todos conceptos básicos para desarrollar este trabajo. Asimismo, en las siguientes páginas, elaboramos un marco teórico con base en las investigaciones que otros autores han realizado anteriormente en este tema.

1.1. Definiciones

Las definiciones de esta sección provienen del libro *Graph Theory* de R. Diestel [13].¹

Definición 1.1 (Grafo)

Un **grafo** es un par $G = (V, E)$ donde V es un conjunto y E son conjuntos de 2 elementos de V . Los elementos de V se llaman **vértices** y los de E se llaman **aristas** del grafo. Un grafo se dice **dirigido** si cada arista posee una dirección. En otras palabras, si las aristas son pares ordenados.

Si a y b son vértices, las aristas se denotarán como ab , salvo que por claridad sea más cómodo representarlos como tuplas (a, b) .

Definición 1.2 (Grado)

Para un grafo G y un vértice $v \in V$, el grado $d_G(v)$ es igual a la cantidad de vecinos de v en G , es decir, la cantidad de aristas que contengan a v en G .

Definición 1.3

Un **camino** $P = (V, E)$ es un grafo de la forma $V = \{x_0, x_1, \dots, x_k\}$, $E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$, donde los x_i son todos distintos.

¹Todas las traducciones son propias.

Definición 1.4

Un **ciclo** C_n es un grafo cuyos vértices son $V(C_n) = [n] = \{1, 2, \dots, n\}$ y cuyas aristas son $E(C) = \{(i, i+1), 1 \leq i < n\} \cup \{(n, 1)\}$. Es decir, es la unión de un camino y una arista que une el inicio y el fin del camino.

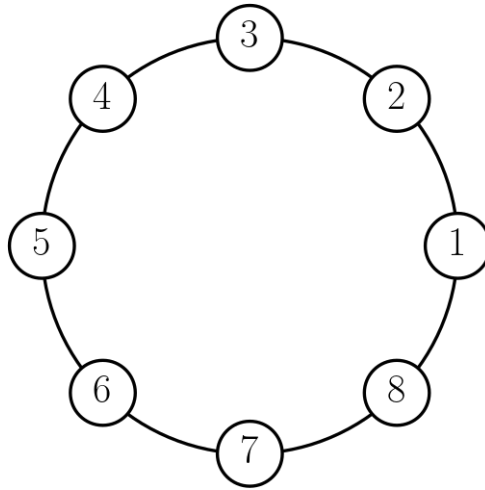


Figura 1.1: El ciclo C_8 con 8 vértices.

Definición 1.5

Un grafo no vacío G se dice **conexo** si todo par de vértices está conectado por un camino en G .

Definición 1.6

Un grafo no vacío se dice **k -vértice-conexo** si $|V(G)| > k$ y el grafo inducido por $V(G) - X$ es conexo para cualquier conjunto $X \subseteq V$ con $|X| < k$. Es decir, si se elimina un conjunto de vértices de tamaño menor a k junto a sus aristas adyacentes, el grafo sigue siendo conexo.

Similarmente, para $|V(G)| > 1$, un grafo se dice **k -arista-conexo** si $G - F$ es conexo para cualquier conjunto $F \subseteq E$ con $|F| < k$.

Definición 1.7 ([52])

Un conjunto de aristas E es un **cubrimiento de aristas** de G si la unión de los extremos de cada arista de E es igual a todos los vértices de G .

Ahora bien, para definir qué es un algoritmo de aproximación recurrimos a *The Design of Approximation Algorithms* de D.P. Williamson y D. Shmoys [54].

Definición 1.8

Un **algoritmo de α -aproximación** para un problema de optimización es un algoritmo en tiempo polinomial que, para todas las instancias del problema, produce una solución cuyo valor está dentro de un factor α del óptimo. Esto quiere decir que, para un problema de minimización, si $f(x)$ es el

valor obtenido para una instancia x , se debe cumplir que $OPT \leq f(x) \leq \alpha OPT$

Estos problemas se modelan como una función que debe ser minimizada pero está sujeta a restricciones; tanto las restricciones como la función son lineales. Si la variable solo toma valores enteros, estamos ante un **problema lineal entero**. Cuando se retiran las restricciones de integralidad a las variables, se trata de una **relajación de un problema lineal**.

Definición 1.9

El **gap de integralidad** de un programa lineal entero de minimización es el máximo valor de la división entre el mínimo para el problema relajado y el mínimo del problema lineal entero entre todas las instancias del problema.

Definición 1.10 ([55])

La clase de complejidad **APX** se define como la clase de problemas de optimización en **NP** que admiten un algoritmo de aproximación de orden constante.

Además, se dice que un problema es **APX-difícil** si existe un $\varepsilon > 0$ tal que encontrar un orden de aproximación de $1 + \varepsilon$ es **NP-difícil** [33].

1.2. Revisión bibliográfica

En esta sección, examinaremos los resultados de diversos autores que han investigado problemas similares al de esta tesis. Esta revisión empezará con los casos más generales, luego atenderá los más específicos y terminará con resultados relacionados al aumento de la conectividad de un ciclo. Seguiremos la estructura y notaciones del *survey* de Nutov y Kortsarz [35]. Para construir la clase general del tipo de problemas que abordaremos, necesitamos primero definir la **S-conectividad**.

Definición 1.11 (S-conectividad)

Sea $G = (V, E)$ un grafo, $S \subseteq V$ un subconjunto de vértices, y $u, v \in V$ se define la **S-conectividad** $\lambda_G^S(u, v)$ de (u, v) en G como el máximo número de caminos entre u y v aristas disjuntos que no comparten vértices de $S \setminus \{u, v\}$.

Instancia: un grafo $G = (V, E)$ (que puede ser dirigido) con costos en las aristas $\{c_e : e \in E\}$. Un subconjunto de vértices $S \subseteq V$ y una función no negativa de requerimientos $r(u, v)$ definida en $V \times V$.

Objetivo: encontrar un subgrafo $G' = (V, E')$ de costo mínimo que cubra todos los vértices tal que cumpla que

$$\lambda_{G'}^S(u, v) \geq r(u, v) \text{ para todo } u, v \in V. \quad (1.1)$$

Observación Sin pérdida de generalidad, se puede considerar a G como un grafo completo, asignándole costo infinito a las aristas no disponibles. Además, cuando $S = \emptyset$, al problema se le denomina Arista-SND y cuando $S = V$, se le llama Vértice-SND.

Network augmentation problem

Los subproblemas de *Survivable Network Problem* se pueden clasificar de acuerdo con el valor de los costos de las aristas, el conjunto S y la función de requerimientos. Dentro de estos subproblemas, hay uno que se asemeja al que abordaremos en esta tesis, por lo que creemos que es importante mencionarlo. Lo definimos a continuación:

NETWORK AUGMENTATION PROBLEM

Instancia: un grafo $G = (V, E)$ (que puede ser dirigido) y una función no negativa de requerimientos $r(u, v)$ definida en $V \times V$.

Objetivo: encontrar un conjunto mínimo de aristas F tal que $G \cup F$ cumpla con (1.1).

Notemos que este problema es un caso particular del anterior. Ahora los costos de las aristas están en el conjunto $\{0, 1\}$. Si la función de requerimientos es constante e igual a k para todo $u, v \in V$, el problema para $S = \emptyset$ equivale a encontrar un conjunto mínimo tal que el grafo resultante sea k -vértice-conexo (respectivamente, si $S = V$, arista-conexo). Para este último caso, tanto si queremos aumentar la arista o vértice-conectividad como si el grafo es dirigido o no, se conocen los siguientes resultados.

Por una parte, el problema de aumentar la arista-conectividad en grafos no dirigidos fue resuelto por Watanabe y Nakamura en 1987 [51] con un algoritmo polinomial de orden $O(kL|V|^4(k|V| + |E|))$, donde $L = \min\{k, |V|\}$. Por otra parte, en 1992, A. Frank resolvió cómo aumentar la arista-conectividad en grafos dirigidos para cualquier función de requerimientos $r(u, v)$ [19].

Por otro lado, en 1995, Frank y Jordan desarrollaron un método para aumentar la vértice-conectividad en grafos dirigidos [18] empleando un algoritmo que dependía del método de la elipsoide. En 1999 ambos diseñaron un algoritmo combinatorial polinomial en número de vértices pero exponencial en la conectividad k para encontrar la aumentación mínima [20]. Posteriormente, Vegh creó dos algoritmos combinatoriales en tiempo polinomial para

resolver el problema de aumentar la vértice-conectividad en grafos dirigidos [50].

En contraste, los resultados obtenidos para aumentar la vértice-conectividad en grafos no dirigidos son escasos, y solo se ha resuelto para casos particulares. En este sentido, para aumentar en uno la vértice-conectividad óptimamente, Végh demostró una fórmula min-max y diseñó un algoritmo combinatorial de tiempo polinomial [49].

Además, en 2017, Durand de Gevigney y Szigeti resolvieron el caso específico de aumentar de forma óptima un grafo no dirigido a uno $2k$ -arista-conexo con k par, tal que al sacar cualquier arista queda un grafo k -arista-conexo [25]. Un algoritmo posterior fue descrito por Hoersch y Szigeti [30].

Ahora bien, cuando se agregan las aristas de costo infinito, muchos de los *network augmentation problems* se convierten en NP-completos, lo cual complica su resolución.

Para el caso general de aumentar la arista-conectividad a una conectividad arbitraria cuando hay restricciones en el uso de las aristas y se tienen pesos generales, la mejor aproximación que se conoce es de orden 2, obtenida por diversos métodos y autores [26, 32]. Por mencionar algunos, en 1995, Williamson *et al.* probaron este orden a través de un algoritmo primal dual [53]. En 2001, Jain [31] demostró este orden para el caso general de un *Survivable Network Problem*, introduciendo la técnica del **redondeo iterativo**. En 2020, Garg y Kumar obtuvieron que una versión ligeramente modificada del PL de Williamson *et al.*, tiene un problema dual *half-integral* cuyo valor es al menos la mitad del valor del problema primal [24]. Solo se ha logrado disminuir este factor de 2 en algunos casos específicos [1, 12, 17, 44].

Con el fin de entender cómo aumenta el orden de aproximación cuando cambiamos aristas por vértices, consideremos el problema de *Survivable Network Design* en el que existen requerimientos de vértice-conectividad. En el caso general del *Survivable network design*, no es posible aproximarse por un orden mejor que $2^{\log^{1-\epsilon} n}$, a menos que $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ [33]. No obstante, para una conectividad k fija, hay un algoritmo de orden de aproximación $O(k^3 \log(n))$ [11].

Algoritmos parametrizados

Otra línea de investigación relacionada a este problema es el estudio de algoritmos parametrizados, los cuales consisten en el diseño de algoritmos, que se ejecutan en tiempo $f(p)n^{O(1)}$, para un cierto parámetro p . Cuando es posible construir este tipo de algoritmo, decimos que el problema es tratable de parámetro fijo (FPT por sus siglas en inglés). En el caso particular de *network augmentation problems*, el parámetro p acota el número de aristas que podemos agregar a nuestro grafo inicial. A continuación veremos algunos ejemplos de estudios sobre algoritmos parametrizados.

Gracias a los resultados de la investigación realizada por Nagamochi en 2003, se sabe que el problema de aumentar la arista-conectividad de 1 a 2 con la menor cardinalidad posible es FPT [40]. En 2010, Guo y Uhlmann demostraron que tanto este problema como la versión vértice-conectividad, admiten un kernel (es decir, una instancia equivalente de tamaño polinomial en p , pero que es independiente de los otros parámetros del problema)

de tamaño $O(p^2)$ [28].

Asimismo, en 2015, Marx y Vegh diseñaron un algoritmo de tiempo $2^{O(p \log p)} n^{O(1)}$ para el caso en que se busca aumentar la arista-conectividad de k a $k + 1$ para un k arbitrario usando el menor peso posible. Además, encontraron un algoritmo parametrizado para aumentar un grafo arbitrario a ser 2-vértice-conexo con peso mínimo [39]. El tiempo de ejecución para el problema de aumentar arista-conectividad de k a $k + 1$ fue mejorado a $9^p n^{O(1)}$ poco después por Basavaraju *et al.* [4].

k-connectivity augmentation problem

Hasta ahora, en los problemas que hemos definido en las subsecciones anteriores, la conectividad objetivo era arbitraria. ¿Qué ocurre en el caso particular de aumentar en uno la conectividad de nuestra red? Este problema puede definirse de la siguiente manera:

K-CONNECTIVITY AUGMENTATION PROBLEM

Instancia: un grafo $G_0 = (V, E_0)$ k -vértice-conexo (respectivamente k -arista-conexo) y un conjunto S de aristas extras entre los vértices de G_0 , tal que $G_0 \cup S$ sea $(k + 1)$ -vértice-conexo (respectivamente $(k + 1)$ -arista-conexo).

Objetivo: encontrar un conjunto de aristas $F \subseteq S$ de mínima cardinalidad, tal que $G_0 \cup F$ sea $(k + 1)$ -vértice-conexo (respectivamente $(k + 1)$ -arista-conexo).

Diversos investigadores han trabajado en el *k*-connectivity augmentation problem usando grafos pertenecientes a clases específicas como G_0 , con el objetivo de mejorar el factor de aproximación de 2. En cuanto a la arista-conectividad, se estudian exclusivamente los casos $k = 1$ (cómo los árboles) y $k = 2$ (cómo los grafos cactus, es decir, grafos conexos en los cuales cada arista pertenece a un solo ciclo)², pues Dinitz *et al.* demostraron [14] que el problema de *k*-edge-connectivity augmentation se reduce a $k = 1$ si k es impar o a $k = 2$ si es par.

El caso más estudiado es el de aumentar en uno la arista-conectividad de un árbol. Se sabe que este problema es APX-difícil [9, 22, 33, 40]. A lo largo de los años, muchos autores utilizaron distintas técnicas para disminuir el orden de aproximación de 2 [1, 7, 8, 15, 17, 34, 36]. Dentro de los avances más recientes, en 2019, Grandoni *et al.* lograron una aproximación con un orden de 1,458 [27], lo que es mejor que el *gap* de integralidad de 1,5 del Problema de programación lineal estándar [7].

Por otra parte, para el caso de aumentar en uno la arista-conectividad de un cactus, en 2019, Byrka *et al.* publicaron un algoritmo con un orden de aproximación cercano a 1,91 [5], reduciendo el problema a encontrar un mínimo *Steiner tree* para un grafo específico. A finales de 2020, Ceccheto *et al.* aplicaron las técnicas desarrolladas por Grandoni *et al.* para un árbol y mejoraron sustancialmente el orden a 1,393 para $k = 1$ y $k = 2$ [6]. Igualmente,

²El caso $k = 1$ no es exactamente igual al problema de aumentar la arista-conectividad de un árbol, pero cada componente 2-arista-conexa de G_0 se puede comprimir a un solo vértice, obteniendo un árbol, por esto se supone que la entrada es un árbol. Para $k = 2$, por motivos similares, se puede suponer que G_0 es un cactus.

gracias a la reducción de Dinitz *et al.*, los resultados de Ceccheto *et al.* mejoran el orden de aproximación del problema de aumentar la arista-conectividad de k a $k + 1$ para cualquier k .

En lo que se refiere a aumentar en uno la vértice-conectividad de un árbol, en 2020, Nutov probó, con base en el resultado de Byrka *et al.* [5], que existe un algoritmo de aproximación de orden cercano a 1,91 [41]. Esto superó el orden de 2 descubierto en 1981 por Frederickson y Ja'Ja [22].

Para el caso de aumentar de 2 a 3 la vértice-conectividad en un grafo con pesos, en 1999, Auletta *et al* introdujeron un algoritmo de aproximación de orden 2 [3]. Una particularidad de este algoritmo es que usa un teorema de Frank y Tardos como caja negra. Este último permite encontrar en tiempo polinomial un grafo dirigido k -out-connected de peso mínimo desde un vértice dado [21]. En 1997, Ravi y Williamson propusieron un algoritmo para encontrar una completación con peso mínimo, que aumentase la vértice-conectividad de un grafo a k con un orden de aproximación de $2H(k)$ (donde $H(k)$ es el número armónico de orden k) [46]. Sin embargo, en 2002, encontraron un error [47]. Este resultado, aunque errado, es interesante porque usa algoritmos poliedrales. Posteriormente, en 2003, Cheriyan *et al.* diseñaron un algoritmo de aproximación de orden $6H(k)$, que funciona cuando el número de vértices es mayor a $6k^2$ [10]. En 2017, Nutov mejoró el orden de aproximación a 4 para el caso general de aumentar la vértice-conectividad de k a $k + 1$ [43].

Por último, en 2019, Gálvez *et al.* estudiaron cómo aumentar en uno la arista-conectividad de un ciclo y describieron dos algoritmos: el primero mejora a $\frac{5}{3}$ la 2-aproximación de Jain y el segundo, a $\frac{3}{2} + \epsilon$. Además, estos autores escribieron una formulación como problema lineal con *gap* de integralidad $\frac{3}{2}$ y demostraron que el problema es APX-difícil [23].

Antes de concluir este capítulo, cabe destacar la complejidad que adquiere el problema de aumentar en uno la conectividad de un ciclo cuando no todas las cuerdas pueden ser agregadas. En el caso en que no existen estas restricciones, el problema puede solucionarse de manera fácil (tanto para arista-conectividad como vértice-conectividad) si se añade un *matching*.

Capítulo 2

Preliminares

En este capítulo definiremos el problema y presentaremos resultados preliminares de las estructuras a usar y del tamaño de una solución minimal.

Desde ahora en adelante, todos los grafos serán simples (i.e. sin aristas paralelas). Supondremos que $n \geq 4$. Debido a la estructura cíclica de C_n , podemos referirnos al vértice 1 como $n + 1$. Llamaremos **cuerda** a una arista entre 2 vértices no consecutivos cíclicamente en C_n . Cada cuerda ab divide al ciclo en dos **lados** no vacíos, V_1 y V_2 , tal que V_1, V_2 y $\{a, b\}$ particionan los vértices del ciclo. Si F es un conjunto de cuerdas, usaremos $V(F)$ para denotar al conjunto de vértices cubiertos por al menos una cuerda de F . Como el ciclo es 2-vértice-conexo, también es conexo cualquier grafo obtenido si se le agregan cuerdas al ciclo.

Definición 2.1 (Par separador)

En un grafo 2-vértice-conexo, un **par separador** son dos vértices tales que al sacarlos junto a sus aristas adyacentes, el grafo se divide en 2 o más componentes conexas. Por lo tanto, los grafos 3-vértice-conexos son exactamente los grafos 2-vértice-conexos de al menos 4 vértices sin pares separadores.

Diremos que un conjunto F de cuerdas 3-conecta un ciclo, si $C_n \cup F$ es 3-vértice-conexo. Entonces, el problema que estudiaremos se puede definir de la siguiente forma.

CYCLE VERTEX-CONNECTIVITY AUGMENTATION PROBLEM (CVCA)

Instancia: un ciclo de n vértices C_n y un conjunto de cuerdas S tales que $C_n \cup S$ es 3-vértice-conexo. Por comodidad podemos referirnos a la instancia como un par ordenado (C_n, S) .

Objetivo: encontrar un conjunto de cuerdas $F \subseteq S$ de mínima cardinalidad tal que $C_n \cup F$ sea 3-vértice-conexo.

Observación Otra forma de ver el problema es notar que las aristas del ciclo tienen costo

0; las que se pueden agregar, costo 1 y el resto, costo infinito. En ese caso, el objetivo es buscar $F \subseteq \binom{V(C_n)}{2} = \{U \subseteq V(C_n) : |U| = 2\}$ de peso mínimo.

De ahora en adelante, llamaremos solución factible a cualquier conjunto $F \subseteq S$ que 3-conecte al ciclo. Además, llamaremos *links* a las cuerdas en S . En el capítulo 7 veremos que el problema es APX-difícil, por lo que nos enfocaremos en encontrar algoritmos de aproximación.

