

UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Matemática

Grafos perfectos.

Una reseña sobre los principales resultados conocidos.

Tesis de Licenciatura

Alumno: Luciano N. Grippo

Director: Dr. Guillermo A. Durán.

Marzo-2006

Dedicado a la memoria de mis abuelos, Oscar y Jose.

Resumen

Este trabajo consiste de seis capítulos. En el primero se introduce la notación necesaria y algunas nociones sobre grafos perfectos. En el segundo capítulo abordamos el teorema débil de grafos perfectos, en la primer sección mostramos como Lovász demuestra este teorema utilizando teoría de grafos, mientras que en la segunda sección damos una breve introducción sobre teoría poliedral, la relación que existe entre poliedros y grafos perfectos y como se utiliza para demostrar el teorema débil; esta demostración alternativa mantiene los rasgos generales de la demostración de Fulkerson. El tercer capítulo contiene una aplicación del famoso algoritmo del elipsoide, algoritmo de suma importancia teórica, que fue aprovechada por Grötschel, Lovász y Schrijver [26] para demostrar que el problema de coloreo se puede resolver en tiempo polinomial, para los grafos perfectos. En el cuarto capítulo veremos los pasos que se siguieron para demostrar la conjetura fuerte sobre grafos perfectos [14] y cuales fueron las ideas utilizadas, sin ahondar en los detalles técnicos de la demostración. En el quinto capítulo estudiaremos el algoritmo de reconocimiento de grafos perfectos [12], si bien la demostración de la conjetura fuerte fue un resultado de gran importancia, este algoritmo polinomial no utiliza de manera directa el teorema fuerte en su implementación. Finalmente en el sexto capítulo, veremos dos variantes importantes de los grafos perfectos: los grafos clique perfectos, K -perfectos y una serie de resultados relevantes obtenidos en estos últimos años.

Agradecimientos

A mi director Guillermo Durán (Willy), por su buena predisposición y aceptarme como alumno a pesar de la distancia.

A mis padres Norma y Norberto, por haberme apoyado incondicionalmente en el transcurso de toda mi carrera.

A Laura, mi gran amor, mi actual compañera y futura esposa, por darme el empuje y todo el amor necesario para seguir adelante.

A Ezequiel y Noelia.

A Liliana y Raul, por hacerme sentir, desde el primer momento, parte de la familia.

A mi hermana Betina y mi cuñado Damián.

A mis dos sobrinos Ignacio y Camila con quienes volví a retomar los juegos de la mi infancia.

A mis adorables Abuelas Josefina y Elia.

A mis amigos de la vida: Javier, Alejandro, Germán, Mauro, Tomás, Pablo y especialmente a Andrés, por **aguantarme** cada vez que tenía algún problema con la computadora.

A mis amigos y compañeros de facultad: Santiago (el hincha de Ford), Leonardo (mi actual vecino), Ezquiel R., Ezequiel D., Nino, Mariano, Daniel, Agustín, el Colo,

Adrián, Hugo, Nadia, Patricia, Sandra y especialmente a Andrés por brindarme su ayuda cuando la necesitaba y a todos aquellos con los que compartí mi paso por exactas.

A mis compañeros y amigos de trabajo actuales y anteriores, Damian, Selva, Juan Pablo, Mariel, Patricia, Alejandro, Romina, Reggina, Jorge, Agustín, Gabriel y especialmente a Laura y Evangelina con quienes aprendí tanto sobre docencia.

Índice general

Resumen	3
Agradecimientos	4
1. Definiciones y conceptos previos	8
1.1. Notación y grafos especiales	8
1.2. Grafos perfectos	10
2. Teorema débil.	13
2.1. Primera Demostración.	14
2.2. Segunda demostración.	19
2.2.1. Definiciones y resultados sobre poliedros.	19
2.2.2. Grafos Pluperfectos.	20
2.2.3. Resultados de programación lineal (PL)	22
2.2.4. Equivalencia entre grafos perfectos y pluperfectos	23
2.2.5. Pares Blocking Y Anti - Blocking. Definiciones y resultados . .	26
2.2.6. Teorema débil	28
3. Polinomialidad de algunos problemas algorítmicos clásicos en grafos perfectos.	29
3.1. El método del elipsoide aplicado a PL.	29
3.1.1. Nociones Geométricas	30
3.1.2. Descripción de la versión básica del algoritmo del elipsoide . .	32
3.1.3. Versión básica	34

3.1.4.	Complejidad y detalles de implementación.	35
3.2.	El método del elipsoide aplicado a problemas de optimización.	36
3.3.	Equivalencia entre los problemas de optimización y separación.	40
3.3.1.	Problemas de optimización y separación	40
3.3.2.	Conjuntos independientes en grafos perfectos.	42
4.	Teorema fuerte.	46
4.1.	Grafos Perfectos sin C_4	47
4.2.	Estrategia de la demostración en el caso general.	49
4.2.1.	Definiciones previas y demostración del Teorema Fuerte asumiendo el teorema de descomposición.	49
4.2.2.	¿Cómo se puede demostrar el teorema 4.20?	52
4.3.	Apariciones de K_4 en un grafo Berge	57
4.4.	Los siguientes pasos	59
5.	Algoritmo de reconocimiento.	64
5.1.	Rutina 1	66
5.1.1.	Reconocimiento de Pirámides.	66
5.1.2.	Reconocimiento de Joyas.	69
5.1.3.	Detector de shortest <i>odd hole clean</i>	70
5.1.4.	El uso de near-cleaners y final de la rutina 1.	72
5.2.	Rutina 2.	75
5.3.	Algoritmo Cleaning.	78
5.4.	Algoritmo de Reconocimiento de grafos Berge.	79
6.	Variantes de grafos perfectos.	81
6.1.	Grafos K-perfectos	81
6.2.	Grafos Clique Perfectos	82
	Bibliografía	85

Capítulo 1

Definiciones y conceptos previos

1.1. Notación y grafos especiales

Definición 1.1 *Un grafo $G = (V(G), E(G))$ está compuesto por dos conjuntos, $V(G)$ es un conjunto finito que llamaremos vértices del grafo. $E(G) \subseteq V(G) \times V(G)$ (con $u \neq v$), consideramos $e = (u, v) = (v, u)$ para $u, v \in V(G)$ y las llamamos aristas del grafo y a u y v lo llamamos extremos de la arista e . En ocasiones, cuando quede claro, usaremos la notación abreviada $G = (V, E)$ o simplemente G .*

Diremos que dos vértices son adyacentes si y sólo si $(u, v) \in E(G)$. Se representará, gráficamente, cada vértice como un punto y si dos vértices son adyacentes estarán unidos por un segmento.

El cardinal del conjunto de vértices y aristas será notado respectivamente como $|V(G)|$ y $|E(G)|$.

Definición 1.2 *Llamaremos el complemento de un grafo $G = (V, E)$ y lo notamos $\overline{G} = (\overline{V}, \overline{E})$ al grafo que se obtiene a partir de G de la siguiente forma: $\overline{V} = V$ (G y \overline{G} tienen los mismos vértices) y $\overline{E}(\overline{G}) = \{(u, v) \in V(\overline{G}) \times V(\overline{G}) | u \neq v, (u, v) \notin E(G)\}$.*

Observación 1.3 *Dos vértices del grafo G u y v son no adyacentes si y sólo si son adyacentes en \overline{G} .*

A continuación introduciremos algunos grafos que serán mencionados en los próximos capítulos.

Definición 1.4 *Diremos que un grafo es completo si y sólo si todos sus vértices son adyacentes entre sí. Al grafo completo con n vértices lo notamos por K_n .*

Definición 1.5 Diremos que un grafo es bipartito si existe dos subconjuntos disjuntos $A, B \subseteq V(G)$ tales que $V = A \cup B$ y dos vértices u y v son adyacentes sólo si $u \in A$ y $v \in B$ o viceversa. Al grafo bipartito con $|A| = m$ y $|B| = n$ tal que todos los vértices de A son adyacentes a todos los vértices de B lo llamamos grafo bipartito completo, y lo notamos por $K_{m,n}$.



Figura 1.1: Grafo completo de cinco vértices y grafo bipartito completo con $|A| = 3$ y $|B| = 3$.

Definición 1.6 Sea $G = (V, E)$ un grafo. Diremos que $H = (X, F)$ es un subgrafo de G si y sólo si $X \subseteq V$ y $F \subseteq E$, si se cumple que $F = \{(u, v) \in E \mid u, v \in X\}$ diremos que H es un subgrafo inducido de G y lo notamos por $G[X]$.

Definición 1.7 Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos, diremos que son isomorfos si existe una función biyectiva $f : V \rightarrow V'$ tal que $(u, v) \in E$ si y sólo si $(f(u), f(v)) \in E'$.

Definición 1.8 Dado un grafo G y un vértice v se define:

$N(v) = \{u \in V \mid (u, v) \in E\}$: conjunto de vecinos del vértice v .

Clique: es un subgrafo completo de G maximal (respecto de la inclusión).

Conjunto independiente: es un conjunto de vértices tal que no son adyacentes entre sí.

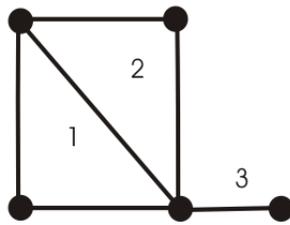
n -ciclo: es un conjunto de vértices $\{v_1, \dots, v_n\}$ tal que v_i es adyacente a v_{i+1} para todo $1 \leq i \leq n - 1$ y v_1 es adyacente a v_n . Lo notamos por C_n y n es la longitud del ciclo.

Camino: es un conjunto de vértices $\{v_1, \dots, v_n\}$ tal que v_i es adyacente a v_{i+1} para todo $1 \leq i \leq n - 1$. Lo notamos por $P_n = v_1 - v_2 - \dots - v_n$. A v_1 y v_n los llamamos extremos o decimos que P es un camino entre v_1 y v_n .

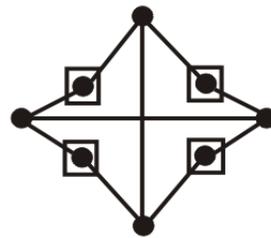
$G \setminus \{v\}$: es el grafo que se obtiene, a partir de G , eliminando v y todas su aristas que lo tienen como extremo.

Hole (o agujero): es un subgrafo inducido de G isomorfo a un ciclo para algún $n \geq 4$. Si la longitud de este ciclo es impar lo llamaremos odd hole (o agujero impar) y en caso contrario even hole (o agujero par).

Anti-hole (o anti-agujero): es un subgrafo inducido en G cuyo complemento es un hole en \overline{G} . La paridad se define igual que antes.



Grafo con 3 cliques



Conjunto independiente

Figura 1.2: a la izquierda estan numeradas las cliques del grafo y a la derecha los vertices recuadrados representan un conjunto independiente del grafo.