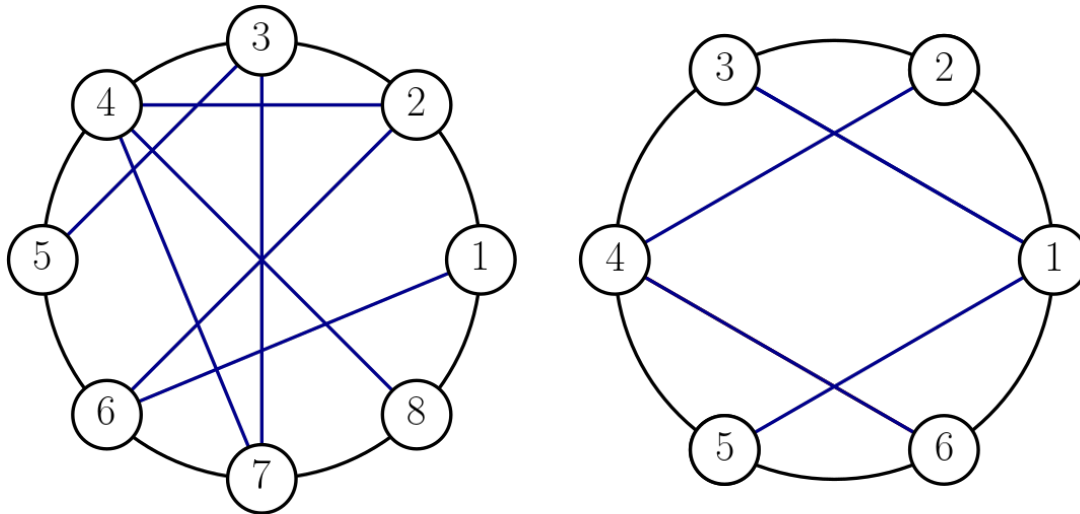


Figura 2.1: **Izquierda:** ejemplo de instancia: las aristas y vértices negros son el ciclo C_8 , y los links azules son el conjunto S . **Derecha:** ejemplo de instancia que es 3-arista-conexa pero no 3-vértice-conexa, pues $(1, 4)$ es un par separador.

Una definición que nos permite determinar cuándo un ciclo se convierte en 3-vértice-conexo tras agregarle aristas, es la siguiente:

Definición 2.2 (Cruzar)

Sean a, b, c, d cuatro vértices distintos de C_n , sin pérdida de generalidad $a < b$ y $c < d$ en el orden usual en $[n]$, decimos que la cuerda ab **cruza** a cd si $c < a < d < b$ o $a < c < b < d$.

Equivalentemente, ab y cd se cruzan si y solo si c y d están en distintos lados de ab . Observemos que, de acuerdo con esta definición, dos cuerdas que comparten un extremo no se cruzan.

La noción de cruzar es natural: si dibujamos C_n en un plano como un círculo y ab y cd son líneas rectas, entonces ab y cd se cruzan si las líneas se intersectan en su interior.

Lema 2.3

F es una solución factible si y solo si toda cuerda de C_n es cruzada por un link de F .

DEMOSTRACIÓN. Por la estructura del ciclo, cualquier par de vértices consecutivos de C_n no forma un par separador de $C_n \cup F$. Luego, se tiene el resultado, porque toda cuerda de C_n

es un par separador de $C_n \cup F$ si y solo si no es cruzada por un link de F . \square

Lema 2.4

Toda solución factible tiene tamaño al menos $n/2$.

DEMOSTRACIÓN. Notemos que si j es un vértice de C_n , F debe tener una arista e que cruza la cuerda $(j - 1)(j + 1)$ de C_n y esta debe ser incidente a j . En consecuencia, F es un cubrimiento por aristas de V , de donde se deduce la cota. \square

Por lo tanto, tiene sentido estudiar soluciones factibles minimales para el CVCA, dado que cualquier conjunto de links que contenga una solución factible es factible. Una forma fácil de construir un algoritmo de 2-aproximación se extrae del siguiente teorema de Mader.

Definición 2.5

Una arista en un grafo k -vértice-conexo es crítica si el grafo deja de ser k -vértice-conexo al eliminarla.

Lema 2.6 (Mader [38])

Sea G k -vértice-conexo y $C' \subseteq G$ un ciclo compuesto solo de aristas críticas, entonces existe $v \in V(C')$ tal que $d_G(v) = k$.

Lema 2.7

Un conjunto minimal F de aristas factible no puede contener ciclos. Por lo tanto, todo conjunto minimal que haga 3-vértice-conexo al ciclo es una 2-aproximación para el problema.

DEMOSTRACIÓN. Si tenemos un ciclo C' en $C_n \cup F$, todos los vértices de ese ciclo tienen grado 4. El lema 2.6 nos indica que al menos una arista de C' no es crítica, lo que contradice la minimalidad de F . Luego podemos eliminar una arista del ciclo C' . Por consiguiente, como es acíclico, F tiene tamaño a lo más de $n - 1$. Usando el lema 2.4, la solución óptima tiene tamaño de al menos $n/2 \geq |F|/2$. \square

Por ende, podemos definir un algoritmo de 2-aproximación de esta manera:

Algoritmo 1 Algoritmo de 2-aproximación

- 1: $F \leftarrow S$
 - 2: **mientras** exista link e tal que $F - e$ es factible **hacer**
 - 3: $F \leftarrow F - e$
 - 4: **fin mientras**
 - 5: **devolver** F
-

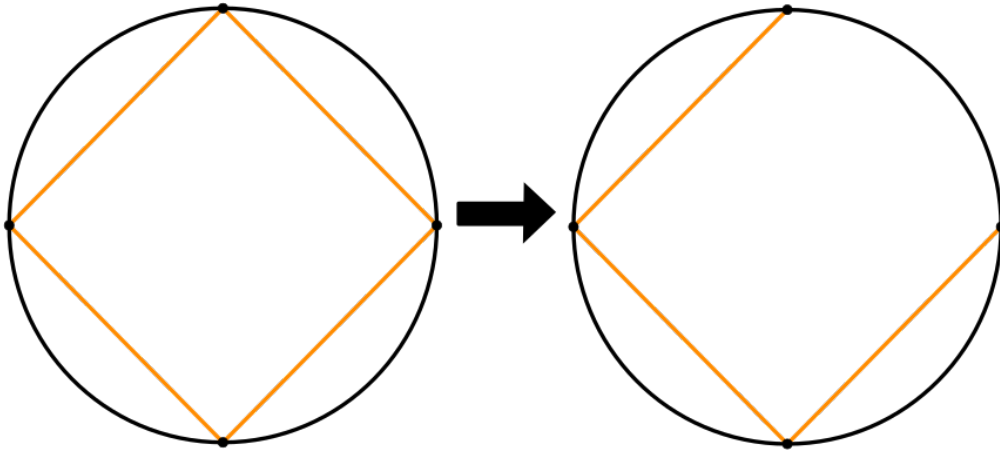


Figura 2.2: Como una solución minimal del problema no tiene ciclos, es posible eliminar una arista del ciclo.

2.1. Componentes circulares

Para obtener un mejor orden de aproximación, en el capítulo 3, desarrollaremos un algoritmo que construye la solución de forma *bottom-up* en vez de *top-down*. El primer paso para lograrlo es definir el concepto de componente circular. A su vez, para entender qué es una componente circular, precisamos definir grafo circular, con base en la propuesta de Even e Itai [16].

Definición 2.8 (Grafo circular)

Un **grafo circular** definido por un conjunto de cuerdas F es el grafo con conjunto de vértices F , tal que dos cuerdas son adyacentes si y solo si se cruzan.

Definición 2.9 (Componente circular)

Una **componente circular** es un conjunto de links $L \subseteq S$ tal que $|L| \geq 2$ y el grafo circular definido por L es conexo. Un **camino de links**, desde un vértice a a un vértice b , es una secuencia $P = e_1 e_2 \dots e_k$ de links tal que cada link cruza al siguiente, a es un extremo de e_1 y b es un extremo de e_k .

Notemos que no se permiten componentes circulares singletons ni vacías. También notemos que, si L es una componente circular, entonces para todo $a, b \in V(L)$ existe un camino de links en L desde a hasta b . El teorema principal de esta subsección, cuya definición de solución factible será la que usaremos a lo largo del presente trabajo, es el siguiente:

Teorema 2.10

Sea $F \subseteq S$ un cubrimiento de aristas de $C_n \cup S$, el conjunto F es una solución factible para CVCA si y solo si F es una componente circular.

Necesitamos un par de propiedades y definiciones sobre cuerdas y componentes circulares antes de demostrar el teorema.

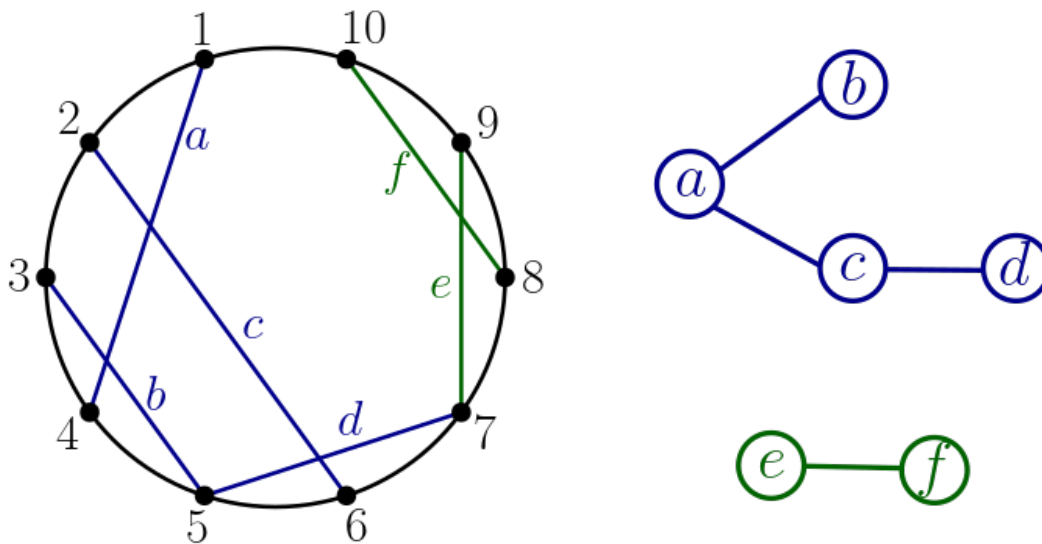


Figura 2.3: **Izquierda:** dos componentes circulares en un grafo, marcadas con las aristas azules y verdes. $\{1, 7\}$ y $\{7, 10\}$ son las únicas aristas de borde de las componentes azul y verde, respectivamente. La secuencia de aristas bac es un ejemplo de camino de links entre 5 y 2. **Derecha:** su representación como grafo circular.

Definición 2.11

Sea L una componente circular y $V(L) = \{v_1, v_2, \dots, v_{|V(L)|}\}$ tal que $v_1 < v_2 < \dots < v_{|V(L)|}$ en el ciclo, decimos que v_i, v_{i+1} (donde los índices son módulo $|V(L)|$) son **consecutivos** en $V(L)$.

Las cuerdas entre vértices no consecutivos de $V(L)$ son **cuerdas internas** de L y las cuerdas entre vértices consecutivos de $V(L)$ son **cuerdas de borde**. Los vértices de $V(L)$ incidentes a una cuerda de borde los llamamos **vértices de borde** y el resto de los vértices de $V(L)$ son **vértices internos**. Los vértices fuera de $V(L)$ son **vértices externos**.

Una cuerda interna de L puede conectar dos vértices internos, un vertice interno y uno de borde o incluso dos vértices de borde (si es que son incidentes a dos cuerdas de borde diferentes). Decimos que L **crusa** a una cuerda ab de C_n si algún link $e \in L$ cruza a ab . El lema siguiente caracteriza a los links cruzados por una componente.

Lema 2.12

Sean ab y cd cuerdas de C_n y $P = e_1 \dots e_k$ un camino de links entre a y b , si ab cruza cd , entonces hay alguna cuerda de P que cruza cd .

DEMOSTRACIÓN. Sean V_1 y V_2 los conjuntos de vértices a cada lado de la cuerda cd (de modo que $[n] = V_1 \cup V_2 \cup \{c, d\}$) tal que $a \in V_1$ y $b \in V_2$, notemos que $V_1 \cup \{c, d\}$ es un intervalo en el ciclo con extremos c y d . También consideremos a e_i como la primera cuerda de P con al menos un extremo en V_2 (este link existe porque e_k tiene un extremo en V_2).

Si $i = 1$, entonces e_1 cruza cd , luego supongamos que $i \geq 2$. Como los extremos de e_{i-1}

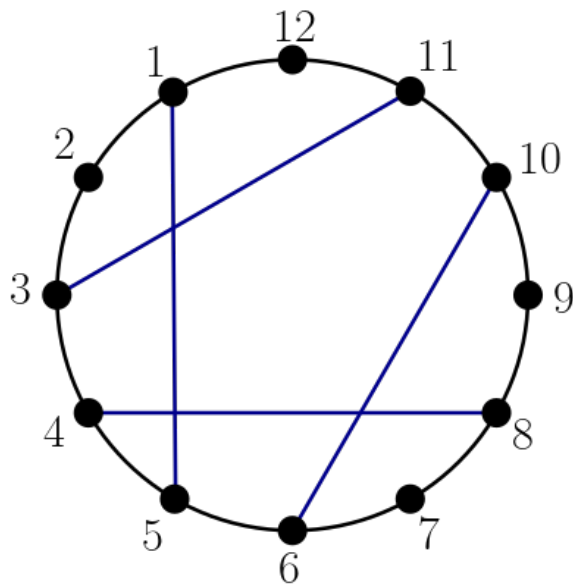


Figura 2.4: Ejemplo de componente circular L en las aristas azules, tales que $V(L) = \{1, 3, 4, 5, 6, 8, 10, 11\}$. Tenemos que $\{4, 5\}$ son vértices internos, que $\{1, 3, 6, 8, 10, 11\}$ son de borde de L , y que $\{2, 7, 9, 12\}$ son vértices externos de L , mientras que las cuerdas de borde son $(1, 3)$, $(1, 11)$, $(6, 8)$ y $(8, 10)$.

están en el intervalo $V_1 \cup \{c, d\}$, uno de los lados de e_{i-1} está totalmente contenido en V_1 . Así, y como e_{i-1} cruza e_i , un extremo de e_i está en V_1 . Concluimos que e_i tiene un extremo en cada lado de cd y por lo tanto lo cruza. \square

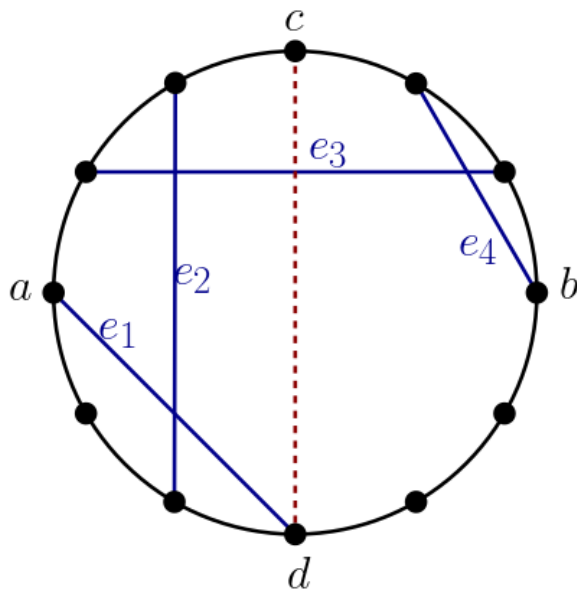


Figura 2.5: Ejemplo del lema 2.12. La cuerda ab es cruzada por cd y el camino de links entre a y b es $P = e_1e_2e_3e_4$, donde el link e_3 atraviesa a la cuerda cd .

Lema 2.13

Sea L una componente circular y ab una cuerda, L cruza ab si y solo si (1) ab es una cuerda interna

de L o (2) ab cruza una cuerda de borde cd de L .

DEMOSTRACIÓN. Dividámoslo en casos, según qué tipo de cuerda sea ab .

Caso 1: ab es una cuerda interna de L . Sean V_1 y V_2 los dos lados definidos por ab , como ab es interno, $V(L)$ tiene vértices en ambos lados. Luego sea $c \in V(L) \cap V_1$, $d \in V(L) \cap V_2$ y W un camino de links entre c y d , como cd cruza a ab , por el lema anterior, existe un link de $W \subseteq L$ que cruza a ab .

Caso 2: ab cruza a una cuerda de borde cd de L . Sea W un camino de links compuesto por cuerdas de L entre c y d , por el lema anterior, existe un link en W que cruza a ab .

Caso 3: ab es una cuerda de borde de L . Como a y b son consecutivos en L , no hay link en L con extremo en uno de los lados de ab y, por lo tanto, L no cruza a ab .

En cualquier otro caso. Al menos un extremo (digamos a) de la cuerda es externo a L . Si partimos de a , c (respectivamente d) es el primer vértice de $V(L)$ que encontramos yendo en dirección de las manecillas del reloj (respectivamente en contra de las manecillas del reloj) a lo largo del ciclo. Luego cd es una cuerda de borde de L . Sea V_1 el lado de cd en donde pertenece a , notemos que $V_1 \cap V(L) = \emptyset$. Como ab no cruza cd por el caso 2, concluimos que b está en el intervalo $V_1 \cup \{c, d\}$. Por ende, un lado de ab está completamente contenido en V_1 y, por lo tanto, este lado no contiene a ningún vértice de $V(L)$. Se sigue que ningún link en L cruza ab . \square

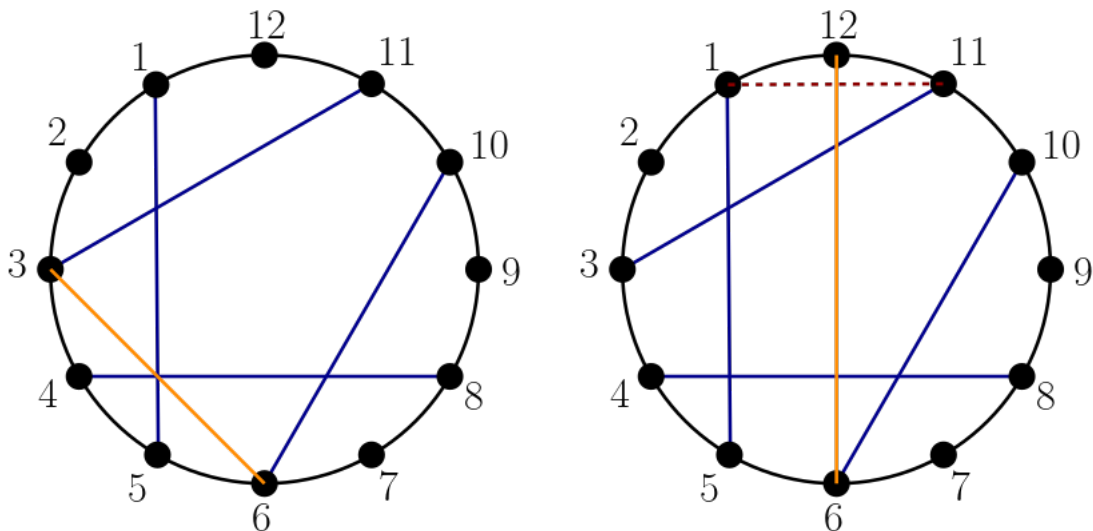


Figura 2.6: Ejemplos de los casos descritos en el lema 2.13, donde una cuerda ab (en amarillo) puede cruzar a una componente circular L (cuyas aristas están en azul). **Izquierda:** ab es una cuerda interna de L . **Derecha:** ab cruza a la cuerda de borde $(1, 11)$.