Definición 1.9 Diremos que un grafo G es conexo si dados dos vértices cualesquiera u y v , existe un camino entre u y v . Si $X \subseteq V$ y $G[X]$ es conexo (subgrafo inducido por X) diremos que X es conexo. A los conjuntos conexos maximales (por la inclusión) los llamamos componentes conexas de G .

1.2. Grafos perfectos

Antes de definir a los grafos perfectos debemos introducir algunos parámetros importantes:

Definición 1.10 Sea $G = (V, E)$ un grafo, definimos:

$\omega(G)$: es el tamaño de una clique de cardinal máximo.

$\chi(G)$: es la mínima cantidad de colores necesario para colorear los vértices, de manera tal que no haya dos vértices adyacentes con el mismo color.

$\alpha(G)$: es el tamaño de un conjunto independiente de cardinal máximo.

$\kappa(G)$: es la mínima cantidad de cliques necesaria para cubrir los vértices de G .

Observación 1.11 *Es fácil ver que $\omega(G) \leq \chi(G)$, $\alpha(G) \leq \kappa(G)$, $\omega(G) = \alpha(\overline{G})$ y $\chi(G) = \kappa(\overline{G})$.*

En 1961 Berge define los grafos perfectos [2].

Definición 1.12 *Un grafo es perfecto si y sólo si $\omega(G[A]) = \chi(G[A])$, $\forall A \subseteq V$ (también llamados χ -perfectos). Diremos que es α -perfecto si $\alpha(G[A]) = \kappa(G[A])$, $\forall A \subseteq V$*

Lovász en 1972 demuestra la equivalencia entre estos grafos.

Teorema 1.13 [31] *Un grafo G es χ -perfecto $\Leftrightarrow G$ es α -perfecto.*

Berge probó que hay muchas clases de grafos que son perfectos: los grafos cordales o triangulados, los bipartitos, los de comparabilidad y todos sus complementos ([2],[1]).

Observación 1.14 *Este teorema es equivalente a decir que G es perfecto si y sólo si \overline{G} es perfecto (ver la observación 1.11).*

Mas tarde Fulkerson demostró este teorema usando teoría poliedral ([20], [22], [23]). Pero pasaron treinta años más hasta que se demostró la conjetura fuerte [14], que caracteriza los grafos perfectos a partir de subgrafos prohibidos.

Teorema 1.15 *Un grafo G es perfecto si y solo si G no contiene a C_{2r+1} y $\overline{C_{2r+1}}$ con $r \geq 2$ como subgrafos inducidos, es decir no contiene odd holes ni odd anti-holes (a estos últimos los llamaremos grafos Berge).*

Observación 1.16 *Claramente los grafos C_{2r+1} no son perfectos, dado que $\chi(C_{2r+1}) = 3$ y $\omega(C_{2r+1}) = 2$ y por lo tanto $\overline{C_{2r+1}}$ tampoco es perfecto (ver el teorema 1.13).*

Observación 1.17 *Si bien C_{2r+1} y $\overline{C_{2r+1}}$ no son perfectos, todos sus subgrafos inducidos sí lo son.*

Definición 1.18 *Diremos que un grafo G es minimalmente imperfecto si G no es perfecto, pero sí lo son todos sus subgrafos inducidos.*

Por lo tanto el teorema 1.15 se puede reformular de la siguiente manera.

Teorema 1.19 *Los únicos grafos minimalmente imperfectos son C_{2r+1} y $\overline{C_{2r+1}}$ con $r \geq 2$.*

Este teorema, antes de ser demostrado en general (como se verá más adelante) fue demostrado en muchos casos particulares. En 1992 Prömel y Steger [35] demostraron que el teorema se cumplía asintóticamente: sea $P(n)$ y $B(n)$ el conjunto de grafos con n vértices perfectos y *Berge* respectivamente, entonces

$$\lim_{n \rightarrow \infty} \frac{|P_n|}{|B_n|} = 1$$

Capítulo 2

Teorema débil.

En este capítulo daremos dos demostraciones del teorema débil de los grafos perfectos [31]. La primera usa sólo teoría de grafos, al igual que la demostración de Lovász. La segunda usa resultados sobre poliedros, manteniendo los rasgos de la demostración de Fulkerson ([20], [22], [23]); que ese mismo año, unos meses después de Lovász, demostró este teorema basándose en la teoría de pares anti-blocking.

En la primera demostración se usa la definición clásica de grafo perfecto [2]. En la segunda se usa una definición equivalente que vincula los grafos perfectos con la familia de poliedros $\{x \in \mathbb{R}_+^n \mid Kx \leq 1\}$, donde K es la matriz de cliques, también llamada matriz de incidencia clique - vértice del grafo.

Para la segunda demostración se necesitan una serie de resultados relativos a poliedros y programación lineal, que serán enunciados sin demostración en esta sección.

Comencemos enunciando el teorema que se demostrará.

Teorema 2.1 [31] *Dado $G=(V,E)$ un grafo, las siguientes propiedades son equivalentes:*

1. $\omega(G[A]) = \chi(G[A])$, $\forall A \subseteq V$.
2. $\alpha(G[A]) = \kappa(G[A])$, $\forall A \subseteq V$.
3. $\omega(G[A]) \cdot \alpha(G[A]) \geq |A|$, $\forall A \subseteq V$.

Observación 2.2 *Observar que probar la equivalencia entre 1. y 2. es equivalente a probar que G es perfecto si y sólo si \overline{G} perfecto.*

2.1. Primera Demostración.

Definición 2.3 Sea $G = (V, E)$ un grafo, $V = \{v_1, \dots, v_n\}$ y $h \in \mathbb{Z}_{\geq 0}^{|V|}$, definimos el grafo $G \circ h$, como el grafo obtenido a partir de G reemplazando por cada $h_i > 0$, v_i por un conjunto independiente H_i de h_i vértices. Cada vértice de H_i es adyacente a todos los vértices de $N(v_i)$. Si $h_i = 0$ eliminamos v_i y todas las aristas que tienen a v_i como extremo.

A los grafos así obtenidos, decimos que se obtiene a partir de $G = (V, E)$ por multiplicación de vértices. Necesitaremos algunos lemas referentes a esta operación y su relación con la perfección de un grafo, para así poder demostrar el teorema principal de este capítulo.



Figura 2.1: A la izquierda un grafo G y a la derecha vemos el mismo grafo multiplicado por el vértice $2'$

Lema 2.4 Si $G = (V, E)$ cumplen (1) y (2) del teorema 2.1 y $h \in \mathbb{Z}_{\geq 0}^{|V|}$ entonces $H = G \circ h$ cumple (1) y (2).

Demostración Si G está formado por un único vértice, el lema se cumple trivialmente. Supongamos que se cumple para todo grafo con menos vértices que $G = (V, E)$. Si algún $h_i = 0$, entonces $H = (G \setminus v_i) \circ \hat{h}^i$ (donde \hat{h}^i se obtiene de h eliminando la coordenada i -ésima). Si $G = (V, E)$ satisface (1) y (2) entonces $G \setminus v_i$ satisface (1) y (2) luego por hipótesis inductiva H satisface (1) y (2).

Por lo tanto podemos suponer que $h_i \geq 1 \forall i = 1, \dots, |V|$, como H se obtiene de G mediante una sucesión de operaciones del tipo $H = G \circ v_i$, donde ésto quiere decir agregar una copia de v_i a G y aristas que tiene por extremos a cada vecino de v_i y el nuevo vértice ; alcanza con probar el lema para un grafo H de este tipo.

Supongamos sin pérdida de generalidad que $i = 1$. H se obtiene de G agregando un vértice v'_1 no adyacente a v_1 y adyacente a todos los vecinos de v_1 . Por lo tanto $\omega(H) = \omega(G)$. Dado que G es $\omega(G)$ - coloreable, H lo puedo colorear pintando a v'_1 del mismo color que v_1 , luego $\chi(H) = \omega(G) = \omega(H)$ como queríamos probar.

Ahora asumamos que G satisface (2) y probemos que $\alpha(H) = \kappa(H)$. Sea \mathcal{K} un cubrimiento por cliques de G con $|\mathcal{K}| = \kappa(G) = \alpha(G)$, y sea K_{v_1} la clique de G que contiene a v_1 . Hay dos casos posibles:

Caso 1: v_1 está contenido en un conjunto independiente máximo S ($|S| = \alpha(G)$).

En este caso $S' = S \cup \{v'_1\}$ es un conjunto independiente de H .

$$\alpha(H) = \alpha(G) + 1$$

Luego $K \cup \{v'_1\}$ es un cubrimiento por cliques de H , por lo tanto:

$$\kappa(H) \leq \kappa(G) + 1 = \alpha(G) + 1 = \alpha(H) \leq \kappa(H)$$

Por lo tanto $\kappa(H) = \alpha(H)$.

Caso 2: Ningún conjunto independiente máximo de G contiene a v_1 .

En este caso:

$$\alpha(H) = \alpha(G)$$

Dado que cada clique de \mathcal{K} interseca cada conjunto independiente máximo exactamente una vez, en particular es cierto para K_{v_1} . Pero v_1 no pertenece a ningún conjunto independiente máximo. Por lo tanto:

$$\kappa(G[V \setminus D]) = \alpha(G[v \setminus D]) = \alpha(G) - 1 = \alpha(H) - 1$$

Tomando un cubrimiento por cliques de $G[V \setminus D]$ de cardinal $\alpha(H) - 1$ junto con el clique $D \cup \{v_1\}$, obtenemos un cubrimiento de H . Por lo tanto:

$$\kappa(H) = \alpha(H)$$

□

Lema 2.5 Sea $G = (V, E)$ un grafo tal que todos sus subgrafos inducidos propios cumplen (2) del teorema 2.1. Y sea H un grafo obtenido de G por multiplicación de vértices. Si G satisface (3) del teorema 2.1, entonces H satisface (3) del teorema 2.1.

Demostración Sea G un grafo que cumple (3) y sea H el mínimo grafo obtenido a partir de G por multiplicación de vértices que no cumple (3). Por lo tanto

$$\omega(H) \cdot \alpha(H) < |V(H)|$$

y (3) se cumple para todo subgrafo inducido de H .

Como en el lema anterior, podemos asumir que cada uno de los vértices fue multiplicado por al menos un vértice y que algún $h_i \geq 2$. Supongamos, sin pérdida de generalidad, $i = 1$ y sea $S = \{\hat{v}_1, \dots, \hat{v}_h\}$ el conjunto independiente correspondiente a v_1 .