Con esto, podemos demostrar el Teorema 2.10.

DEMOSTRACIÓN TEOREMA 2.10. Sea $F \subseteq S$ un cubrimiento por aristas, supongamos primero que F es una componente circular; como F es cubrimiento, toda cuerda ab de C_n es interna para F , luego, por el lema 2.13, ab es cruzada por algún link en F . El lema 2.3 implica que F es factible para el problema.

Para la otra dirección, sea F una solución factible para el problema, supongamos por contradicción que el grafo circular G' de F es desconexo y L , una componente conexa de G' . Notamos que L no puede ser un singleton ab , pues por hipótesis tenemos que existe una cuerda de F que cruza a ab , así que L no podría ser maximalmente conexo en G' . Luego L es una componente circular.

Supongamos que L tiene una cuerda de borde ab , por el lema 2.13, L no cruza a ab . Como F cruza a ab , existe un link $cd \in F \setminus L$ tal que cd cruza a ab . El lema 2.13 implica que $L \cup \{cd\}$ es también una componente circular, lo que contradice la maximalidad de L . Concluimos que todas las cuerdas con extremos en $V(L)$ son cuerdas internas de L . La única forma en que esto pueda pasar es que $V(L) = [n]$. Luego cada cuerda del ciclo es interna a L . En particular, cualquier cuerda $cd \in F$ es cruzada por L y, por la maximalidad de L , $cd \in L$. Concluimos que $F = L$ y que por ende F es una componente circular. \square

2.2. Completaciones minimales

Para desarrollar los algoritmos necesitamos un procedimiento que transforme un conjunto de componentes circulares en una solución factible, controlando el costo incurrido.

Definición 2.14

Sea $F \subseteq S$ un conjunto de links, llamamos a un conjunto $Q \subseteq S$ una **completación** de F si $F \cup Q$ es factible para CVCA. Una **completación minimal** de F es una completación minimal por inclusión.

Decimos que dos componentes circulares L_1 y L_2 se cruzan si existe $e_1 \in L_1$ y $e_2 \in L_2$ tal que e_1 cruza a e_2 . Si no existen esos links, diremos que no se cruzan o no son cruzantes.

A grandes rasgos, el algoritmo que presentamos en esta tesis construye una colección \mathcal{L} de componentes circulares disjuntas par a par, tal que se usen pocos links para cubrir muchos vértices. Enseguida, este mismo algoritmo encuentra una completación minimal de esta colección. El siguiente teorema nos permite acotar el tamaño de esta completación.

Teorema 2.15

Sean L_1, \dots, L_k una colección de $k \geq 1$ componentes circulares no cruzantes y $F = L_1 \cup \dots \cup L_k$, toda completación minimal Q de F usa a lo más $n - 3 - \sum_{i=1}^k (|V(L_i)| - 3)$ links.

Este teorema lo probaremos en dos etapas. En un primer momento, lo probaremos para el caso en que solo tenemos una componente circular y, en un segundo momento, lo extenderemos al caso general.

Paso 1. Completando una única componente circular

Lema 2.16

Sea $L \subseteq S$ una componente circular, toda completación minimal Q de L tiene tamaño a lo más $n - |V(L)|$.

DEMOSTRACIÓN. Probemos primero que existe una completación de tamaño $n - |V(L)|$ por inducción en $i = n - |V(L)|$. Si $i = 0$, L es un cubrimiento de aristas y, por el teorema 2.10, es factible, podemos elegir $Q = \emptyset$. Para $i \geq 1$, notamos que debe existir una cuerda de borde ab de L . Como S es factible, debe existir un link cd que cruce la cuerda ab . Luego, por el lema 2.13, $L' = L \cup \{cd\}$ es una componente que cubre al menos un vértice más que L . Por inducción existe una completación Q' de L' con a lo más $n - |V(L')|$ links. Luego $Q = Q' \cup \{cd\}$ es una completación de L , con $|Q| = |Q'| + 1 \leq n - |V(L')| + 1 \leq n - |V(L)|$, lo que concluye la primera demostración.

Si \tilde{Q} es una completación minimal de L , consideramos la nueva instancia de CVCA $(C_n, L \cup \tilde{Q})$. En esta instancia, \tilde{Q} es la única completación de L . Si retomamos la demostración del párrafo anterior, tenemos que $|\tilde{Q}| \leq n - |V(L)|$. \square

Gracias al lema anterior es posible demostrar, sin usar el teorema de Mader, que una solución minimal para el problema es una 2-aproximación.

Proposición 2.17

Toda completación minimal de una sola cuerda tiene a lo más $n - 3$ links, y toda solución minimal para el problema tiene tamaño de a lo más $n - 2$.

DEMOSTRACIÓN. Sea Q una completación minimal de un link $\{e\}$, como $Q \cup \{e\}$ es una solución para el problema, existe un $f \in Q$ que cruza e . Luego $Q \setminus \{f\}$ es una completación minimal de la componente circular $\{e, f\}$. Ya que $Q \setminus \{f\}$ cubre a todos los vértices, excepto, tal vez, los extremos de e y f , tenemos que $|V(Q \setminus \{f\})| \geq n - 4$. Entonces, del lema anterior concluimos que $|Q| = 1 + |Q \setminus \{f\}| \leq 1 + n - |V(\{e, f\})| = 1 + n - 4 = n - 3$.

Ahora bien, si observamos que Q es una solución minimal para el problema, también notamos que $Q - e$ es una solución minimal de $\{e\}$ para cualquier link $e \in Q$. Y por lo expuesto en el párrafo anterior, podemos afirmar que $|Q| = 1 + |Q - e| \leq n - 2$. \square

Paso 2. Completando una colección arbitraria de componentes circulares no cruzantes

Definición 2.18

La **zona abierta** definida por una cuerda de borde ab de una componente circular L es el intervalo de vértices $I(ab, L)$ correspondiente al lado de ab que no tiene ningún vértice de $V(L)$. La **zona cerrada** asociada es $I(ab, L) \cup \{a, b\}$.

El **grafo de zona** definido por una cuerda de borde ab de una componente circular L es el grafo $Z(ab, L, S)$ que se obtiene de $G = ([n], C_n \cup S)$ al hacer las siguientes operaciones: (1) borrar todos los links de L , (2) añadir un link e entre a y b y (3) contraer todos los vértices en $C_n \setminus (I(ab, L) \cup \{a, b\})$ a un único vértice v_0 .

Ejemplo En la figura 2.4, $I((1, 3), L) = \{2\}$, y su zona cerrada asociada es $\{1, 2, 3\}$.

En $Z = Z(ab, L, S)$ removemos todos los *loops* y nos quedamos con solo un link para cada clase paralela. Para cada conjunto $Q \subseteq S$ de cuerdas de G , definimos $\psi(Q)$ como el conjunto de cuerdas asociadas en Z , sin considerar los links que forman parte del ciclo externo (es decir, los links que al contraer nos dan las aristas av_0 y bv_0). Con esta construcción obtenemos una nueva **instancia de zona** $(C_{n'}, \psi(S) \cup \{ab\})$ asociada a ab y L . Notemos que $n' = |I(ab, L) \cup \{a, b\}| + 1 < n$, pues L cubre al menos 4 vértices.

Sea \mathcal{L} una colección de componentes circulares no cruzantes, $L \in \mathcal{L}$ una componente fija y $P(L)$ sus cuerdas de borde, como las componentes no se cruzan, cada componente circular $K \in \mathcal{L} \setminus \{L\}$ está completamente contenida en una zona cerrada de L (i.e $V(K)$ es un subconjunto de la zona cerrada) y cada zona cerrada de L puede contener cero o más componentes de $\mathcal{L} \setminus \{L\}$. Para cada cuerda de borde $ab \in P(L)$, $\mathcal{L}(ab, L)$ es el conjunto de componentes circulares en $\mathcal{L} \setminus \{L\}$ que están en la zona cerrada $I(ab, L) \cup \{a, b\}$ y $F^{ab} = \bigcup_{K \in \mathcal{L}(ab, L)} K$ es el conjunto de links que están contenidos en alguna componente circular de la zona cerrada $I(ab, L) \cup \{a, b\}$. Luego podemos probar el siguiente lema auxiliar.

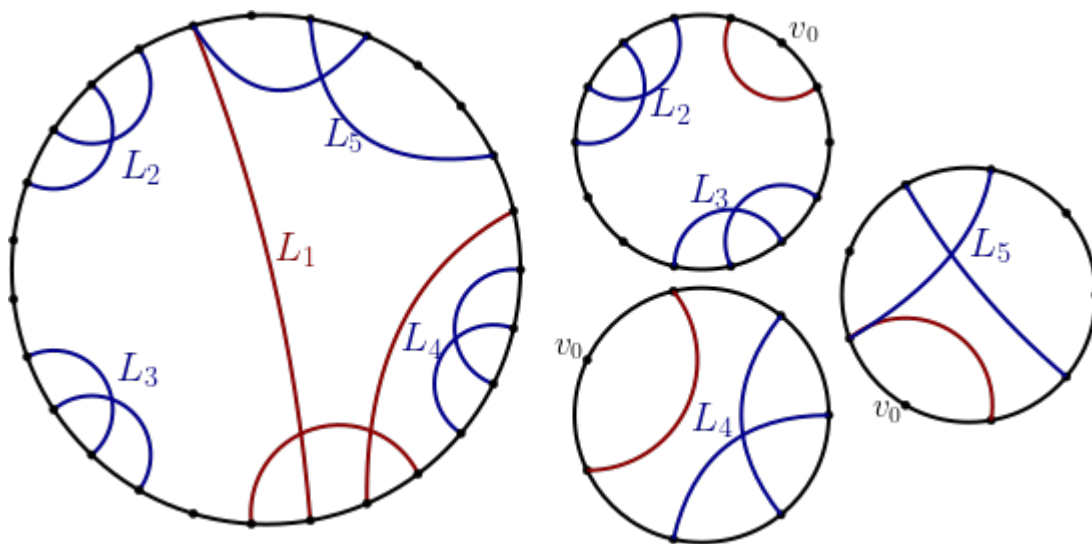


Figura 2.7: **Izquierda:** grafo con 5 componentes. La componente circular L_1 define 3 zonas, las cuales pueden tener solo una componente (las que tienen a L_4 y L_5) que no necesariamente es vértice disjunta a L_1 , dos o más componentes (en la zona que incluye L_2 y L_3) y vértices extras (como los que están alrededor de L_3). **Derecha:** sus grafos de zona correspondientes.

Lema 2.19

Sea $Q \subseteq S$ un conjunto de links, si Q es una completación de $F^{ab} \cup L$ en la instancia original (C_n, S) , entonces $\psi(Q)$ es una completación de $\psi(F^{ab}) \cup \{ab\}$ en la instancia de zona $(C_{n'}, \psi(S) \cup \{ab\})$. Por otro lado, si $\psi(Q)$ es una completación de $\psi(F^{ab}) \cup \{ab\}$ en la instancia de zona, entonces $Q \cup F^{ab}$ cruza todas las cuerdas de C_n con ambos extremos en $I(ab, L) \cup \{a, b\}$.

DEMOSTRACIÓN. Sea Q una completación de L en la instancia original y cd una cuerda en $C_{n'}$, si tanto c como d son vértices no contraídos (es decir, que están en la zona $I(ab, L) \cup \{a, b\}$), entonces no son cruzados por L en la instancia original, luego deben ser cruzados por un link $e \in Q \cup F^{ab}$. Por lo tanto, la arista respectiva $\psi(e) \in \psi(Q) \cup \psi(F^{ab})$ cruza cd en la instancia de zona. Por otro lado, si uno de los vértices de la cuerda cd es v_0 , entonces cd es cruzada por ab . En cualquier caso, cd es cruzado por $\psi(Q) \cup \psi(F^{ab}) \cup \{ab\}$.

Para la dirección opuesta, suponemos que $\psi(Q)$ es una completación de $\psi(F^{ab}) \cup \{a, b\}$ en la instancia de zona. Notamos que todos los vértices de $I(ab, L) \cup \{a, b\}$ son vértices no contraídos. En consecuencia, para cada cuerda cd en C_n entre dos vértices de ese conjunto, existe un link f' en $\psi(Q) \cup \psi(F^{ab})$ que cruza la cuerda en $C_{n'}$. Cualquier link $f \in Q \cup F^{ab}$ con $\psi(f) = f'$ cruza la cuerda cd en la instancia original. \square

DEMOSTRACIÓN TEOREMA 2.15. Procedemos por inducción en n . Sea Q una completación minimal de $F = L_1 \cup \dots \cup L_k$, el caso en que $n = 4$ es trivial pues L_1 cubre todos los vértices. Por lo tanto, \emptyset es la completación minimal buscada.

Consideremos una cuerda $ab \in P(L_1)$. El lema 2.19 nos garantiza que $\psi(Q)$ es una completación de $\psi(F^{ab}) \cup \{ab\}$ en la instancia de zona correspondiente.

Entonces, $\psi(Q) \cup \{ab\}$ es una completación de $\psi(F^{ab})$ en la misma instancia de zona. Por ende, existe un subconjunto $Q'_{ab} \subseteq Q$ con $|Q'_{ab}| = |\psi(Q'_{ab})|$ tal que $\psi(Q'_{ab})$ o $\psi(Q'_{ab}) \cup \{ab\}$ son completaciones minimales de $\psi(F^{ab})$ en esa instancia de zona (y en cualquier caso, Q'_{ab} tiene tamaño a lo más del que tendría una completación minimal de $\psi(F^{ab})$ en el grafo de zona).

Tenemos dos casos: cuando $\mathcal{L}(ab, L_1)$ es vacío y cuando no lo es. Consideremos primero el caso en que $\mathcal{L}(ab, L_1)$ es no vacío. Recordemos que la instancia tiene $n_{ab} < n$ vértices. Ahora usemos la hipótesis de inducción en la instancia de zona para concluir que

$$|Q'_{ab}| \leq n_{ab} - 3 - \sum_{J \in \mathcal{L}(ab, L_1)} (|V(J)| - 3) = |I(ab, L_1)| - \sum_{J \in \mathcal{L}(ab, L_1)} (|V(J)| - 3).$$

En el segundo caso, cuando $\mathcal{L}(ab, L_1)$ y F^{ab} son vacíos, $\psi(Q)$ es una completación de un solo link $\{ab\}$ en la instancia de zona, por lo que existe un conjunto $Q'_{ab} \subseteq Q$ con $|Q'_{ab}| = |\psi(Q'_{ab})|$ tal que $\psi(Q'_{ab})$ es una completación minimal de $\{ab\}$. Por la proposición 2.17, $|Q'_{ab}| = |\psi(Q'_{ab})| \leq n_{ab} - 3 = |I(ab, L_1)| = |I(ab, L_1)| - \sum_{J \in \mathcal{L}(ab, L_1)} (|V(J)| - 3)$, pues la suma es vacía.

Definamos $Q' = \bigcup_{ab \in P(L_1)} Q'_{ab}$, demostremos que Q' es una completación de $F = L_1 \cup \dots \cup L_k$ en la instancia original. Sea cd una cuerda cualquiera de $C_{n'}$, probemos que $Q' \cup F$

la cruza. Si cd es cruzada por L_1 , entonces queda demostrado inmediatamente, así que supongamos que no es cruzada por L_1 . Por el lema 2.13 cd no es una cuerda interna de L_1 ni cruza una cuerda de borde de L_1 , luego solo nos queda que cd conecta dos vértices en $I(ab, L_1) \cup \{a, b\}$ para alguna cuerda de borde $ab \in P(L_1)$. El lema 2.19 implica que cd es cruzada por un link en $Q'_{ab} \cup F^{ab} \subseteq Q' \cup F$, por lo que la afirmación queda demostrada.

Por último, como $Q' \subseteq Q$ y Q es una completación minimal de F necesariamente $Q' = Q$. De acuerdo con las cotas de arriba y el hecho de que las zonas abiertas $\{I(ab, L_1) : ab \in P(L_1)\}$ particionan $[n] \setminus V(L_1)$, obtenemos que

$$\begin{aligned}
|Q| &\leq \sum_{ab \in P(L_1)} |Q'_{ab}| \leq \sum_{ab \in P(L_1)} \left(|I(ab, L_1)| - \sum_{J \in \mathcal{L}(ab, L_1)} (|V(J)| - 3) \right) \\
&= (n - |V(L_1)|) - \sum_{J \in \mathcal{L} \setminus \{L_1\}} (|V(J)| - 3) = n - 3 - \sum_{J \in \mathcal{L}} (|V(J)| - 3). \quad \square
\end{aligned}$$

Capítulo 3

Algoritmo de búsqueda local

En este capítulo se presenta el enfoque algorítmico del trabajo. Este se presenta en dos partes: en la primera, desarrollamos el algoritmo en sí y, en la segunda, exploramos algunas propiedades de los conjuntos formados por nuestro algoritmo. Antes de iniciar, necesitamos definir un conjunto de links especiales que serán fundamentales para nuestro algoritmo.

Definición 3.1

Sea $F \subseteq S$ un conjunto de links, decimos que F es **singleton-libre** si el grafo circular G' de F no tiene singletons.

Las componentes conexas de G' forman una familia \mathcal{L} de componentes circulares maximales no cruzantes. Definimos la **utilidad** de F como

$$U(F) = -|F| + \sum_{J \in \mathcal{L}} (|V(J)| - 3).$$

En un inicio, nos enfocamos en encontrar conjuntos singleton-libres de forma creciente (i.e. solo agregando links), cuya utilidad con respecto a un parámetro creciente (por ejemplo, el número de vértices cubiertos) sea grande. El siguiente lema muestra que conjuntos de alta utilidad proporcionan soluciones con mejores garantías de aproximación.

Lema 3.2

Si Q es una completación minimal de un conjunto singleton-libre F , entonces $Q \cup F$ es una solución para el problema con a lo más $n - 3 - U(F)$ links.

DEMOSTRACIÓN. Por el teorema 2.15, Q tiene tamaño a lo más $n - 3 - \sum_{J \in \mathcal{L}} (|V(J)| - 3) = n - 3 - |F| - |U(F)|$, luego $|Q| + |F| \leq n - 3 - U(F)$. \square

Para cualquier $\alpha \in (1/2, 1)$ y cualquier tamaño fijo $N_{\max} \in \mathbb{N}$, definimos un algoritmo

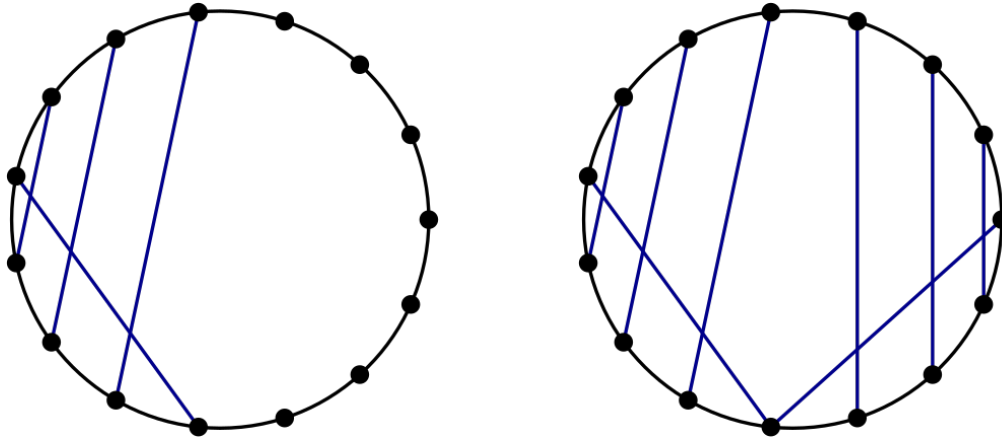


Figura 3.1: Ejemplos de utilidad para los conjuntos de aristas azules F . **Izquierda:** F solo tiene una componente y $U(F) = 1$ **Derecha:** F tiene dos componentes que tienen un vértice en común y $U(F) = 2$

de búsqueda local que consiste en dos fases. En la primera construimos un conjunto de links singleton-libre F , añadiendo a lo más N_{\max} links en cada paso. En cada iteración, la ganancia marginal de utilidad es a lo menos $(1 - \alpha)$ veces el número de vértices nuevos cubiertos, con lo que, al final la utilidad de F es al menos $(1 - \alpha)|V(F)|$. En la segunda fase el algoritmo encuentra una completación minimal Q de F y retorna el par (Q, F) . El conjunto $Q \cup F$ es la solución propuesta.