Por la minimalidad de H , (3) se cumple para $H[V(H) \setminus \{v_1\}]$ con lo cual:

$$|V(H)| - 1 = |V(H) \setminus \hat{v}_1| \leq \omega(H[V(H) \setminus \hat{v}_1]) \cdot \alpha(H[V(H) \setminus \hat{v}_1]) \leq \omega(H) \cdot \alpha(H) \leq |V(H)| - 1$$

Por lo tanto:

$$\omega(H) \cdot \alpha(H) = |V(H)| - 1 = \omega(H[V(H) - \hat{v}_1]) \cdot \alpha(H[V(H) - \hat{v}_1])$$

Definamos:

$$p = \omega(H) = \omega(H[V(H) - \hat{v}_1])$$

$$q = \alpha(H) = \alpha(H[V(H) - \hat{v}_1])$$

consecuentemente

$$pq = |V(H)| - 1$$

Dado que $X = H[V(H) \setminus S]$ se obtiene de $G \setminus \{v_1\}$ por multiplicación de vértices, el lema 2.4 implica que X satisface (2). Por lo tanto X puede ser cubierto por q cliques K_1, \dots, K_q ($|K_1| \geq \dots \geq |K_q|$). Obviamente,

$$\sum_{i=1}^q |K_i| = |V(H) - S| = |V(H)| - h = pq - (h - 1)$$

dado que $|K_i| \leq p$, a lo sumo $h - 1$ de los K_i contribuyen p a la suma. Por lo tanto,

$$|K_1| = \dots = |K_{q-h+1}| = p$$

Sea $\hat{H} = H[K_1 \cup \dots \cup K_{q-h+1} \cup \{\hat{v}_1\}]$. Entonces

$$|V(\hat{H})| = p(q - h + 1) + 1 < pq + 1 = |V(H)|.$$

Por minimalidad de H , $\omega(\hat{H}) \cdot \alpha(\hat{H}) \geq |V(\hat{H})|$. Como $p = \omega(H) \geq \omega(\hat{H})$, consecuentemente

$$\alpha(\hat{H}) \geq \frac{|V(\hat{H})|}{p} \geq (q - h + 1) + 1 > q - h + 1$$

Sea \hat{S} un conjunto independiente de cardinal $q - h + 2$, luego $\hat{v}_1 \in \hat{S}$. Por otro lado \hat{S} debería contener dos vértices de una misma clique.

Por lo tanto $T = \hat{S} \cup S$ es un conjunto independiente de H con $q + 1$ vértices, contradiciendo la definición de q . \square

Demostración del teorema 2.1 Asumamos que el teorema es válido para todo grafo con menos vértices que G .

(1) \Rightarrow (3) Supongamos que podemos colorear $G[A]$ con $\omega(G[A])$ colores, dado que hay a lo sumo $\alpha(G[A])$ vértices de un mismo color se sigue que $\omega(G[A]) \cdot \alpha(G[A]) \geq |A|$.

(3) \Rightarrow (1) Si $G = (V, E)$ satisface (3), entonces, por hipótesis inductiva, cada uno de los subgrafos inducidos propios de G satisfacen (1) y (3). Luego alcanza con probar que $\alpha(G) = \chi(G)$.

Si existiera un conjunto independiente S de vértices de G tal que $\omega(G[V - S]) < \omega(G)$, podríamos pintar S de naranja y $V - S$ con $\omega(G) - 1$ colores (sin usar el naranja), por lo tanto $\omega(G) = \chi(G)$.

Supongamos que $G[V - S]$ contiene un $\omega(G)$ - clique $K(S)$, para todo conjunto independiente S de G . Sea \mathcal{L} la colección de todos los conjuntos independientes de G , observar que $S \cap K(S) = \emptyset$.

Para cada $x_i \in V$, h_i denota el número de cliques $K(S)$ que contienen a x_i . Sea $H = (X, F)$ el grafo que se obtiene multiplicando cada x_i por h_i . Consecuetemente por el lema 2.5,

$$\omega(H) \cdot \alpha(H) \geq |X|.$$

Por otro lado, usando simples argumentos combinatorios, se puede mostrar fácilmente que

$$\begin{aligned} |X| &= \sum_{x_i \in V} h_i \\ &= \sum_{S \in \mathcal{L}} |K(S)| = \omega(G) \cdot |\mathcal{L}| \end{aligned}$$

Efectivamente, si se considera la matriz de incidencia cuyas filas están indexadas por los vértices x_1, \dots, x_n y cuyas columnas se corresponden a las cliques $K(S)$ para $S \in \mathcal{L}$. Entonces, h_i es igual al número de coordenadas no nulas en la fila i , $|K(S)|$ es igual al número de coordenadas no nulas en su correspondiente columna, la cual por definición es igual a $\omega(G)$.

$$\omega(H) \leq \omega(G)$$

Dado que a lo sumo una copia de cualquier vértice de G puede estar en una clique de H

$$\alpha(H) = \max_{T \in \mathcal{L}} \sum_{x_i \in T} h_i.$$

Dado que si un conjunto independiente máximo contiene una copia de x_i , entonces debe contener a todas las respectivas copias, si restringimos nuestra atención a las filas de la matriz que corresponden a los vértices que pertenecen a T , obtenemos

$$\begin{aligned} \alpha(H) &= \max_{T \in \mathcal{L}} \left[\sum_{S \in \mathcal{L}} |T \cap K(S)| \right] \\ &\leq |\mathcal{L}| - 1 \end{aligned}$$

Y por lo tanto

$$\omega(H) \cdot \alpha(H) \leq \omega(G) \cdot (|\mathcal{L}| - 1) < |X|,$$

llegando así a un contradicción.

(2) \Leftrightarrow (3). Esto se satisface trivialmente, por las siguientes equivalencias:

G satisface (2) $\Leftrightarrow \overline{G}$ satisface (1) $\Leftrightarrow \overline{G}$ satisface (3) $\Leftrightarrow G$ satisface (3) □

2.2. Segunda demostración.

Esta demostración utiliza definiciones y resultados sobre poliedros, que enunciaremos sin demostración.

2.2.1. Definiciones y resultados sobre poliedros.

Definición 2.6 *Un poliedro $P \subseteq \mathbb{R}^n$ es un conjunto de puntos que satisfacen un número finito de inecuaciones lineales; es decir $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, donde $(A|b)$ es una matriz de $m \times (n + 1)$.*

Definición 2.7 *Un poliedro $P \subseteq \mathbb{R}^n$ acotado, se lo llama polígono.*

Definición 2.8 *Dado un conjunto $S \subseteq \mathbb{R}^n$, diremos que $x \in \mathbb{R}^n$ es una combinación convexa de S , si existen un número finito de puntos $\{x_i\}_{i=1}^t$ de S y $\lambda \in \mathbb{R}^n$ con $\sum_{i=1}^n \lambda_i = 1$ tales que $x = \sum_{i=1}^n \lambda_i x_i$. Al conjunto formado por los puntos de \mathbb{R}^n , que son combinación convexa de S , lo llamaremos cápsula convexa de S , y lo notamos $\text{conv}(S)$.*

Definición 2.9 *Sea $P \subseteq \mathbb{R}^n$ un poliedro, diremos que $x \in P$ es un vértice de P si no existen $x_1, x_2 \in P$ ($x_1 \neq x_2$) tales que $x = \frac{1}{2}x_1 + \frac{1}{2}x_2$.*

Definición 2.10 *Diremos que un poliedro $P \subseteq \mathbb{R}_+^n$ es entero si todos sus vértices tienen coordenadas enteras.*

$$K_G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$



Figura 2.2: La matriz de clique y su grafo derivado.

2.2.2. Grafos Pluperfectos.

Definición 2.11 Una matriz de cliques K_G de un grafo G , es una matriz de ceros y unos, cuyas filas representan todas las cliques de G y las columnas representan los vértices de G . El elemento (i, j) de la matriz es 1 si el vértice j pertenece a la clique i , y 0 en caso contrario.

Definición 2.12 Llamaremos polígono asociado al grafo G a $P = \{x \in \mathbb{R}^n | K_G x \leq 1 \text{ y } x \geq 0\}$.

Donde $1 = (1, \dots, 1)$ y $0 = (0, \dots, 0)$.

Definición 2.13 Un grafo G es pluperfecto si y sólo si su polígono asociado es entero.

Proposición 2.14 Sea $P_G = \{x \in \mathbb{R}^n | Ax \leq 1\} \subseteq \mathbb{R}^n$ un poliedro, entonces $P = \text{conv}\{b_1, \dots, b_r\}$ donde las b_i 's son los vértices de P .

Definición 2.15 Diremos que $x \in \mathbb{R}^n$ ($n = |V(G)|$) es el vector característico de un conjunto independiente S de G si: $x_v = 1 \forall v \in S$ y 0 en caso contrario.

Lema 2.16 Sea P_G el polígono asociado al grafo pluperfecto G ; x es un vértice de P si y sólo si x es un vector asociado a un conjunto independiente de G .

Demostración (\Rightarrow) Si x es un vértice de P , por definición sus coordenadas deben ser enteras. Dado que K_G es una matriz clique de G y $x \geq 0$, se deduce que x debe ser un vector de ceros y unos. Por lo tanto representa un subconjunto S de vértices de G , luego alcanza con probar que estos vértices no son adyacentes entre si. Supongamos que existieran $u, v \in S$ adyacentes en G , por lo tanto existe un fila K_i de K_G que tiene un 1 en la columna u y en la v . Luego $K_i x \geq 2$, contradiciendo el hecho que $x \in P$.

(\Leftarrow) Sea x es el vector característico de algún conjunto independiente de G , claramente $x \in P$. Luego, por la proposición 2.14, se ve que x pertenece a la cápsula convexa formada por algunos vértices de P (llamémoslos, $\{b_1, \dots, b_s\}$), es decir

$$x^k = \sum_{i=1}^s c_i b_i^k \quad \text{donde} \quad 1 = \sum_{i=1}^s c_i \quad \text{y} \quad 0 \leq c_i \leq 1.$$

Por lo tanto si $x^k = 1$ entonces $b_i^k = 1 \forall i$, si $x^k = 0$ entonces $b_i^k = 0 \forall i$. Por lo tanto $x = b_i$ y resulta ser un vértice de P , como queríamos probar. \square

En otras palabras, existe una biyección entre los conjuntos independientes de un grafo pluperfecto G y los vértices de su polítopo asociado.

Proposición 2.17 *Todo subgrafo inducido de un grafo pluperfecto G es pluperfecto.*

Demostración El poliedro inducido por un subgrafo inducido de G , por ejemplo $G[V \setminus U]$ donde $U \subseteq V$ ($|U| = k$), resulta ser $\hat{P} = \{x \in \mathbb{R}^{n-k} \mid \hat{K} \leq 1 \text{ y } x \geq 0\}$; donde \hat{K} se obtiene de K_G (matriz de cliques de G) eliminando las k columnas correspondientes a los vértices de U . Se ve fácilmente que todo vértice de \hat{P} es una proyección de un vértice de P y por lo tanto sus coordenadas son enteras. \square

Por el principio de dualidad (ver la proposición 2.19) y debido a la relajación sobre las restricciones tenemos la desigualdad:

$$\alpha(G) \leq \max\{1x \mid K_G x \leq 1, x \in \mathbb{R}_+^n\} = \min\{1y \mid y K_G \geq 1, y \in \mathbb{R}_+^m\} \leq \kappa(G)$$

Para el caso de los grafos perfectos vale la igualdad. En cambio, si por ejemplo, G fuera un ciclo con $2k + 1$ vértices resulta $\alpha(G) = k$, $\kappa(G) = k + 1$ y el problema de programación lineal relajado tiene como resultado $k + \frac{1}{2}$.