Algoritmo 2 Algoritmo de búsqueda local

entrada: instancia (C_n, S) del problema. $\alpha \in (1/2, 1)$, $N_{\max} \in \mathbb{N}$.

Fase 1– Construir una colección de componentes circulares no cruzantes

- 1: $F \leftarrow \emptyset$.
 - 2: **mientras** podamos encontrar un conjunto $K \subseteq S \setminus F$ tal que $|K| \leq N_{\max}$, $F \cup K$ sea singleton-libre y $U(F \cup K) - U(F) \geq (1 - \alpha)|V(F \cup K) \setminus V(F)|$ **hacer**
 - 3: $F \leftarrow F \cup K$
 - 4: **fin mientras**
-

Fase 2– Encontrar una completación

- 5: encontrar una completación minimal Q de F .
 - 6: **devolver** (Q, F)
-

Para ganar intuición sobre el algoritmo, supongamos que, en una iteración de la fase 1, el algoritmo encuentra un conjunto K de a lo más N_{\max} links para añadir a la solución actual F . Sea \mathcal{L} la familia de componentes circulares actuales de F , el caso más simple es cuando K cruza a exactamente una componente circular $L \in \mathcal{L}$ y $V(K)$ no interseca a ninguna otra componente de \mathcal{L} . En ese caso, la ganancia marginal de utilidad $U(F \cup K) - U(F)$ es simplemente $-|K| + |V(L \cup K) \setminus V(L)| = -|K| + |V(F \cup K) \setminus V(F)|$. Luego, el algoritmo agrega K a F mientras se cumpla que $-|K| + |V(L \cup K) \setminus V(L)| \geq (1 - \alpha)|V(L \cup K) \setminus V(L)|$ o, equivalentemente, si la razón de nuevos links $|K|$ con respecto al número de vértices nuevos cubiertos $|V(F \cup K) \setminus V(F)|$ es a lo más α . Es decir, el algoritmo solamente añade

conjuntos K de links que cubran nuevos vértices con pocas aristas.

La intuición detrás de añadir conjuntos K que cruzan o tocan más de una componente es menos palpable, pero, en la práctica, conectar más componentes produce ganancias marginales de mayor utilidad.

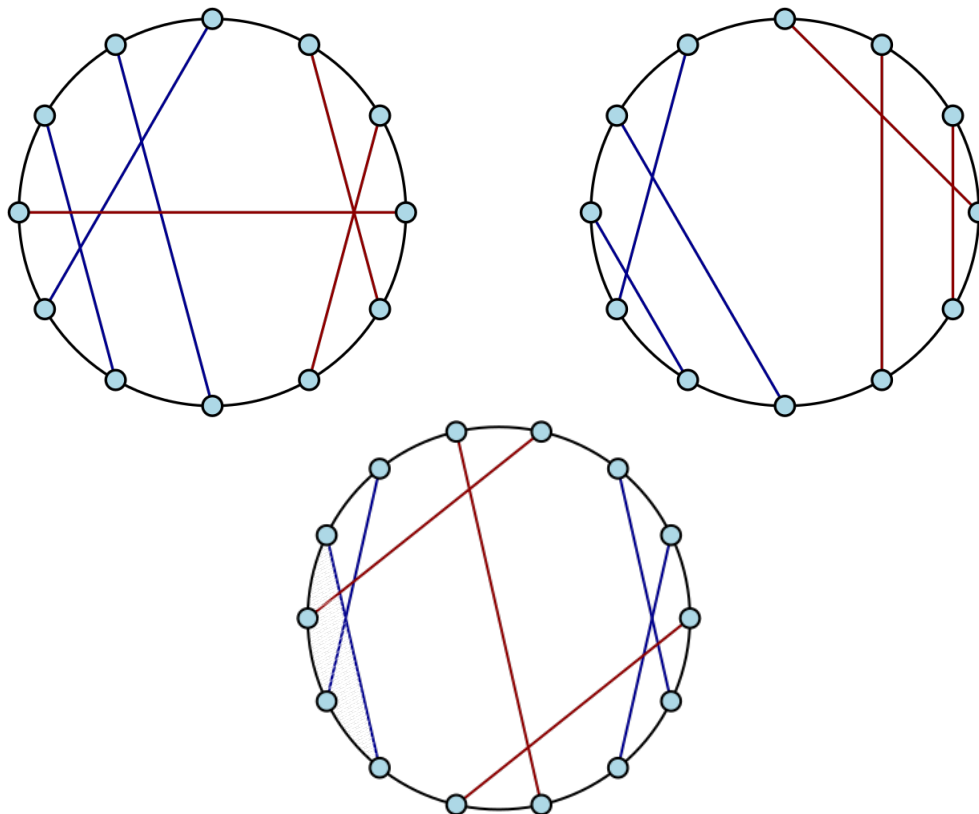


Figura 3.2: El algoritmo puede tomar varios pasos locales. En la figura los links azules representan al conjunto F y los rojos, al conjunto K . **Izquierda superior:** K expande una componente, **Derecha superior:** K es una nueva componente circular, **Centro inferior:** K une dos componentes circulares.

Lema 3.3

Para cualquier N_{\max} fijo y $\alpha \in (1/2, 1)$, el algoritmo de búsqueda local corre en tiempo polinomial y retorna un par (Q, F) , tal que $Q \cup F$ es factible para el problema y $|Q \cup F| < n - (1 - \alpha)|V(F)| = \alpha|V(F)| + (n - |V(F)|)$.

DEMOSTRACIÓN. Sea $m = |S|$, cada iteración de la primera fase se puede hacer probando todos los $m^{O(N_{\max})}$ subconjuntos posibles de $S \setminus L$ de tamaño a lo más N_{\max} , por ende, la primera fase toma tiempo polinomial. Encontrar una completación minimal Q se puede hacer también en tiempo polinomial, si se remueven iterativamente links innecesarios de $S \setminus F$.

Sea q el número de iteraciones de la primera fase y F_i el conjunto F obtenido al final de la i -ésima iteración (donde $F_0 = \emptyset$), por una suma telescópica y porque $V(F_{i-1}) \subseteq V(F_i)$

para cada i , tenemos

$$U(F) = \sum_{i=1}^q U(F_i) - U(F_{i-1}) \geq (1 - \alpha) \sum_{i=1}^q |V(F_i) \setminus V(F_{i-1})| = (1 - \alpha)|V(F)|.$$

Por el lema 3.2, $Q \cup F$ es factible y tiene tamaño a lo más $n - 3 - (1 - \alpha)|V(F)| < n - (1 - \alpha)|V(F)| = \alpha|V(F)| + (n - V(F))$. \square

3.1. Conjuntos críticos

Definición 3.4

Un conjunto singleton-libre $F \subseteq S$ se denomina (α, N_{\max}) -crítico si no hay conjunto $K \subseteq S \setminus F$ de tamaño a lo más N_{\max} , tal que $F \cup K$ sea singleton-libre y $U(F \cup K) - U(F) \geq (1 - \alpha)|V(F \cup K) \setminus V(F)|$.

A lo largo de esta sección, F^* es un conjunto (α, N_{\max}) -crítico, con $N_{\max} \geq 1$ (por ejemplo, el conjunto F al final de la primera fase del algoritmo). También definimos $V' = V(C_n) \setminus V(F^*)$ y \mathcal{L} como el conjunto de componentes circulares no cruzantes asociadas a F^* . Todos los vértices de borde de las componentes circulares de \mathcal{L} , los denotamos como $B(F^*) \subseteq V(F^*)$ y definimos $A(F^*) = V(F^*) \setminus B(F^*)$. Notemos que $A(F^*)$ corresponde exactamente a los vértices internos de las componentes en \mathcal{L} , pues los vértices internos de una componente circular no pueden ser vértices de borde de otra componente circular, ya que en este caso sería vértice de borde para ambas componentes. Recordemos que $P(J)$ es el conjunto de cuerdas de borde de una componente J . Definimos el **perímetro** $P(F^*) = \bigcup_{J \in \mathcal{L}} P(J)$. La siguiente observación proviene de que el perímetro $P(F^*)$ es un conjunto de cuerdas que cubre todos los vértices de $B(F^*)$.

Lema 3.5

$$|P(F^*)| \geq |B(F^*)|/2.$$

DEMOSTRACIÓN. El lema es trivial porque una cuerda de borde incide a lo más en 2 vértices de $|B(F^*)|$, por lo tanto, debe haber al menos $|B(F^*)|/2$ cuerdas distintas. \square

El siguiente lema nos será útil pues implica que los links en $S \setminus F^*$ cruzan a lo más una componente $L \in \mathcal{L}$ y, si lo hacen, cruzan a lo más una cuerda de borde de L y, en ese caso, al menos un extremo debe estar en $V(L) \subseteq V(F^*)$.

Lema 3.6

Cualquier link $e \in S$ puede cruzar a lo más una cuerda en $P(F^*)$. Si ambos extremos de e están en V' , entonces e no puede cruzar ninguna cuerda en $P(F^*)$.

Antes, demostraremos un par de lemas técnicos que facilitarán la demostración.

Lema 3.7

Sea $e = ab$ una cuerda en C_n y \mathcal{L}_e una colección de componentes circulares no cruzantes tal que todas ellas cruzan a e ,

$$|V(\bigcup_{J \in \mathcal{L}_e} J)| \geq |\mathcal{L}_e| + 2 + \sum_{J \in \mathcal{L}_e} (|V(J)| - 3).$$

DEMOSTRACIÓN. El resultado es directo si $|\mathcal{L}_e| = 1$. Así que supondremos que la colección \mathcal{L}_e contiene $k \geq 2$ componentes circulares. Como todos los vértices de una componente $L \in \mathcal{L}_e$ (y por consiguiente, todas las cuerdas internas o de borde de L) están completamente contenidas en una zona de toda otra componente $K \in \mathcal{L}_e \setminus \{L\}$, sabemos que e no puede ser una cuerda interna o de borde de cualquier $L \in \mathcal{L}_e$. Por el lema 2.13, e debe cruzar una arista de borde de cada componente. De hecho, si ambos extremos de e están fuera de $V(L)$, e cruza exactamente dos aristas de borde f_L y g_L de L . Y si un extremo está en $V(L)$ y el otro afuera, e solo cruza una cuerda de borde f_L de L . Abusando de la notación, diremos que $g_L = f_L$ en este caso.

Notemos que si dibujamos C_n en el plano como un círculo y dibujamos los links como segmentos de línea recta, entonces e cruza todas las cuerdas de borde en el multiconjunto $\{f_L, g_L\}_{L \in \mathcal{L}_e}$ en un cierto orden. Justamente, se pueden enumerar las componentes circulares de \mathcal{L}_e como L_1, \dots, L_k de forma que e cruza las cuerdas descritas arriba en el orden $f_{L_1}, g_{L_1}, f_{L_2}, g_{L_2}, \dots, f_{L_k}, g_{L_k}$, donde para algún i puede pasar que g_{L_i} sea igual a $f_{L_{i+1}}$. De forma más precisa, sin pérdida de generalidad, podemos suponer que uno de los extremos de e es el vértice n y el otro extremo es el vértice $j \in [n - 1]$. Luego, todas las cuerdas de borde en la lista de arriba pueden verse como intervalos en $[1, n - 1]$ que contienen a j y que no se cruzan, por lo que pueden ser ordenados por inclusión, con lo cual se obtiene el orden antes mencionado. Que ambas aristas de borde de una componente aparecen antes que las dos aristas de borde de la siguiente, se puede probar por que toda componente circular está contenida en una zona cerrada de cualquier otra componente en la lista.

Para todo $i \in [k - 1]$, todos los vértices de $V(L_1 \cup \dots \cup L_i)$ están en la zona cerrada definida por la cuerda de borde $f_{L_{i+1}}$ de L_{i+1} . De aquí concluimos que $V(L_1 \cup \dots \cup L_i)$ y $V(L_{i+1})$ pueden intersectar a lo más en dos vértices. Por lo tanto, $|V(L_1 \cup \dots \cup L_{i+1})| \geq |V(L_1 \cup \dots \cup L_i)| + |V(L_{i+1})| - 2$. Si iteramos este argumento, obtenemos $|V(L_1 \cup \dots \cup L_k)| \geq 2 + \sum_{i=1}^k (|V(L_i)| - 2) = k + 2 + \sum_{i=1}^k (|V(L_i)| - 3)$, tal como se buscaba. \square

Lema 3.8

Sea e una cuerda de C_n , e cruza a lo más una componente circular de \mathcal{L} .

DEMOSTRACIÓN. Sea \mathcal{L}_e la familia de componentes circulares en \mathcal{L} que cruzan a e , $X = \bigcup_{J \in \mathcal{L}_e} J$ tal que $X \cup \{e\}$ es la componente circular única creada al añadir e a F^* ; notemos que si e cruza al menos una componente circular entonces $F^* \cup \{e\}$ es singleton-libre.

Supongamos que $|\mathcal{L}_e| \geq 3$. Por el lema 3.7, tenemos que $|V(X \cup \{e\})| \geq |V(X)| \geq 5 + \sum_{J \in \mathcal{L}_e} (|V(J)| - 3)$. Por ende,

$$\begin{aligned} U(F^* \cup \{e\}) &= -|F^* \cup \{e\}| + (|V(X \cup \{e\})| - 3) + \sum_{J \in \mathcal{L} \setminus \mathcal{L}_e} (|V(J)| - 3) \\ &\geq -(|F^*| + 1) + 2 + \sum_{J \in \mathcal{L}} (|V(J)| - 3) \geq U(F^*) + 1. \end{aligned}$$

Como e conecta a lo más 2 vértices nuevos, $U(F^* \cup \{e\}) - U(F^*) \geq 1 \geq |V(F^* \cup \{e\}) \setminus V(F^*)|/2 \geq (1 - \alpha)|V(F^* \cup \{e\}) \setminus V(F^*)|$, lo que no es posible dado que F^* es (α, N_{\max}) -crítico y $N_{\max} \geq 1$.

Supongamos ahora que $|\mathcal{L}_e| = 2$ y digamos que $\mathcal{L}_e = \{L_1, L_2\}$. Como L_1 y L_2 no se cruzan, $|V(L_1) \cap V(L_2)| \leq 2$ y, por ende, $|V(L_1)| + |V(L_2)| - 2 \leq |V(L_1 \cup L_2)|$. Luego,

$$\begin{aligned} U(F^* \cup \{e\}) - U(F^*) &= -1 + (|V(X \cup \{e\})| - 3) - (|V(L_1)| - 3) - (|V(L_2)| - 3) \\ &= 2 + |V(\{e\} \cup L_1 \cup L_2)| - |V(L_1)| - |V(L_2)| \\ &\geq |V(\{e\} \cup L_1 \cup L_2)| - |V(L_1 \cup L_2)|. \end{aligned}$$

Si e tiene al menos un extremo en $V' = V(C_n) \setminus V(F^*)$, entonces el lado derecho de la ecuación de arriba es al menos 1 y, por lo tanto, $U(F^* \cup \{e\}) - U(F^*) \geq 1 \geq |V(F^* \cup \{e\}) \setminus V(F^*)|/2 \geq (1 - \alpha)|V(F^* \cup \{e\}) \setminus V(F^*)|$, lo que contradice la definición de conjunto (α, N_{\max}) -crítico.

Por último, si $|V(F^* \cup \{e\}) \setminus V(F^*)| = 0$, tenemos que $U(F^* \cup \{e\}) - U(F^*) \geq |V(\{e\} \cup L_1 \cup L_2)| - |V(L_1 \cup L_2)| \geq 0 = (1 - \alpha)|V(F^* \cup \{e\}) \setminus V(F^*)|$, lo que tampoco es posible. \square

DEMOSTRACIÓN LEMA 3.6. Supongamos que e cruza al menos una cuerda de borde en $P(F^*)$, por el lema 2.13, también cruza al menos una componente circular y, por el lema 3.8, debe cruzar exactamente una. Llamemos $L \in \mathcal{L}$ a esa componente cruzada por e .

Notemos que e puede cruzar a lo más 2 cuerdas de borde de L y que la única forma en que eso pasa es cuando los extremos de e están fuera de $V(L)$ y en zonas distintas de L . En este caso, $U(F^* \cup \{e\}) - U(F^*) = -1 + |V(L \cup \{e\})| - |V(L)| = 1 \geq |V(F^* \cup \{e\}) - V(F^*)|(1 - \alpha)$, lo cual no es posible dado que F^* es (α, N_{\max}) -crítico.

Luego, cualquier link $e \in S$ puede cruzar a lo más una cuerda en $P(F^*)$. Más aún, si ambos extremos están en V' y e cruza una cuerda en $P(F^*)$, dicho link, por lo visto en el párrafo anterior, tiene que cruzar 2 cuerdas de $P(F^*)$, lo que no puede pasar. En ese caso, e no puede cruzar ninguna cuerda de P . \square

Queremos usar estas propiedades para obtener cotas inferiores para el tamaño de OPT cuando F^* es pequeño. Dada esa circunstancia, cualquier matching maximal $M \subseteq \text{OPT}[V']$ fuera de $V(F^*)$ debe ser grande. La siguiente definición y lema nos ayudarán en esta tarea.

Definición 3.9

Un link $e \in S$ **conecta** un conjunto de links X si el grafo circular asociado a $X \cup \{e\}$ es conexo.

Lema 3.10

Supongamos que $N_{\max} \geq \lceil (5 - 2\alpha)/(2\alpha - 1) \rceil + 1$, sea $M \subseteq S$ cualquier matching consistente en links con ambos extremos en V' , cada link $e \in S \setminus (M \cup F^*)$ conecta a lo más $\max \left\{ 0, \left\lceil \frac{5 - 2X(e) - V_{F^*}(e) - (4 - V_M(e) - V_{F^*}(e))\alpha}{2\alpha - 1} \right\rceil \right\}$ links de M , donde $X(e)$ es la indicatriz de que e cruza F^* , $V_M(e) = |V(\{e\}) \cap V(M)|$ y $V_{F^*}(e) = |V(\{e\}) \cap V(F^*)|$. En particular, ningún link de $S \setminus (M \cup F^*)$ conecta más que $\ell = \lceil (5 - 2\alpha)/(2\alpha - 1) \rceil$ links de M .

DEMOSTRACIÓN. Sea $\gamma = \max \left\{ 0, \left\lceil \frac{5 - 2X(e) - V_{F^*}(e) - (4 - V_M(e) - V_{F^*}(e))\alpha}{2\alpha - 1} \right\rceil \right\}$ la cota del lema, siempre se tiene que $\gamma \leq \ell \leq N_{\max} - 1$.

Sea M_e el conjunto de links de M que e conecta (este conjunto contiene el conjunto de links de M que e cruza), supongamos por contradicción que $|M_e| > \gamma$. Si $|M_e| > N_{\max} - 1$, redefinimos M_e como cualquier subconjunto de M_e con $N_{\max} - 1$ links que e conecta. Para encontrar este subconjunto eliminamos hojas que no sean e de algún árbol cobertor T del grafo circular de $M_e \cup \{e\}$, dado que este es conexo. También decimos que $V_{M_e}(e) = V(e \cap V(M_e))$.