Se puede generalizar $\alpha(G)$ y $\kappa(G)$ al caso en que los vértices de G tengan un peso, de la siguiente manera:

$$\alpha(G, c) = \text{máx}\{cx \mid K_G x \leq 1, x \in \{0, 1\}^n\}$$

$$z(G, c) = \text{máx}\{cx \mid Kx \leq 1, x \in \mathbb{R}_+^n\}$$

$$= \text{mín}\{1y \mid yK \geq c, y \in \mathbb{R}_+^n\}$$

$$\kappa(G, c) = \text{mín}\{1y \mid yK \geq c, y \in \mathbb{Z}_+^n\}$$

Por lo tanto tenemos $\alpha(G, 1) = \alpha(G)$, $\kappa(G, 1) = \kappa(G)$, y para todo $c \in \{0, 1\}^n$ con $S = \{j \in V(G) \mid c_j = 1\}$ se tiene $\alpha(G, c) = \alpha(H)$ y $\kappa(G, c) = \kappa(H)$, donde H resulta ser el subgrafo de G inducido por H . Por dualidad y relajación:

$$\alpha(G, c) \leq z(G, c) \leq \kappa(G, c)$$

Para grafos pluperfectos la primera desigualdad resulta ser una igualdad, ya que el máximo, si es finito, siempre se alcanza en uno de sus extremos; y estos resultan ser los vectores asociados a los conjuntos independientes de G . A continuación probaremos un resultado que nos da la equivalencia entre grafos pluperfectos y perfectos. Pero antes necesitamos enunciar algunos resultados sobre programación lineal.

2.2.3. Resultados de programación lineal (PL)

Llamaremos primal (P) al siguiente problema de programación lineal:

$$\text{máx } cx$$

$$Ax \leq b$$

$$x \geq 0$$

Y llamaremos el dual del problema anterior (D) a:

$$\text{mín } by$$

$$yA \geq c$$

$$y \geq 0$$

Donde $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$.

Proposición 2.18 (*Holgura complementaria*) Si x^* es una solución óptima de (P) e y^* es una solución óptima de (D). Entonces $x_j^*(yA - c)_j = 0 \quad \forall 1 \leq j \leq n$ y $y_i^*(Ax - b)_i = 0 \quad \forall 1 \leq i \leq m$.

Proposición 2.19 (*Principio de dualidad*) Si (P) ó (D) tienen una solución óptima finita, entonces ambos tienen solución óptima finita (x^* de (P) e y^* de (D)) tales que $cx^* = by^*$.

Consideremos el siguiente problema de programación lineal, sobre un poliedro P .

$$z_{PL} = \text{máx}\{cx \mid x \in P\}$$

Proposición 2.20 *Los siguientes enunciados son equivalentes:*

- (1) P es entero
- (2) PL tiene una solución óptima entera para todo $c \in \mathbb{R}^n$ para los que existe una solución óptima.
- (3) PL tiene una solución óptima entera para todo $c \in \mathbb{Z}^n$ para los que existe una solución óptima.
- (4) z_{LP} es entero para todo $c \in \mathbb{Z}^n$ para el que LP tiene una solución óptima.

2.2.4. Equivalencia entre grafos perfectos y pluperfectos

Lema 2.21 [17] *Las siguientes propiedades sobre un grafo son equivalentes:*

- (1) $P = \{x \in \mathbb{R}_+^n \mid K_G x \leq 1\}$ es entero. (G es pluperfecto).
- (2) $\alpha(H) = \kappa(H)$ para todo H subgrafo inducido de G .

Demostración (1) \Rightarrow (2) Dado que G es pluperfecto por la proposición 2.17 sabemos que todo subgrafo H de G es pluperfecto. Por lo tanto si H es un subgrafo inducido por $U \subseteq V$ y $c_j = 1$ para $j \in U$ y $c_j = 0$ en caso contrario, se tiene:

$$\alpha(H) = \text{máx}\{cx \mid K_G x \leq 1, x \geq 0\} \quad PLMax$$

Por la proposición 2.19 y dado que G es pluperfecto, alcanza con probar:

$$\text{mín}\{y \mid yK \geq c, y \geq 0\} \quad (PLMin)$$

tiene solución entera para todo $c \in \{0, 1\}^n$.

Demostraremos esto por inducción en el número de coordenadas positivas de c , que resulta ser $|U|$. Notar que si $c = 0$ entonces $y = 0$ resulta ser una solución óptima de (PLMin). Asumamos que la hipótesis es cierta para todo subgrafo propio de G y probemos que vale para G , es decir $\text{mín}\{1y \mid yK_G \geq 1, y \geq 0\}$ tiene una solución óptima entera.

Sea k^i , $i = 1, \dots, m$ las filas de K . Dado que $y = 0$ no es una solución factible de (PLMin), hay un i , por ejemplo $i = r$ tal que $y_r > 0$ (donde y es una solución óptima de (PLMin)). Por la proposición 2.18 resulta que $y_r(1 - K_G x)_r = 0$, para toda solución óptima x . Por lo tanto $k^r x = 1$, para toda solución óptima x . Esto quiere decir que todo conjunto independiente maximal tiene un vértice en común con la clique $C = \{j \in V \mid k_{rj} = 1\}$, debido a que G es un grafo pluperfecto.

Si $k^r = 1$ entonces $r = 1$, y una solución óptima de (PLMax) es $y_1 = 1$, y no hay nada que probar. Si $k^r < 1$ por hipótesis inductiva, $\text{mín}\{1y \mid yK \geq 1 - k^r, y \geq 0\}$ tiene una solución entera que llamaremos y^0 .