En cualquier caso, tenemos que $2 \leq \gamma + 2 \leq |M_e \cup \{e\}| \leq N_{\max}$ y que $F^* \cup M_e \cup \{e\}$ es singleton-libre. Más aún, $|V(M_e \cup \{e\} \cup F^*) \setminus V(F^*)| = 2|M_e| + |V(\{e\}) \setminus V(M_e \cup F^*)| = 2|M_e| + 2 - V_{M_e}(e) - V_{F^*}(e)$. Luego, como F^* es (α, N_{\max}) -crítico, se sigue que

$$U(M_e \cup \{e\} \cup F^*) - U(F^*) < (1 - \alpha)(2|M_e| + 2 - V_{M_e}(e) - V_{F^*}(e)).$$

Si e no cruza F^* , entonces $M_e \cup \{e\}$ es una componente circular que no cruza F^* . Por ende, $U(M_e \cup \{e\} \cup F^*) - U(F^*) = -|M_e \cup \{e\}| + |V(M_e \cup \{e\})| - 3 = -|M_e| - 1 + (2|M_e| + 2 - V_{M_e}(e)) - 3 = |M_e| - V_{M_e}(e) - 2$.

Si e cruza F^* , solo cruza una componente circular L y sabemos que exactamente un extremo está en $V(L)$. Sea v el otro extremo de e , notemos que $V_{M_e}(e) = 1$ si $v \in V(M_e)$ y que $V_{M_e}(e) = 0$ si $v \notin V(M_e)$. De este modo,

$$\begin{aligned} U(M_e \cup \{e\} \cup F^*) - U(F^*) &= -|M_e \cup \{e\}| + |V(M_e \cup \{e\} \cup L)| - |V(L)| \\ &= -|M_e| - 1 + |\{v\} \cup V(M_e)| \\ &= -|M_e| - 1 + 2|M_e| + (1 - V_{M_e}(e)) \\ &= |M_e| - V_{M_e}(e). \end{aligned}$$

En resumen, siempre se cumple que

$$|M_e| - V_{M_e}(e) - 2 + 2X(e) < (1 - \alpha)(2|M_e| + 2 - V_{M_e}(e) - V_{F^*}(e))$$

Una vez resuelta la desigualdad, obtenemos que $|M_e| < \frac{4 - 2X(e) - V_{F^*}(e) - \alpha(2 - V_{M_e}(e) - V_{F^*}(e))}{2\alpha - 1}$. Como $V_{M_e}(e) \leq V_M(e)$, la desigualdad se mantiene si reemplazamos $V_{M_e}(e)$ por $V_M(e)$.

Además, como $|M_e|$ es entero, esto significa que

$$\begin{aligned} |M_e| &\leq \left\lceil \frac{4 - 2X(e) - V_{F^*}(e) - \alpha(2 - V_M(e) - V_{F^*}(e))}{2\alpha - 1} - 1 \right\rceil \\ &= \left\lceil \frac{5 - 2X(e) - V_{F^*}(e) - \alpha(4 - V_M(e) - V_{F^*}(e))}{2\alpha - 1} \right\rceil = \gamma < |M_e|, \end{aligned}$$

lo que es una contradicción. □

Capítulo 4

Análisis del algoritmo de búsqueda local

Ahora presentaremos dos análisis del algoritmo que construimos en el capítulo anterior, para mostrar que este nos otorga mejores garantías de aproximación. El primer análisis ocupa cotas inferiores obtenidas de los lemas vistos y, pese a ser rudimentario, muestra que es posible superar el orden de aproximación de 2. En el segundo análisis, desarrollamos un programa lineal cuyas restricciones ocupan todas las propiedades y constantes que hemos hallado anteriormente, y para encontrar soluciones óptimas para el problema primal y el dual, utilizamos *solvers* computacionales.

Antes de iniciar propiamente con los análisis, es importante definir algunos términos y notaciones que usaremos a lo largo de todo este capítulo. Llamamos F a un conjunto (α, N_{\max}) -crítico, OPT a una solución óptima para **CVCA**, M a un matching maximal de $\text{OPT}[V']$ (es decir, con ambos extremos fuera de $V(F)$) y $\text{ALG}(F)$ al conjunto obtenido cuando se añade una completación minimal a F . Para facilitar la notación, digamos que $m = |M|/n$, $f = |V(F)|/n$, $\text{opt} = |\text{OPT}|/n$ y $\text{alg}(F) = |\text{ALG}(F)|/n$. En particular, si (F, Q) es el par obtenido por nuestro algoritmo de búsqueda local, nuestra garantía de aproximación es $|\text{ALG}(F)|/|\text{OPT}| = \text{alg}(F)/\text{opt}$.

El primer análisis está completamente contenido en la demostración del siguiente lema, que nos proporciona un orden de aproximación mejor que 2.

Lema 4.1

Si $N_{\max} \geq \ell + 1$, donde $\ell = \lceil \frac{5-2\alpha}{2\alpha-1} \rceil$, entonces $\text{opt} \geq \max\{1/2, 1 - f - m, m + \frac{m}{\ell}\}$. De modo que si $\alpha = 3/4$, entonces $\text{alg}(F)/\text{opt} \leq 63/32 = 1,96875$.

DEMOSTRACIÓN. Por el lema 2.4, $\text{opt} > 1/2$. Para deducir $\text{opt} \geq 1 - f - m$, notemos que como M es un matching maximal no hay ningún link $e \in \text{OPT}$ entre dos vértices de $V(C_n) \setminus (V(M) \cup V(F))$. Por lo tanto, todo vértice en $V(C_n) \setminus (V(M) \cup V(F))$ es cubierto por un link diferente en $\text{OPT} \setminus M$. Concluimos que $\text{opt} \geq m + |V(C_n) \setminus (V(M) \cup V(F))|/n = m + (1 - 2m - f) = 1 - f - m$.

Como OPT es factible, $\text{OPT} \setminus M$ debe conectar a todos los links de M . Por el lema 3.10,

cada link en $S \setminus (M \cup F^*)$ conecta a lo más ℓ links de M . Luego, $\text{OPT} \setminus M$ contiene al menos $|M|/\ell$ links, por ende, $\text{opt} \geq m + m/\ell$ o, equivalentemente, $m \leq \text{opt}\ell/(\ell + 1)$. De esta manera se prueba la última cota inferior para opt .

Por el lema 3.3, $\text{alg}(F) < 1 - (1 - \alpha)f = \alpha + (1 - \alpha)(1 - f) = \alpha + (1 - f - m)(1 - \alpha) + m(1 - \alpha)$. Usando $1 \leq 2\text{opt}$, $(1 - f - m) \leq \text{opt}$ y $m \leq \text{opt}\ell/(\ell + 1)$, obtenemos $\text{alg}(F)/\text{opt} < 2\alpha + (1 - \alpha) + (\ell/(\ell + 1))(1 - \alpha) = (2\ell + 1 + \alpha)/(\ell + 1)$. Entre los valores de $\alpha \in (1/2, 1)$ que hacen a $(5 - 2\alpha)/(2\alpha - 1)$ entero, la cota más pequeña ocurre cuando $\alpha = 3/4$ (o $\ell = 7$). Así obtenemos la garantía de aproximación, tal y como se ve en la figura 4.1. \square

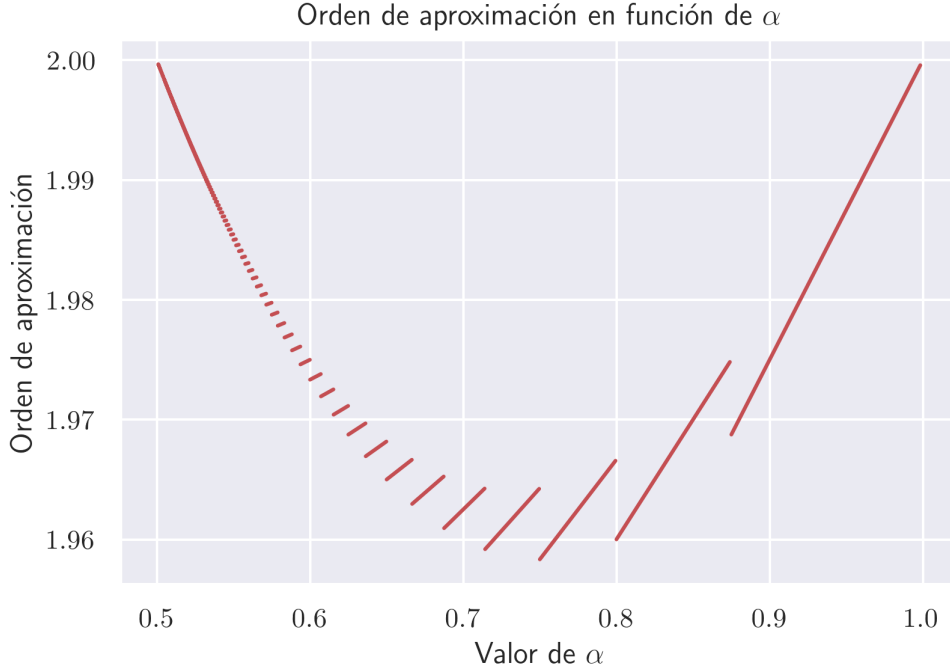


Figura 4.1: Gráfico de la función $h(\alpha) = \frac{2\ell+1+\alpha}{\ell+1}$ que acota superiormente el orden de aproximación.

4.1. Refinamiento del análisis

Para mejorar el orden obtenido construiremos una formulación como programa lineal que nos otorgue una cota inferior para $|\text{OPT}|$. Para esto, particionamos OPT en varios subconjuntos (en donde uno será M), les asignamos variables que representen sus respectivos tamaños y buscamos algunas relaciones y restricciones que estos tamaños deben cumplir.

Recordemos que $A = A(F)$ y $B = B(F)$ son respectivamente los vértices internos y de borde de $V(F)$. Sea $C = V(M)$ y $D = V(C_n) \setminus (A \cup B \cup C)$, tal que A, B, C y D particionan $[n]$, y sea $\text{OPT}_{ij} \subseteq \text{OPT} \setminus M$, donde $i, j \in \{A, B, C, D\}$, el conjunto de links de $\text{OPT} \setminus M$ con un extremo en i y el otro en j ¹; notemos que algunos de esos conjuntos son vacíos, por ejemplo OPT_{dd} , ya que M es matching maximal. Más aún, denotamos OPT_{ij}^k ,

¹Estos subíndices están escritos con letra minúscula y en orden alfabético.

donde $k \in \{M, P, MP, R\}$, al conjunto de links en OPT_{ij}^k que cruzan M (pero no cruzan a ninguna cuerda del perímetro P), P (pero no cruzan a M), ambos (MP) o ninguno (R). Así, obtenemos la partición $\text{OPT} = M \cup \bigcup_{ij,k} \text{OPT}_{ij}^k$.

La variable x_{ij}^k representa a $|\text{OPT}_{ij}^k|$. Además, definamos las variables x_A , x_B y x_M , que denotan respectivamente el número de vértices en A y B y el número de links de M . Igualmente, definamos ℓ_{ij}^k para $k \in \{M, MP\}$ como la cota superior (dada por los lemas 3.6 y 3.10) del número de links de M que un link $e \in \text{OPT}_{ij}^k$ puede cruzar. Estos valores solo dependen de α y no de F . Por lo anterior, podemos deducir el programa lineal $PL(n, |V(F)|, \alpha)$, descrito en la figura 4.2.

$$\begin{array}{ll}
\text{(PL)} & \text{mín } x_M + \sum_{ij,k} x_{ij}^k \\
\sum_k (2x_{aa}^k + x_{ab}^k + x_{ac}^k + x_{ad}^k) & \geq x_A \\
\sum_k (x_{ab}^k + 2x_{bb}^k + x_{bc}^k + x_{bd}^k) & \geq x_B \\
\sum_k (x_{ad}^k + x_{bd}^k + x_{cd}^k + 2x_{dd}^k) & \geq n - 2x_M - x_A - x_B \\
\sum_{ij} (x_{ij}^P + x_{ij}^{MP}) & \geq x_B/2 \\
\sum_{ij} (\ell_{ij}^M x_{ij}^M + \ell_{ij}^{MP} x_{ij}^{MP}) & \geq x_M \\
x_A + x_B & \geq |V(F)| \\
2x_M + x_A + x_B & \leq n \\
x & \geq 0 \\
x_{ik}^k & = 0 \text{ para } (ij, k) \in \mathcal{Z}.
\end{array}
\quad
\begin{array}{l}
\ell_{ac}^{MP} = \lceil (2 - 2\alpha)/(2\alpha - 1) \rceil \\
\ell_{ad}^{MP} = \lceil (2 - 3\alpha)/(2\alpha - 1) \rceil_+ \\
\ell_{bb}^M = \lceil (3 - 2\alpha)/(2\alpha - 1) \rceil \\
\ell_{bc}^M = \lceil (4 - 2\alpha)/(2\alpha - 1) \rceil \\
\ell_{bc}^{MP} = \lceil (2 - 2\alpha)/(2\alpha - 1) \rceil \\
\ell_{bd}^M = \lceil (4 - 3\alpha)/(2\alpha - 1) \rceil \\
\ell_{bd}^{MP} = \lceil (2 - 3\alpha)/(2\alpha - 1) \rceil_+ \\
\ell_{cc}^M = \lceil (5 - 2\alpha)/(2\alpha - 1) \rceil \\
\ell_{cd}^M = \lceil (5 - 3\alpha)/(2\alpha - 1) \rceil.
\end{array}$$

Los ℓ_{ij}^k restantes son 0 y $\mathcal{Z} = \{(aa, M), (aa, MP), (aa, P), (ab, M), (ab, MP), (ac, M), (ac, R), (ad, M), (ad, R), (bb, MP), (cc, P), (cc, MP), (cd, P), (cd, MP), (dd, M), (dd, P), (dd, MP), (dd, R)\}$.

Figura 4.2: Programa lineal $PL(n, |V(F)|, \alpha)$. El subíndice + indica que es la parte positiva del número.

Notemos que n , $|V(F)|$ y todos los $\ell_{ij}^k = \ell_{ij}^k(\alpha)$ son las únicas constantes en el PL, por lo que el valor solo depende de n , $|V(F)|$ y de α . La deducción de los valores de ℓ_{ij}^k y la demostración de que $\text{OPT}_{ij}^k = \emptyset$ para todo $(ij, k) \in \mathcal{Z}$ se ve en el siguiente lema.

Lema 4.2

$\text{OPT}_{ij}^k = \emptyset$ para todo $(ij, k) \in \mathcal{Z}$. Además, para $k \in \{M, MP\}$, existen valores ℓ_{ij}^k tales que cada link en OPT_{ij}^k solo puede cruzar ℓ_{ij}^k links de M . Estos valores serán computados en la demostración de este lema.

Asimismo, las variables x_{ij}^R para todo $ij \neq aa$ y las variables x_{ac}^P , x_{bc}^P y x_{bd}^P pueden tomarse iguales a cero en cualquier solución óptima del programa lineal.

DEMOSTRACIÓN. Haremos esto en dos etapas: en primer lugar, analizaremos todas las posibles combinaciones de ij , donde $i, j \in \{A, B, C, D\}$, y, en segundo lugar, demostraremos que en una solución óptima es posible escoger varias variables como cero.

Paso 1: Análisis exhaustivo de las variables del PL

1. $ij = aa$: si un link e con ambos extremos en A cruza una arista $m \in M$ o algun $p \in P$, entonces ambos extremos de e están en componentes circulares distintas y e cruza ambas. Esto no puede pasar por el lema 3.8. Por ende, $\text{OPT}_{aa}^M = \text{OPT}_{aa}^P = \text{OPT}_{aa}^{MP} = \emptyset$.
2. $ij = ab$: si un link e con extremos en A y en B cruza una arista $m \in M$, deducimos que ambos extremos están en componentes circulares diferentes y e cruza exactamente una de ellas (no puede cruzar dos por el lema 3.8); pero, en ese caso, e debe cruzar a una cuerda de borde $p \in P$. Concluimos que $\text{OPT}_{ab}^M = \emptyset$. Sin embargo, e no puede cruzar tanto a un link de M como a uno de P , pues, por el lema 3.10, tenemos que $X(e) = 1$, $V_F(e) = 2$ y $V_M(e) = 0$, lo que nos lleva a que e no pueda conectar más que $\text{máx} \left\{ \left\lceil \frac{1-2\alpha}{2\alpha-1} \right\rceil, 0 \right\} = 0$ links de M . Por ende, $\text{OPT}_{ab}^{MP} = \emptyset$.
3. $ij = ac$: cualquier link con extremos en A y C debe cruzar P ; luego, $\text{OPT}_{ac}^M = \text{OPT}_{ac}^R = \emptyset$. Conforme al lema 3.10, con $X(e) = 1$, $V_F(e) = 1$ y $V_M(e) = 1$, obtenemos que cualquier link en OPT_{ac}^{MP} conecta a lo más $\ell_{ac}^{MP} := \left\lceil \frac{2-2\alpha}{2\alpha-1} \right\rceil$ links de M .
4. $ij = ad$: cualquier link con extremos en A y D debe cruzar P , en consecuencia, $\text{OPT}_{ad}^M = \text{OPT}_{ad}^R = \emptyset$. Conforme al lema 3.10, con $X(e) = 1$, $V_F(e) = 1$ y $V_M(e) = 0$, obtenemos que cualquier link en OPT_{ad}^{MP} conecta a lo más $\ell_{ad}^{MP} := \left\lceil \frac{2-3\alpha}{2\alpha-1} \right\rceil_+$ links de M .
5. $ij = bb$: un link e con ambos extremos en B no puede cruzar M y P al mismo tiempo, pues, si usamos el lema 3.10 y consideramos que $X(e) = 1$, $V_F(e) = 2$ y $V_M(e) = 0$, el link conectaría a lo más $\text{máx} \left\{ \left\lceil \frac{1-2\alpha}{2\alpha-1} \right\rceil, 0 \right\} = 0$ links de M . Por ende, $\text{OPT}_{bb}^{MP} = \emptyset$. Un link e que solo cruza M conecta a lo más $\ell_{bb}^M = \left\lceil \frac{3-2\alpha}{2\alpha-1} \right\rceil$ links de M , conforme al lema 3.10 y con $X(e) = 0$, $V_F(e) = 2$ y $V_M(e) = 0$.
6. $ij = bc$: si usamos el lema 3.10 y consideramos que $X(e) = 0$, $V_F(e) = 1$ y $V_M(e) = 1$, obtenemos que los links con extremos en B y C que no cruzan P conectan a lo más $\ell_{bc}^M = \left\lceil \frac{4-2\alpha}{2\alpha-1} \right\rceil$ links de M . Si $X(e) = 1$, $V_F(e) = 1$ y $V_M(e) = 1$, notamos que los links con extremos en B y C que cruzan P conectan a lo más $\ell_{bc}^{MP} = \left\lceil \frac{2-2\alpha}{2\alpha-1} \right\rceil$ links de M .
7. $ij = bd$: cuando usamos el lema 3.10 y asignamos que $X(e) = 0$, $V_F(e) = 1$ y $V_M(e) = 0$, tenemos que links con extremos en B y D que no cruzan P conectan a lo más $\ell_{bd}^M = \left\lceil \frac{4-3\alpha}{2\alpha-1} \right\rceil$ links de M . Si $X(e) = 1$, $V_F(e) = 1$ y $V_M(e) = 0$, obtenemos que los links con extremos en B y D que cruzan P conectan a lo más $\ell_{bd}^{MP} = \left\lceil \frac{2-3\alpha}{2\alpha-1} \right\rceil_+$ links de M .
8. $ij = cc$: un link con ambos extremos en C no puede cruzar P por el lema 3.8, por lo tanto, $\text{OPT}_{cc}^P = \text{OPT}_{cc}^{MP} = \emptyset$. También notamos que, conforme al lema 3.6 y con $X(e) = 0$, $V_F(e) = 0$ y $V_M(e) = 2$, esos links pueden conectar a lo más $\ell_{cc}^M = \left\lceil \frac{5-2\alpha}{2\alpha-1} \right\rceil$ links de M .
9. $ij = cd$: un link con un extremo en C y otro en D no puede cruzar P por el lema 3.8, por ende, $\text{OPT}_{cd}^P = \text{OPT}_{cd}^{MP} = \emptyset$. También notamos, gracias al lema 3.6 y si consideramos que $X(e) = 0$, $V_F(e) = 0$ y $V_M(e) = 1$, que esos links pueden conectar a lo más $\ell_{cd}^M = \left\lceil \frac{5-3\alpha}{2\alpha-1} \right\rceil$ links de M .
10. $ij = dd$: $\text{OPT}_{dd} = \emptyset$, pues si tuviéramos un link así lo hubiéramos agregado al matching maximal M .

Paso 2: Nuevas simplificaciones a través de análisis de las soluciones del problema lineal

Definamos nuevas soluciones con el mismo valor objetivo para que ciertas variables valgan cero; estas nuevas soluciones seguirán satisfaciendo las restricciones. Así que llamemos x^* , x_A^* , x_B^* y x_M^* a una solución óptima del PL.

- $x_{cd}^R = 0$: si $x_{cd}^{R*} = \varepsilon > 0$. Entonces construyamos una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{cd}^{R'} = x_{cd}^{R*} - \varepsilon$, $x_{cd}^{M'} = x_{cd}^{M*} + \varepsilon$, lo que mantiene factibilidad y valor objetivo.
- $x_{bd}^R = 0$: si $x_{bd}^{R*} = \varepsilon > 0$. Entonces podemos construir una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{bd}^{R'} = x_{bd}^{R*} - \varepsilon$, $x_{bd}^{M'} = x_{bd}^{M*} + \varepsilon$, lo que conserva factibilidad y valor objetivo.
- $x_{bc}^R = 0$: si $x_{bc}^{R*} = \varepsilon > 0$. Entonces construyamos una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{bc}^{R'} = x_{bc}^{R*} - \varepsilon$, $x_{bc}^{M'} = x_{bc}^{M*} + \varepsilon$, lo que mantiene factibilidad y valor objetivo.
- $x_{ab}^R = 0$: si $x_{ab}^{R*} = \varepsilon > 0$. Entonces podemos construir una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{ab}^{R'} = x_{ab}^{R*} - \varepsilon$, $x_{aa}^{R'} = x_{aa}^{R*} + \varepsilon/2$, $x_{bb}^{M'} = x_{bb}^{M*} + \varepsilon/4$, $x_{bb}^{R'} = x_{bb}^{R*} + \varepsilon/4$, lo que conserva factibilidad y valor objetivo.
- $x_{bb}^R = 0$: si $x_{bb}^{R*} = \varepsilon > 0$. Entonces construyamos una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{bb}^{R'} = x_{bb}^{R*} - \varepsilon$, $x_{bb}^{M'} = x_{bb}^{M*} + \varepsilon$, lo que mantiene factibilidad y valor objetivo.
- $x_{ac}^P = 0$: si $x_{ac}^{P*} = \varepsilon > 0$. Entonces podemos construir una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{ac}^{P'} = x_{ac}^{P*} - \varepsilon$, $x_{ac}^{MP'} = x_{ac}^{MP*} + \varepsilon$, lo que conserva factibilidad y valor objetivo.
- $x_{bc}^P = 0$: si $x_{bc}^{P*} = \varepsilon > 0$. Entonces construyamos una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{bc}^{P'} = x_{bc}^{P*} - \varepsilon$, $x_{bd}^{P'} = x_{bd}^{P*} + \varepsilon$, lo que mantiene factibilidad y valor objetivo.
- $x_{bd}^P = 0$: si $x_{bd}^{P*} = \varepsilon > 0$. Entonces podemos construir una nueva solución (x', x'_A, x'_B, x'_M) definiendo $x_{bd}^{P'} = x_{bd}^{P*} - \varepsilon$, $x_{ad}^{P'} = x_{ad}^{P*} + \varepsilon$, $x'_A = x_A^* + \varepsilon$, $x'_B = x_B^* - \varepsilon$, lo que conserva factibilidad y valor objetivo.
- $x_{cc}^R = 0$, porque no está en ninguna restricción. □

El programa lineal que obtenemos luego de todas estas simplificaciones es el siguiente:

$$\begin{aligned}
 \text{(PL)} \quad & \text{mín } x_M + x_{aa}^R + x_{ab}^P + x_{ac}^{MP} + x_{ad}^P + x_{ad}^{MP} \\
 & + x_{bb}^M + x_{bb}^P + x_{bc}^M + x_{bc}^{MP} + x_{bd}^M + x_{bd}^{MP} + x_{cc}^M + x_{cd}^M \\
 & 2x_{aa}^R + x_{ab}^P + x_{ac}^{MP} + x_{ad}^P + x_{ad}^{MP} \geq x_A \\
 x_{ab}^P + 2x_{bb}^P + 2x_{bb}^M + x_{bc}^M + x_{bc}^{MP} + x_{bd}^M + x_{bd}^{MP} & \geq x_B \\
 x_{ac}^{MP} + x_{bc}^M + x_{bc}^{MP} + x_{cc}^M + x_{cd}^M & \geq n - 2x_M - x_A - x_B \\
 x_{ab}^P + x_{ac}^{MP} + x_{ad}^P + x_{ad}^{MP} + x_{bb}^P + x_{bc}^{MP} + x_{bd}^{MP} & \geq x_B/2 \\
 \ell_{ac}^{MP} x_{ac}^{MP} + \ell_{ad}^{MP} x_{ad}^{MP} + \ell_{bb}^M x_{bb}^M + \ell_{bc}^M x_{bc}^M + \ell_{bc}^{MP} x_{bc}^{MP} \\
 + \ell_{bd}^M x_{bd}^M + \ell_{bd}^{MP} x_{bd}^{MP} + \ell_{cc}^M x_{cc}^M + \ell_{cd}^M x_{cd}^M & \geq x_M \\
 x_A + x_B & \geq |V(F)| \\
 2x_M + x_A + x_B & \leq n \\
 x & \geq 0
 \end{aligned}$$

Lema 4.3

La asignación $x_{ij}^k = |\text{OPT}_{ij}^k|$, $x_A = |A|$, $x_B = |B|$ y $x_M = |M|$ es una solución factible para $PL(n, |V(F)|, \alpha)$.

DEMOSTRACIÓN. Las primeras 3 restricciones se cumplen de inmediato pues A, B y D deben ser cubiertas por links en OPT y $|D| = n - 2|M| - |A| - |B|$. La cuarta restricción está dada por que las cuerdas de P deben ser cruzadas por algún link en $\text{OPT} \setminus M$, pero un link cualquiera de $\text{OPT} \setminus M$ puede cruzar a lo más una cuerda en P por el lema 3.6; por ende, el lado izquierdo es al menos $|P|$, lo cual, a su vez, por el lema 3.5, es al menos $|B|/2$. La quinta restricción se cumple porque todos los links de M deben estar conectados al resto de la solución, pero el número de links que un link de $S \setminus F$ puede cruzar está acotado gracias al lema 3.10. Las últimas dos restricciones se satisfacen pues $|A| + |B| = |V(F)|$ y $2|M| + |A| + |B| = n - |D| \leq n$. \square

Para $1/2 < \alpha \leq 1$ y $0 \leq x \leq 1$, definimos

$$\begin{aligned} R(\alpha) &= 3 + 3\ell_{cd}^M(\alpha), & W(x, \alpha) &= (R(\alpha) - S(\alpha)x)/T(\alpha), \\ S(\alpha) &= 3 + 2\ell_{bd}^M(\alpha) + \ell_{ad}^{MP}(\alpha), & f_\alpha &= (2R(\alpha) - T(\alpha))/2S(\alpha) = 3/2S(\alpha), \\ T(\alpha) &= 3 + 6\ell_{cd}^M(\alpha), & \psi_\alpha(x) &= (3 - x)/W(x, \alpha). \end{aligned}$$

Notemos que $W(x, \alpha)$ es decreciente en α y f_α está definido de tal forma que $W(f_\alpha, \alpha) = \frac{2R(\alpha) - 3}{2T(\alpha)} = \frac{3 + 6\ell_{cd}^M(\alpha)}{2(3 + 6\ell_{cd}^M(\alpha))} = 1/2$, lo que implica que $\psi_\alpha(f_\alpha) = \frac{3 - f_\alpha}{W(f_\alpha, \alpha)} = 6 - 2f_\alpha$. A partir de que $x \leq \lceil x \rceil \leq x + 1$, observamos que $3S(\alpha) - R(\alpha) = 6 + 6\ell_{bd}^M + 3\ell_{ad}^{MP} - 3\ell_{cd}^M \geq 6 + 6\frac{4-3\alpha}{2\alpha-1} + 3\frac{2-3\alpha}{2\alpha-1} - 3\frac{4-\alpha}{2\alpha-1} = 12\frac{1-\alpha}{2\alpha-1} > 0$ para todo $\alpha \in (\frac{1}{2}, 1)$. De este modo, la función $\psi_\alpha(x)$ satisface $\psi'_\alpha(x) = \frac{T(\alpha)^2}{(R(\alpha) - S(\alpha)x)^2} \left(\frac{S(\alpha)x - R(\alpha)}{T(\alpha)} + \frac{(3-x)S(\alpha)}{T(\alpha)} \right) = \frac{T(\alpha)}{(R(\alpha) - S(\alpha)x)^2} (3S(\alpha) - R(\alpha)) \geq 0$, por lo tanto, es creciente. Esto nos permite probar el siguiente lema y encontrar mejores órdenes de aproximación.

Lema 4.4

Sea F un conjunto (α, N_{\max}) -crítico con $N_{\max} > \ell_{cc}^M$, el valor óptimo de $PL(n, |V(F)|, \alpha)$ es $\max\{n/2, nW(|V(F)|/n, \alpha)\}$. En particular, esto es igual a $n/2$ si y solo si $|V(F)|/n \geq f_\alpha$.

DEMOSTRACIÓN. Para esto, estudiaremos el problema dual, dado por 4.3, y obtendremos puntos factibles con la ayuda de *solvers* que nos proporcionan el mismo valor objetivo. Esto último, por dualidad débil, nos garantiza que el problema primal obtiene un mínimo para la solución óptima.

Observamos que cualquier solución que es factible para el primal (LP) es también factible cuando el valor de $|V(F)|$ decrece. Luego tenemos que el conjunto factible de (LP) crece cuando $|V(F)|$ decrece, por lo tanto, el óptimo de (LP) debe ser una función decreciente en $|V(F)|$.

$$\begin{aligned}
(\text{DL}) \quad & \text{máx } y_3(n - |V(F)|) + y_6|V(F)| - y_7n \\
& 2y_1 \leq 1 & y_2 + \ell_{bc}^M y_5 \leq 1 \\
& 2y_2 + \ell_{bb}^M y_5 \leq 1 & y_1 + 2y_4 + \ell_{ac}^{MP} y_5 \leq 1 \\
& y_1 + y_3 + 2y_4 \leq 1 & y_1 + y_3 + 2y_4 + \ell_{ad}^{MP} y_5 \leq 1 \\
& y_1 + y_2 + 2y_4 \leq 1 & y_2 + y_3 + 2y_4 + \ell_{ad}^{MP} y_5 \leq 1 \\
& 2y_2 + 2y_4 \leq 1 & 2y_3 - y_5 - 2y_7 \leq 1 \\
& y_2 + y_3 + \ell_{bd}^M y_5 \leq 1 & -y_1 + y_6 - y_7 \leq 0 \\
& y_2 + \ell_{bc}^M y_5 \leq 1 & -y_2 - y_4 + y_6 - y_7 \leq 0 \\
& y_3 + \ell_{cd}^M y_5 \leq 1 & y_1, y_2, y_3, y_4, y_5, y_6, y_7 \geq 0 \\
& \ell_{cc}^M y_5 \leq 1
\end{aligned}$$

Figura 4.3: Programa lineal DL($n, |V(F)|, \alpha$)

El comportamiento de (LP) depende de si $|V(F)|$ está en $I_1 = \left[0, \frac{3n}{2\ell_{ad}^{MP} + 4\ell_{bd}^M + 6}\right)$ o en $I_2 = \left[\frac{3n}{2\ell_{ad}^{MP} + 4\ell_{bd}^M + 6}, n\right]$. La forma de la solución también depende de si $\alpha \in (1/2, 2/3]$ o $\alpha > 2/3$.

En la tabla 4.1 y en la primera columna de la tabla 4.2, describimos algunas soluciones primales y duales para cuando $|V(F)| \in I_1$. Es fácil verificar que los puntos son factibles y el valor objetivo común equivale a $\frac{(3 + 3\ell_{cd}^M)n - (3 + 2\ell_{bd}^M + \ell_{ad}^{MP})|V(F)|}{3 + 6\ell_{cd}^M} = nW\left(\frac{|V(F)|}{n}, \alpha\right)$.

Por una parte, notemos que cuando $|V(F)|$ está en el extremo derecho de I_1 , el valor del PL equivale a $n/2$. Por otra parte, si $|V(F)| \in I_2$, el valor óptimo de PL es a lo más $n/2$. En la segunda columna de la tabla 4.2, mostramos un punto dual factible para $|V(F)| \in I_1 \cup I_2$ de valor $n/2$. Por dualidad débil, el valor del primal es siempre más grande que el valor de cualquier solución dual factible y, como el óptimo es decreciente en $|V(F)|$, concluimos que para $|V(F)| \in I_2$, (PL) tiene valor exactamente $n/2$. \square

Lema 4.5

Sea F un conjunto (α, N_{\max}) -crítico con $N_{\max} > \ell_{cc}^M$, se cumple que $\text{alg}(F)/\text{opt} \leq 2 - 2(1 - \alpha)f_\alpha$. Más aún, la expresión es minimizada en $\alpha = 8/11$, así que el orden de aproximación es de $233/121 \approx 1,92562$.

DEMOSTRACIÓN. Por el lema 3.3, $\text{alg}(F) < \alpha f + (1 - f) = (1 - \alpha)(3 - f) + 3\alpha - 2$. Consideremos que $\text{opt} \geq \max\{1/2, W(f, \alpha)\}$ por el lema 4.4 y que $\psi_\alpha(\cdot)$ es creciente, entonces tenemos

para $f \leq f_\alpha$,

$$\begin{aligned} \frac{\text{alg}(F)}{\text{opt}} &< (1 - \alpha) \frac{3 - f}{\text{opt}} + \frac{3\alpha - 2}{\text{opt}} \leq (1 - \alpha) \frac{3 - f}{W(f, \alpha)} + \frac{3\alpha - 2}{\text{opt}} \\ &= (1 - \alpha) \psi_\alpha(f_\alpha) + \frac{(3\alpha - 2)}{\text{opt}} \leq (1 - \alpha) \psi_\alpha(f_\alpha) + 2(3\alpha - 2) = 2 - 2(1 - \alpha)f_\alpha. \end{aligned}$$

Por otro lado, si $f \geq f_\alpha$, $\frac{\text{alg}(F)}{\text{opt}} \leq 2\text{alg}(F) < 2(1 - f(1 - \alpha)) \leq 2 - 2(1 - \alpha)f_\alpha$.

Ahora, observamos que la función $f_\alpha = 3/(6 + 4\ell_{bd}^M(\alpha) + 2\ell_{ad}^{MP}(\alpha)) = \frac{3}{(6 + 4\lceil \frac{4-3\alpha}{2\alpha-1} \rceil + 2\lceil \frac{2-3\alpha}{2\alpha-1} \rceil)}$ es una función escalonada creciente con saltos en el conjunto infinito numerable \mathcal{A} , en el cual se cumple que $(2 - 3\alpha)/(2\alpha - 1)$ o $(4 - 3\alpha)/(2\alpha - 1)$ son enteros. Luego, $(1 - \alpha)f_\alpha$ es el área del rectángulo maximal con base $[\alpha, 1]$ que está bajo la curva de f_α . La garantía que obtenemos es 2 menos el doble del área, que es minimizada en el punto $\alpha^* = 8/11 \in \mathcal{A}$, tal como se ve en la figura 4.4. \square

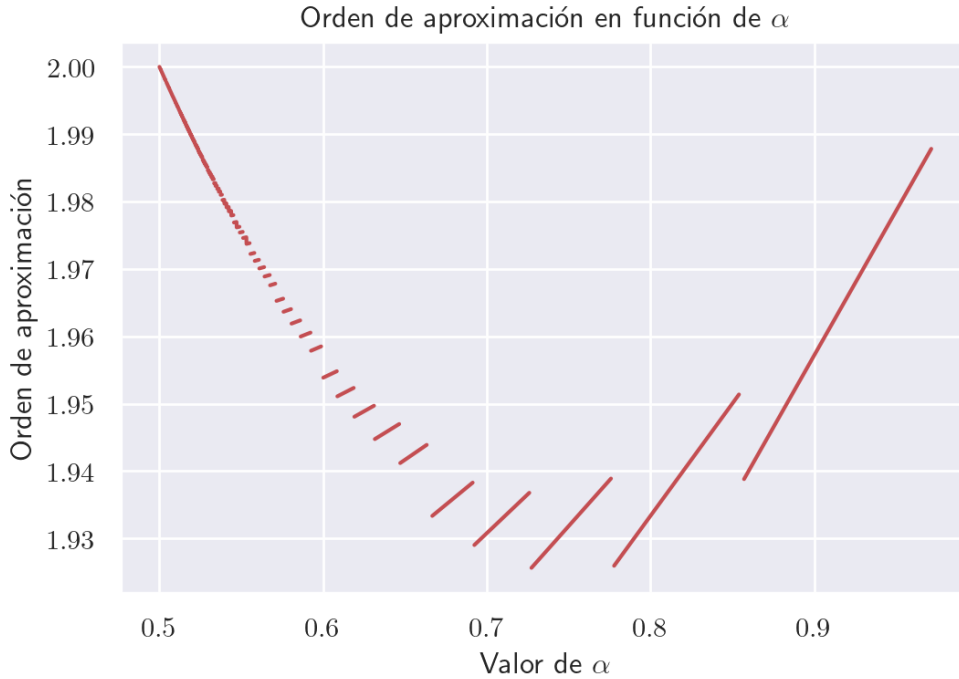


Figura 4.4: Grafico de la función $h(\alpha) = 2 - 2(1 - \alpha)f_\alpha$ que acota superiormente el orden de aproximación.

Variable	Punto óptimo para $I_1(\alpha > 2/3)$	Punto óptimo para $I_1(\alpha \leq 2/3)$
x_A	$\frac{1}{3} V(F) $	$\frac{1}{3} V(F) $
x_B	$\frac{2}{3} V(F) $	$\frac{2}{3} V(F) $
x_M	$x_M^{(1)}$	$x_M^{(1)}$
x_{ad}^P	$\frac{1}{3} V(F) $	0
x_{ad}^{MP}	0	$\frac{1}{3} V(F) $
x_{bd}^M	$\frac{2}{3} V(F) $	$\frac{2}{3} V(F) $
x_{cd}^M	$x_{cd}^{M(1)}$	$x_{cd}^{M(1)}$
Otras	0	0

Tabla 4.1: Soluciones óptimas para el PL en función de $|V(F)|$, donde denotamos $X_M^{(1)} = \frac{3\ell_{cd}^M n + (\ell_{ad}^{MP} + 2\ell_{bd}^M - 6\ell_{cd}^M)|V(F)|}{3 + 6\ell_{cd}^M}$ y $x_{cd}^{M(1)} = \frac{3n - |V(F)|(2\ell_{ad} + 4\ell_{bd}^M + 6)}{3 + 6\ell_{cd}^M}$.

Variable	Punto óptimo para $ V(F) \in I_1$	Punto factible para $ V(F) \in I_1 \cup I_2$
y_1	$\frac{3\ell_{cd}^M - 2\ell_{bd}^M - \ell_{ad}^{MP}}{3(2\ell_{cd}^M + 1)}$	$\frac{1}{2}$
y_2	$\frac{\ell_{cd}^M - \ell_{bd}^M}{2\ell_{cd}^M + 1}$	$\frac{1}{2}$
y_3	$\frac{\ell_{cd}^M + 1}{2\ell_{cd}^M + 1}$	$\frac{1}{2}$
y_4	$\frac{\ell_{bd}^M - \ell_{ad}^{MP}}{3(2\ell_{cd}^M + 1)}$	0
y_5	$\frac{1}{2\ell_{cd}^M + 1}$	0
y_6	$\frac{3\ell_{cd}^M - 2\ell_{bd}^M - \ell_{ad}^{MP}}{3(2\ell_{cd}^M + 1)}$	$\frac{1}{2}$
y_7	0	0

Tabla 4.2: Soluciones óptimas del problema dual (DL).

Capítulo 5

Un algoritmo de búsqueda local refinado

La primera fase de nuestro algoritmo termina cuando F es un conjunto (α, N_{\max}) -crítico. Si aumentamos un poco el valor de α al final de la fase, podemos encontrar un conjunto F' con utilidad más grande que F y, por lo tanto, obtener una mejor solución. Podemos aprovecharnos de esta intuición y diseñar un algoritmo de búsqueda local refinado. Esto nos permite probar una versión generalizada del lema 4.5 y demostrar nuestro resultado principal.

Algoritmo 3 Algoritmo de búsqueda local refinado

Input: Instancia (C_n, S) del problema. $N_{\max} \in \mathbb{N}$. Lista de valores $A = \{\alpha_1, \alpha_2, \dots, \alpha_s, \alpha_{s+1}\}$, donde $1/2 < \alpha_1 < \alpha_2 < \dots < \alpha_s \leq \alpha_{s+1} = 1$.

- 1: $F_0 \leftarrow \emptyset$.
 - 2: **para** cada $i \in [s]$ **hacer**
 - 3: A partir del conjunto actual F_{i-1} , correr iteraciones de la primera fase del algoritmo de búsqueda local con $\alpha = \alpha_i$ para encontrar un conjunto (α_i, N_{\max}) -crítico F_i que contenga a F_{i-1} .
 - 4: **fin para**
 - 5: Encontrar una completación minimal Q de F_s
 - 6: **devolver** (Q, F_s) .
-

Lema 5.1

Sea ALG el output del algoritmo de búsqueda local refinado para $N_{\max} \geq \ell_{cc}(\alpha_1) \geq \ell_{cc}(\alpha_i) \forall i \in [s]$, se tiene $|\text{ALG}|/|\text{OPT}| \leq 2 - 2 \sum_{i=2}^{m+1} (\alpha_i - \alpha_{i-1}) f_{\alpha_i}$.

DEMOSTRACIÓN. Sean los conjuntos F_0, F_1, \dots, F_s creados por el algoritmo de búsqueda local refinado y $f^i = |V(F_i)|/n$ para todo $i \in [s]$, F_i se obtiene desde F_{i-1} iterativamente agregándole conjuntos de links tal que la utilidad marginal de esos conjuntos es al menos $(1 - \alpha_i)$ el número de vértices nuevos agregados. Concluimos que $U(F_i) - U(F_{i-1}) \geq (1 - \alpha_i)(|V(F_i)| - |V(F_{i-1})|)$. Ya que $U(F_0) = 0 = |V(F_0)|$ y que $\alpha_{s+1} = 1$, concluimos que $U(F_s) = \sum_{j=1}^s U(F_j) - U(F_{j-1}) \geq \sum_{j=1}^s (1 - \alpha_j)(|V(F_j)| - |V(F_{j-1})|) = n \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) f^j$.

Dada la observación anterior, que $3n = 3n\alpha_1 + 3n \sum_{j=1}^s (\alpha_{j+1} - \alpha_j)$, que $2\text{opt} \geq 1$ y el lema 3.3, también vemos que

$$\begin{aligned} \frac{|\text{ALG}|}{|\text{OPT}|} &< \frac{-2n + (3n - U(F_s))}{n \text{opt}} \leq \frac{-2n + (3n - n \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) f^j)}{n \text{opt}} \\ &= \frac{-2n + 3n\alpha_1}{n \text{opt}} + \frac{3n \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) - n \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) f^j}{n \text{opt}} \\ &\leq 2(-2 + 3\alpha_1) + \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) \frac{(3 - f^j)}{\text{opt}}. \end{aligned}$$

Por un lado, notemos que si $f^j \geq f_{\alpha_j}$, entonces $(3 - f^j)/\text{opt} \leq (3 - f_{\alpha_j})/\text{opt} \leq 6 - 2f_{\alpha_j}$. Por otro lado, si $f^j \leq f_{\alpha_j}$, como $f^j = |V(F_j)|/n$ y F_j es $(\alpha_j, N_{\text{máx}})$ -crítico, por el lema 4.4, deducimos que $\text{opt} \geq W(f^j, \alpha_j)$ y, luego, $(3 - f^j)/\text{opt} \leq \psi_\alpha(f^j) \leq \psi_\alpha(f_{\alpha_j}) = 6 - 2f_{\alpha_j}$.

Ahora usemos estas cotas en la expresión para $|\text{ALG}|/|\text{OPT}|$ y también advirtamos que $6 = 6\alpha_1 + 6 \sum_{j=1}^s (\alpha_{j+1} - \alpha_j)$ para concluir que

$$\frac{|\text{ALG}|}{|\text{OPT}|} \leq (6\alpha_1 - 4) + \sum_{j=1}^s (\alpha_{j+1} - \alpha_j)(6 - 2f_{\alpha_j}) = 2 - 2 \sum_{j=1}^s (\alpha_{j+1} - \alpha_j) f_{\alpha_j}. \quad \square$$

Teorema 5.2

Si $\varepsilon > 0$, existe una $(1,87029 + \varepsilon)$ -aproximación para el problema de aumentación de vértice-conectividad del ciclo.

DEMOSTRACIÓN. Recordemos que $f_\alpha = \frac{3}{(6+4\lceil \frac{4-3\alpha}{2\alpha-1} \rceil + 2\lceil \frac{2-3\alpha}{2\alpha-1} \rceil_+)}$ es una función de α creciente y escalonada con saltos en el conjunto infinito numerable \mathcal{A} , en el cual se cumple que $(2 - 3\alpha)/(2\alpha - 1)$ o $(4 - 3\alpha)/(2\alpha - 1)$ son enteros. La función $\sum_{j=1}^s (\alpha_{j+1} - \alpha_j) f_{\alpha_j}$ es el área de la aproximación inferior con triángulos cuyas bases son todos los intervalos $[\alpha_j, \alpha_{j+1}]$. El área máxima que podemos obtener es exactamente el área $\int_{1/2}^1 f_\alpha d\alpha$ bajo la curva f_α para $\alpha \in (1/2, 1]$. Sin embargo, necesitamos usar solo una cantidad finita de puntos de \mathcal{A} . Además, tampoco podemos usar puntos arbitrariamente cercanos a $1/2$ pues estos harían que $N_{\text{máx}}$ sea arbitrariamente grande, lo que afectaría el tiempo de ejecución del algoritmo. En consecuencia, para todo $\varepsilon > 0$, podemos lograr, en tiempo $n^{O(1/\varepsilon)}$, una garantía de $2 + \varepsilon - 2 \int_{1/2}^1 f_\alpha d\alpha \approx 1,87029 + \varepsilon$, puesto que se hacen $O(1/\varepsilon)$ iteraciones de la fase 1 y cada iteración toma $|S|^{O(N_{\text{máx}})} = n^{O(1/\varepsilon)}$. \square

Capítulo 6

Formulación como problema de programación lineal

En este capítulo se discutirán algunas propiedades de dos formulaciones lineales del problema: una formulación natural y la otra menos evidente, pero con más restricciones.

6.1. Primer programa lineal

Por el lema 2.3, sabemos que, en una solución factible, todas las cuerdas deben ser cruzadas por un link de la solución. Definamos primeramente la variable x_{kl} , que indica si se usa la cuerda $kl \in S$ en una solución para el problema. Ahora, se puede definir un PL natural de la siguiente forma:

$$\begin{aligned} & \text{mín} \sum_{kl \in S} x_{kl} \\ & \text{sujeto a} \sum_{\substack{kl \in S \\ kl \text{ cruza } ij}} x_{kl} \geq 1 \quad \forall 1 \leq i < j \leq n, \\ & 0 \leq x \leq 1. \end{aligned}$$

Llamaremos a la primera restricción una **desigualdad de corte**. Por el lema 2.3, si F es una solución factible, se puede ver que la asignación $x_{kl} = \begin{cases} 1 & \text{si } kl \in F \\ 0 & \text{si no} \end{cases}$ nos da una solución factible para el PL. Este programa es una relajación del problema pues estamos considerando un x entre 0 y 1 en vez de un $x \in \{0, 1\}$.

Lema 6.1

El gap de integralidad de este PL es al menos 2.

DEMOSTRACIÓN. Veamos el ejemplo de la figura 6.1. Consideremos un ciclo con n vértices y

n par, donde en S están todos los pares $(i, i + 2)$ con $i \in [n]$, formando dos ciclos Z_1 y Z_2 . Si asignamos un valor $\frac{1}{2}$ a cada arista, obtenemos una solución factible con un valor de $\frac{n}{2}$, pero notemos que solo podemos sacar una arista de cada ciclo Z_i , pues si sacamos dos de un ciclo, digamos $a = (j - 1, j + 1)$ y $b = (k - 1, k + 1)$, con j y k de la misma paridad, los dos vértices j y k que quedan descubiertos forman un par separador. Esto ocurre porque las únicas aristas que lo atravesaban en S eran a y b . Como un ciclo puede tener un número arbitrariamente alto de vértices, el valor integral es $n - 2$, pero el del PL es de $\frac{n}{2}$. Esto nos da una razón $\frac{n-2}{\frac{n}{2}} = 2 - \frac{4}{n}$, que tiende a 2 cuando n tiende a infinito. \square

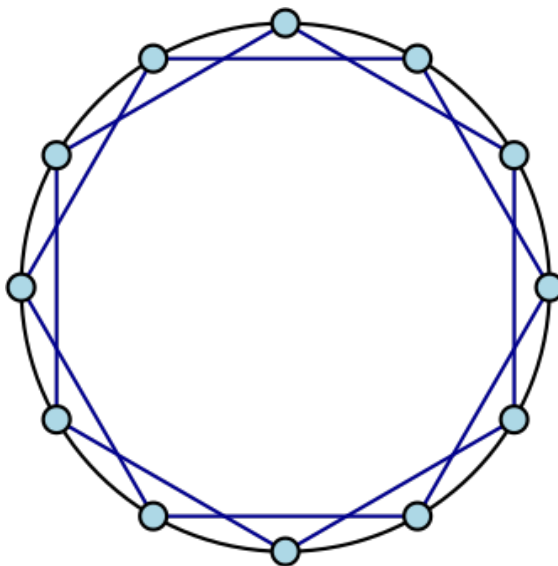


Figura 6.1: Ejemplo que muestra que el *gap* de integralidad del PL inicial es 2

6.2. Segundo programa lineal

Para obtener una mejor formulación, se le añade una segunda restricción al programa.

Definición 6.2

Sea C_n un ciclo, llamaremos **polígono de cuerdas** a un polígono $P = (v_1, \dots, v_k, v_1)$ de k vértices, donde $v_1 < v_2 < \dots < v_k$ en C_n , el cual está inscrito en el ciclo y cumple que ningún par de vértices es consecutivo en C .

Proposición 6.3

Sea F una solución factible para el problema y P un polígono de cuerdas, llamemos Γ_P al conjunto de aristas de S que atraviesan a P , luego $|\Gamma_P \cap F| \geq \left\lceil \frac{k}{2} \right\rceil$.

DEMOSTRACIÓN. Esta desigualdad proviene de las desigualdades de corte para un polígono y del hecho de que una cuerda cruza a lo más dos aristas del polígono. Por ende, la cantidad de aristas que cruzan un polígono es mayor a $\frac{k}{2}$. El cajon superior se agrega porque necesitamos que lo cruce un número entero de aristas. \square

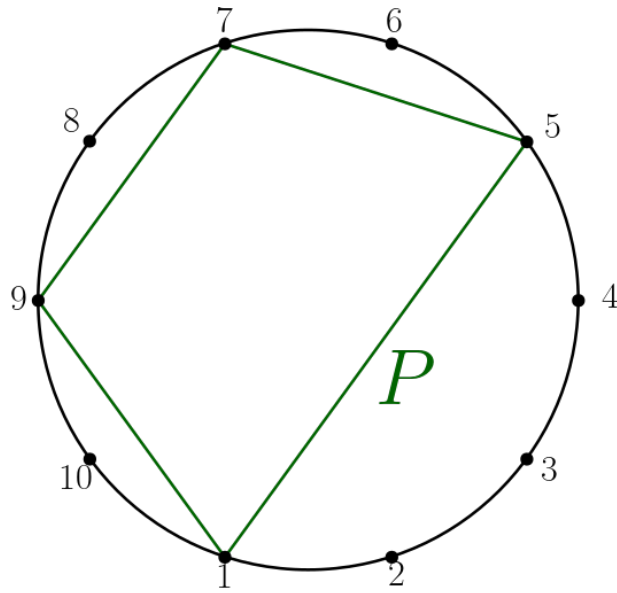


Figura 6.2: Ejemplo de un polígono de cuerdas P de 4 vértices definido por los vértices $(1, 5, 7, 9)$

Como estas desigualdades incluyen a las desigualdades de cuerdas cuando $k = 2$, podemos definir un nuevo PL que incluye las **desigualdades de polígonos** de la siguiente forma:

$$\begin{aligned} & \text{mín } \sum_{kl \in S} x_{kl} \\ & \text{sujeto a } \sum_{kl \in \Gamma_P} x_{kl} \geq \left\lceil \frac{|P|}{2} \right\rceil \text{ para todo } P \text{ polígono de cuerdas.} \\ & 0 \leq x \leq 1. \end{aligned}$$

No es necesario considerar polígonos de todos los tamaños posibles, como lo expone el lema que enunciamos a continuación.

Lema 6.4

Para $|P| \geq 2$, basta solo considerar $|P|$ impares y $|P| = 2$. Es decir, las desigualdades de polígono para $|P|$ par mayor que 3 son redundantes.

DEMOSTRACIÓN. Sea P un polígono de $2k$ aristas, agreguemos una cuerda extra a para dividirlo en un polígono de $2k - 1$ aristas y otro de 3, tal como se muestra en la figura 6.3. Numeramos las variables asociadas a las cuerdas del polígono como x_1, \dots, x_{2k} , donde x_i representa la arista entre v_{i-1} y v_i , y definimos x' como la variable asociada a la cuerda a . Sin pérdida de generalidad, la cuerda extra a tiene extremos en v_{2k-1} y v_1 . Al usar las desigualdades de polígono para los polígonos a ambos lados de a obtenemos que $\sum_{i=1}^{2k-2} x_i + x' \geq k$ y que $x_{2k-1} + x_{2k} + x' \geq 2$. Esto implica que, para el polígono de $2k$ aristas, se cumple que $\sum_{i=1}^{2k} x_i \geq k + 2 - 2x' \geq k$, lo que nos permite concluir la proposición. \square

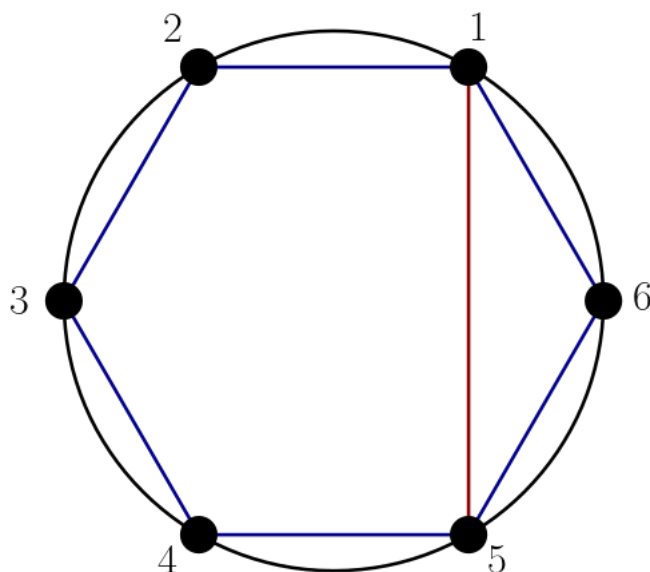


Figura 6.3: Ejemplo de un polígono de cuerdas P de 6 vértices con una cuerda entre 1 y 5, la cual forma dos polígonos de 5 y 3 lados.

Notemos que la asignación que usamos en la demostración del lema 6.1 ya no es factible en el nuevo PL en esa misma instancia. Si consideramos un polígono de cuerdas $P = (i, j, k)$ inscrito en el círculo, vemos que hay 3 links en S que cruzan a P , pero, en ese caso, la suma de los pesos de las variables asociadas da $3/2$, que es menor a 2. Esto da indicios de que la formulación es más robusta para modelar instancias estándar. Ahora, podemos demostrar que la cota inferior del *gap* de integralidad para este programa es menor a la obtenida en el lema 6.1.

Lema 6.5

El *gap* de integralidad del segundo PL es al menos $\frac{3}{2}$.

DEMOSTRACIÓN. Tomemos el mismo ejemplo del lema 6.1, pero asignemos peso $\frac{2}{3}$ a cada arista. Esta asignación cumple con todas las restricciones ya que, para un polígono cualquiera de $2k + 1$ vértices, la arista comprendida entre los vértices adyacentes a cada vértice del polígono cruza a este. Por ende, $\frac{2}{3}(2k + 1) = \frac{4k+2}{3}$, que es mayor o igual a $k + 1 = \lceil \frac{2k+1}{2} \rceil$ para $k \geq 1$. Análogo a lo hecho en el lema 6.1, conseguimos una razón $\frac{n-2}{\frac{2n}{3}} = \frac{3}{2} - \frac{3}{n}$, que tiende a $\frac{3}{2}$ cuando n tiende a infinito. \square

Lema 6.6

Las desigualdades de triángulo no implican las de pentágono, donde $|P| = 5$.

DEMOSTRACIÓN. Si consideramos la figura 6.4, con $n = 10$ y $S = K'_5 \cup K''_5$, donde K'_5 es el grafo completo entre los 5 vértices pares y K''_5 el grafo completo entre los 5 vértices impares, y asignamos a cada arista un peso de $1/4$, entonces es fácil ver que a cada cuerda la atraviesan al menos 4 links y que todos los polígonos de 3 aristas son cruzados por al

menos 8 links. Por lo tanto, se cumplen las desigualdades de triángulo. Sin embargo, los 2 polígonos de 5 aristas son atravesados por solo 10 links y, como $10/4 < 3$, no se cumplen las desigualdades para $|P| = 5$. \square

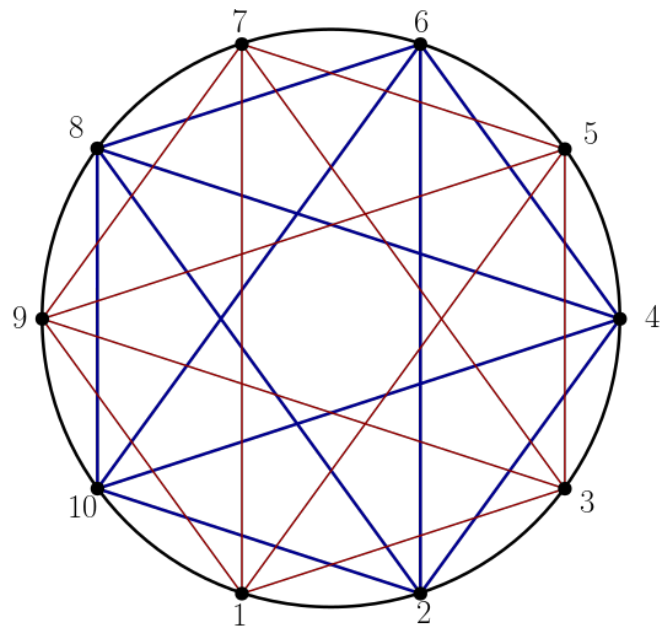


Figura 6.4: Ejemplo de instancia en donde las desigualdades de polígono con $|P| = 5$ son necesarias. Las aristas azules y rojas representan cada uno de los grafos completos K_5 que componen S .

Capítulo 7

Complejidad del problema

En este capítulo demostraremos que el problema es APX-difícil. Para esto usaremos el siguiente resultado de Gálvez *et al.* [23]. Con esto, podemos demostrar que encontrar soluciones arbitrariamente cercanas al óptimo en tiempo polinomial para nuestro problema es básicamente imposible.

Teorema 7.1 ([23])

Existe una constante $\delta_0 > 0$ tal que es NP-difícil decidir si una instancia de aumentación de arista-conectividad en un ciclo admite una solución de tamaño $\frac{n}{2}$ o si ninguna solución factible tiene tamaño a lo más $(1 + \delta_0)\frac{n}{2}$.

Teorema 7.2

Existe una constante $\delta > 0$ tal que no hay $(1 + \delta)$ -aproximación para CVCA, salvo que $\mathbf{P} = \mathbf{NP}$.

DEMOSTRACIÓN. Sea (C_n, S) una instancia para aumentación de arista-conectividad de un ciclo, demostremos que:

1. Si (C_n, S) admite una solución de tamaño $\frac{n}{2}$, entonces la misma instancia interpretada como una instancia para el CVCA admite una solución de tamaño $\frac{n}{2}$, y que
2. Si (C_n, S) no admite una solución factible de tamaño a lo más $(1 + \delta_0)\frac{n}{2}$, tampoco lo tiene el CVCA al recibir como input el par (C_n, S) .

Abordemos la primera afirmación usando el teorema 2.10. Así probamos que una solución de tamaño $\frac{n}{2}$ para arista-conectividad tiene una sola componente circular.

En efecto, si la solución tiene más de una componente, tal como se muestra en la figura 7.1, ningún par de componentes circulares comparte aristas de borde, pues cada vértice del ciclo es incidente a exactamente un link. Más aún, una de estas, digamos L_1 , debe ser contigua (i.e. una componente circular tal que $V(L_1)$ es un intervalo del ciclo). Luego, es posible desconectar esta componente borrando las dos aristas del ciclo incidentes a sus vértices de borde. En la figura, se trataría de las aristas $(1, 12)$ y $(6, 7)$. De este modo, concluye

la demostración.

Por último, la segunda afirmación es trivial porque 3-vértice-conectividad implica 3-arista-conectividad. \square

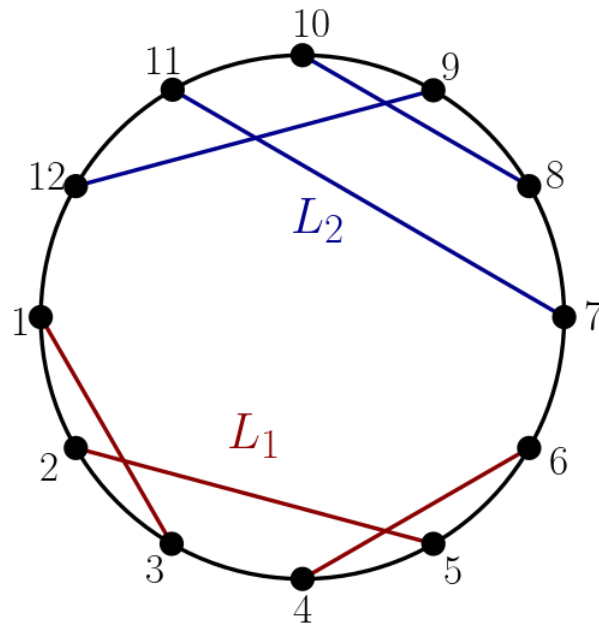


Figura 7.1: Ejemplo de cómo sería una solución de tamaño $n/2$ cuando hay más de una componente circular.

Conclusiones

Los objetivos del trabajo se cumplieron porque logramos mejorar el orden de aproximación conocido para el problema a través de un algoritmo simple, con un desarrollo que no depende de resultados anteriores. Esto nos indica que se podría disminuir el orden de aproximación mediante un análisis más sofisticado; por ejemplo, añadiendo restricciones más fuertes a la formulación lineal 4.2, lo que es posible porque hay variables que no fueron utilizadas.

En cuanto a las debilidades del algoritmo, cabe señalar que no se puede aplicar al problema donde el ciclo inicial ya tiene links añadidos y se busca encontrar la completación minimal. Pensemos en una instancia que tenga k componentes circulares, en ese caso, un óptimo podría tener solo 1 link, pero el algoritmo podría agregar $3k$ links. Por lo que la pregunta que surge es ¿cómo mejorar el algoritmo para este caso e incluso para el caso general, que consiste en encontrar una completación minimal para un grafo 2-vértice-conexo cualquiera? o, más ambiciosamente, ¿es posible mejorar las cotas obtenidas por Nutov para el k -vertex-connectivity augmentation problem, cuando $k \geq 2$? [43]

Asimismo, un algoritmo de búsqueda local similar a los que usamos en este trabajo no permite obtener un orden menor a $3/2$. Todo indica que para mejorar el orden de aproximación es necesario trabajar con técnicas más finas, como las que se han usado para el problema de arista-conectividad [6, 27]. Esas técnicas incluyen reducirse a instancias especiales del problema, analizar detalladamente los polítopos asociados y estudiar formulaciones lineales cuidadosamente diseñadas.

Otro aspecto que queda pendiente es cómo explotar la formulación lineal del problema obtenida mediante algoritmos poliedrales. De encontrar un algoritmo así, se podría estudiar el caso en que las aristas tienen pesos arbitrarios, lo que es un primer paso para mejorar el orden de 2 de Auletta *et al.* [3]. También queda pendiente encontrar una cota superior para el *gap* de integralidad, tarea que sí ha sido realizada por Gálvez *et al.* [23].

Bibliografía

- [1] Adjashvili, D. “Beating Approximation Factor Two for Weighted Tree Augmentation with Bounded Costs”. En: *ACM Trans. Algorithms* 15.2 (2019), 19:1-19:26.
- [2] Albert, R., Albert, I. y Nakarado, G. L. “Structural vulnerability of the North American power grid”. En: *Phys. Rev. E* 69.2 (feb. de 2004). doi: [10.1103/physreve.69.025103](https://doi.org/10.1103/physreve.69.025103).
- [3] Auletta, V., Dinitz, Y., Nutov, Z. y Parente, D. “A 2-Approximation Algorithm for Finding an Optimum 3-Vertex-Connected Spanning Subgraph”. En: *J. Algorithms* 32.1 (jul. de 1999), págs. 21-30. doi: [10.1006/jagm.1999.1006](https://doi.org/10.1006/jagm.1999.1006).
- [4] Basavaraju, M., Fomin, F. V., Golovach, P., Misra, P., Ramanujan, M. S. y Saurabh, S. “Parameterized Algorithms to Preserve Connectivity”. En: *ICALP 2014*. Springer Berlin Heidelberg, 2014, págs. 800-811. doi: [10.1007/978-3-662-43948-7_66](https://doi.org/10.1007/978-3-662-43948-7_66).
- [5] Byrka, J., Grandoni, F. y Ameli, A. J. “Breaching the 2-approximation barrier for connectivity augmentation: a reduction to Steiner Tree”. En: *STOC 2020*. ACM, jun. de 2020. doi: [10.1145/3357713.3384301](https://doi.org/10.1145/3357713.3384301).
- [6] Cecchetto, F., Traub, V. y Zenklusen, R. “Bridging the Gap Between Tree and Connectivity Augmentation: Unified and Stronger Approaches”. En: *STOC 2021*. 2021.
- [7] Cheriyan, J., Karloff, H., Khandekar, R. y Könemann, J. “On the integrality ratio for tree augmentation”. En: *Oper. Res. Lett.* 36.4 (jul. de 2008), págs. 399-401. doi: [10.1016/j.orl.2008.01.009](https://doi.org/10.1016/j.orl.2008.01.009).
- [8] Cheriyan, J. y Gao, Z. “Approximating (Unweighted) Tree Augmentation via Lift-and-Project, Part I: Stemless TAP”. En: *Algorithmica* 80.2 (2018), págs. 530-559.
- [9] Cheriyan, J., Jordán, T. y Ravi, R. “On 2-Coverings and 2-Packings of Laminar Families”. En: *Algorithms - ESA' 99*. Springer Berlin Heidelberg, 1999, págs. 510-520. doi: [10.1007/3-540-48481-7_44](https://doi.org/10.1007/3-540-48481-7_44).
- [10] Cheriyan, J., Vempala, S. y Vetta, A. “An Approximation Algorithm for the Minimum-Cost k -Vertex Connected Subgraph”. En: *SIAM J. Comput.* 32.4 (ene. de 2003), págs. 1050-1055. doi: [10.1137/s0097539701392287](https://doi.org/10.1137/s0097539701392287).
- [11] Chuzhoy, J. y Khanna, S. “An $O(k^3 \log n)$ -Approximation Algorithm for Vertex-Connectivity Survivable Network Design”. En: *Theory Comput.* 8.1 (2012), págs. 401-413. doi: [10.4086/toc.2012.v008a018](https://doi.org/10.4086/toc.2012.v008a018).
- [12] Cohen, N. y Nutov, Z. “A $(1 + \ln 2)(1 + \ln 2)$ -approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius”. En: *Theor. Comput. Sci.* 489-490 (2013), págs. 67-74.
- [13] Diestel, R. *Graph Theory*. 5th. Graduate Texts in Mathematics. Springer-Verlag, 2016.
- [14] Dinitz, E., Karzanov, A., y Lomonosov, M. “On the Structure of the System of Minimum Edge Cuts of a Graph”. En: *Studies Discrete Opt.* (1976), págs. 290-306.

- [15] Even, G., Feldman, J., Kortsarz, G. y Nutov, Z. "A 1.8 approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2". En: *ACM Trans. Algorithms* 5.2 (2009), 21:1-21:17.
- [16] Even, S. e Itai, A. "Queues, Stacks and Graphs". En: *Theory of Machines and Computations*. Elsevier, 1971, págs. 71-86. doi: [10.1016/b978-0-12-417750-5.50011-7](https://doi.org/10.1016/b978-0-12-417750-5.50011-7).
- [17] Fiorini, S., Groß, M., Könemann, J. y Sanità, L. "Approximating Weighted Tree Augmentation via Chvátal-Gomory Cuts". En: *SODA 2018*. 2018, págs. 817-831.
- [18] Frank, A. y Jordan, T. "Minimal Edge-Coverings of Pairs of Sets". En: *J. Combin. Theory Ser. B* 65.1 (sep. de 1995), págs. 73-110. doi: [10.1006/jctb.1995.1044](https://doi.org/10.1006/jctb.1995.1044).
- [19] Frank, A. "Augmenting Graphs to Meet Edge-Connectivity Requirements". En: *SIAM J. Discrete Math.* 5.1 (feb. de 1992), págs. 25-53. doi: [10.1137/0405003](https://doi.org/10.1137/0405003).
- [20] Frank, A. y Jordán, T. "Directed vertex-connectivity augmentation". En: *Math. Program.* 84.3 (abr. de 1999), págs. 537-553. doi: [10.1007/s101070050038](https://doi.org/10.1007/s101070050038).
- [21] Frank, A. y Tardos, É. "An application of submodular flows". En: *Linear Algebra Appl.* 114-115 (mar. de 1989), págs. 329-348. doi: [10.1016/0024-3795\(89\)90469-2](https://doi.org/10.1016/0024-3795(89)90469-2).
- [22] Frederickson, G. N. y Ja'Ja', J. "Approximation Algorithms for Several Graph Augmentation Problems". En: *SIAM J. Comput.* 10.2 (mayo de 1981), págs. 270-283. doi: [10.1137/0210019](https://doi.org/10.1137/0210019).
- [23] Gálvez, W., Grandoni, F., Ameli, A. J. y Sornat, K. "On the Cycle Augmentation Problem: Hardness and Approximation Algorithms". En: *Theory Comput. Syst.* (mar. de 2021). doi: [10.1007/s00224-020-10025-6](https://doi.org/10.1007/s00224-020-10025-6).
- [24] Garg, N. y Kumar, N. "Dual Half-integrality for Uncrossable Cut Cover and its Application to Maximum Half-Integral Flow". En: *arXiv preprint arXiv:2007.14156* (2020).
- [25] Gevigney, O. D. de y Szigeti, Z. "On $(2, k)$ -connected graphs". En: *J. Graph Theory* 91.4 (dic. de 2018), págs. 305-325. doi: [10.1002/jgt.22433](https://doi.org/10.1002/jgt.22433).
- [26] Goemans, M. X., Goldberg, A. V., Plotkin, S. A., Shmoys, D. B., Tardos, É. y Williamson, D. P. "Improved Approximation Algorithms for Network Design Problems". En: *SODA 1994*. 1994, págs. 223-232.
- [27] Grandoni, F., Kalaitzis, C. y Zenklusen, R. "Improved approximation for tree augmentation: saving by rewiring". En: *STOC 2018*. ACM Press, 2018. doi: [10.1145/3188745.3188898](https://doi.org/10.1145/3188745.3188898).
- [28] Guo, J. y Uhlmann, J. "Kernelization and complexity results for connectivity augmentation problems". En: *Networks* (2009), págs. 131-142. doi: [10.1002/net.20354](https://doi.org/10.1002/net.20354).
- [29] Haldane, A. G. y May, R. M. "Systemic risk in banking ecosystems". En: *Nature* 469.7330 (ene. de 2011), págs. 351-355. doi: [10.1038/nature09659](https://doi.org/10.1038/nature09659).
- [30] Hoersch, F. y Szigeti, Z. "2-node-connectivity network design". En: *arXiv preprint arXiv:2004.08645* (2020).
- [31] Jain, K. "A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem". En: *Combinatorica* 21.1 (ene. de 2001), págs. 39-60. doi: [10.1007/s004930170004](https://doi.org/10.1007/s004930170004).
- [32] Khuller, S. y Thurimella, R. "Approximation Algorithms for Graph Augmentation". En: *J. Algorithms* 14.2 (1993), págs. 214-225.
- [33] Kortsarz, G., Krauthgamer, R. y Lee, J. R. "Hardness of Approximation for Vertex-Connectivity Network Design Problems". En: *SIAM J. Comput.* 33.3 (ene. de 2004), págs. 704-720. doi: [10.1137/s0097539702416736](https://doi.org/10.1137/s0097539702416736).
- [34] Kortsarz, G. y Nutov, Z. "A Simplified 1.5-Approximation Algorithm for Augmenting Edge-Connectivity of a Graph from 1 to 2". En: *ACM Trans. Algorithms* 12.2 (2016), 23:1-23:20.

- [35] Kortsarz, G. y Nutov, Z. *Approximating minimum cost connectivity problems*. <https://www.openu.ac.il/home/nutov/survey-connectivity.pdf>.
- [36] Kortsarz, G. y Nutov, Z. "LP-relaxations for tree augmentation". En: *Discret. Appl. Math.* 239 (2018), págs. 94-105.
- [37] Labs, N. C. R. *Informe técnico final: Estudio y Recomendaciones sobre la resiliencia de la infraestructura de la Internet chilena*. https://www.nic.cl/estudios/resiliencia_uchile_corfo_subtel.pdf. 2018.
- [38] Mader, W. "Ecken vom Gradn in minimalenn-fach zusammenhängenden Graphen". En: *Arch. Math.* (Basel) 23.11 (1972), págs. 219-224.
- [39] Marx, D. y Végh, L. A. "Fixed-Parameter Algorithms for Minimum-Cost Edge-Connectivity Augmentation". En: *ACM Trans. Algorithms* 11.4 (jun. de 2015), págs. 1-24. doi: 10.1145/2700210.
- [40] Nagamochi, H. "An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree". En: *Discrete Appl. Math.* 126.1 (mar. de 2003), págs. 83-113. doi: 10.1016/s0166-218x(02)00218-4.
- [41] Nutov, Z. "2-node-connectivity network design". En: *arXiv preprint arXiv:2002.04048* (2020).
- [42] Nutov, Z. "Approximating Node-Connectivity Augmentation Problems". En: *Algorithmica* 63.1-2 (jun. de 2011), págs. 398-410. doi: 10.1007/s00453-011-9538-x.
- [43] Nutov, Z. "Improved Approximation Algorithms for Minimum Cost Node-Connectivity Augmentation Problems". En: *Theory Comput. Syst.* 62.3 (jun. de 2017), págs. 510-532. doi: 10.1007/s00224-017-9786-5.
- [44] Nutov, Z. "On the Tree Augmentation Problem". En: *ESA 2017*. Vol. 87. 2017, 61:1-61:14.
- [45] Petrank, E. "The hardness of approximation: Gap location". En: *Comput. Complexity* 4.2 (jun. de 1994), págs. 133-157. doi: 10.1007/bf01202286.
- [46] Ravi, R. y Williamson, D. P. "An approximation algorithm for minimum-cost vertex-connectivity problems". En: *Algorithmica* 18.1 (mayo de 1997), págs. 21-43. doi: 10.1007/bf02523686.
- [47] Ravi, R. y Williamson, D. P. "Erratum: An Approximation Algorithm for Minimum-Cost Vertex-Connectivity Problems". En: *Algorithmica* 34.1 (sep. de 2002), págs. 98-107. doi: 10.1007/s00453-002-0970-9.
- [48] Solé, R. V. y Montoya, M. "Complexity and fragility in ecological networks". En: *Proc. R. Soc. Lond. B.* 268.1480 (oct. de 2001), págs. 2039-2045. doi: 10.1098/rspb.2001.1767.
- [49] Végh, L. A. "Augmenting Undirected Node-Connectivity by One". En: *SIAM J. Discrete Math.* 25.2 (ene. de 2011), págs. 695-718. doi: 10.1137/100787507.
- [50] Végh, L. A. y Benczúr, A. A. "Primal-dual approach for directed vertex connectivity augmentation and generalizations". En: *ACM Trans. Algorithms* 4.2 (mayo de 2008), págs. 1-21. doi: 10.1145/1361192.1361197.
- [51] Watanabe, T. y Nakamura, A. "Edge-connectivity augmentation problems". En: *J. Comput. System Sci.* 35.1 (ago. de 1987), págs. 96-144. doi: 10.1016/0022-0000(87)90038-9.
- [52] Weisstein, E. W. *Edge Cover*. <https://mathworld.wolfram.com/EdgeCover.html>.
- [53] Williamson, D. P., Goemans, M. X., Mihail, M. y Vazirani, V. V. "A primal-dual approximation algorithm for generalized Steiner network problems". En: *Combinatorica* 15.3 (sep. de 1995), págs. 435-454. doi: 10.1007/bf01299747.

- [54] Williamson, D. P. y Shmoys, D. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.
- [55] Zoo, C. *APX definition*. https://complexityzoo.net/Complexity_Zoo:A#apx.