Probaremos que $y^0 + e_r$ es una solución óptima de (PLMin). Notar que resulta factible $(y^0 + e_r)K_G \geq 1 - k^r + k^r = 1$. Dado el subgrafo $H = G[V \setminus C]$ obtenemos un conjunto independiente máximo de H , borrando en un conjunto independiente máximo de G el vértice que pertenece a la clique C . Por lo tanto $\alpha(H) = \alpha(G) - 1$, y

$$1y^0 + 1 = \kappa(H) + 1 = \alpha(G) \leq \kappa(G) \leq 1(y^0 + e_r) = 1y^0 + 1$$

Luego $y^0 + e_r$ resulta una solución óptima entera de (PLMin).

(2) \Rightarrow (1) Probaremos que $\alpha(G, c) = z(G, c)$ para todo $c \in \mathbb{Z}_+^n$, y por la proposición 2.20 se deduce que G es pluperfecto. Por (2) sabemos que la igualdad anterior se

cumple para todo $c \in \{0, 1\}^n$. Para los $c \in \mathbb{Z} \setminus \mathbb{Z}_+^n$ definimos $\bar{c}_j = c_j$ si $c_j \geq 1$ y $\bar{c}_j = 0$ en otro caso. Dado que si $c_j < 0$ y x es una solución óptima de $\alpha(G, c)$ ó $z(G, c)$ entonces $x_j = 0$, se tiene que $\alpha(G, c) = \alpha(G, \bar{c})$ y $Z(G, c) = z(G, \bar{c})$. Por lo tanto, por (2), se cumple que $\alpha(G, c) = z(G, c)$ para todo $c \in \mathbb{Z}^n$ con $c \leq 1$.

Para los $c \in \mathbb{Z}_+^n$ con algún $c_j \geq 2$ lo probaremos por inducción. Supongamos que vale $\alpha(G, c') = z(G, c')$ para todo $c' < c$. Sea $c' = c - e_j$. Dado que $c'_j \geq 1$ se sigue por 2.18 que existe un r tal que $k_{rj} = 1$ y $k^r x = 1$ para toda solución óptima de $\max\{c'x \mid Kx \leq 1, x \geq 0\}$. Por lo tanto por hipótesis inductiva se tiene que $\alpha(G, \tilde{c}) = z(G, \tilde{c})$ donde $\tilde{c} = c - k^r$.

Probaremos que $\alpha(G, c) = \alpha(G, \tilde{c}) + 1$. dado que $c > \tilde{c}$, tenemos que $\alpha(G, c) \geq \alpha(G, \tilde{c})$. Sea \tilde{y} una solución óptima de:

$$z(G, \tilde{y}) = \min\{1y \mid yK \geq 1, y \geq 0\}$$

Dado que $(\tilde{y} + e_r)K \geq \tilde{c} + k^r = c$, tenemos $z(G, c) \leq z(G, \tilde{c}) + 1$ y junto con $\alpha(G, c) \leq z(G, c)$ y $\alpha(G, \tilde{c}) = z(G, \tilde{c})$ se obtiene:

$$\alpha(G, \tilde{c}) \leq \alpha(G, c) \leq \alpha(G, \tilde{c}) + 1$$

Finalmente $\alpha(G, c) = \alpha(G, \tilde{c})$ ó $\alpha(G, c) = \alpha(G, \tilde{c}) + 1$.

Supongamos $\alpha(G, c) = \alpha(G, \tilde{c})$ y sea \tilde{x} el vector característico de algún conjunto independiente, tal que $\tilde{c}\tilde{x} = \alpha(G, \tilde{c})$. Entonces $k^r \tilde{x} = 0$ y $c'\tilde{x} = \alpha(G, c')$ dado que $\alpha(G, \tilde{c}) \leq \alpha(G, c') \leq \alpha(G, c)$. Pero ésto es una contradicción, porque antes probamos que $k^r x = 1$ para todo x vector característico de una conjunto independiente tal que $c'x = \alpha(G, c')$. Por lo tanto $\alpha(G, c) = \alpha(G, \tilde{c}) + 1$.

Luego

$$\alpha(G, c) \leq z(G, c) \leq z(G, \tilde{c}) + 1 = \alpha(G, \tilde{c}) + 1 = \alpha(G, c)$$

Por lo tanto $z(G, c) = \alpha(G, c)$, queda así demostrado el lema. \square

Observar que al probar (2) \Rightarrow (1), probamos que (2) implica, trivialmente, que $\max\{cx \mid x \in P\}$ tiene una solución óptima entera par todo $c \in \{0, 1\}^n$ (donde $P = \{x \in \mathbb{R}_+^n \mid Kx \leq 1\}$). Luego usando esta implicación se prueba que P es entero. Por lo tanto se cumple el siguiente resultado.

Corolario 2.22 Si $\max\{cx \mid Kx \leq 1, x \geq 0\}$ tiene solución entera para todo $c \in \{0, 1\}^n$ entonces $P = \{x \in \mathbb{R}_+^n \mid Kx \leq 1\}$ es entero.

Daremos una breve introducción sobre los pares de matrices Blocking y Anti-blocking, que nos permitirá comprender la segunda demostración del teorema débil.

2.2.5. Pares Blocking Y Anti - Blocking. Definiciones y resultados

Hay diversos resultados en optimización combinatoria que tratan de problemas de maximización y minimización que aparecen en pares duales. Por ejemplo si consideramos un digrafo con una fuente y un sumidero, el máximo numero de caminos, disjuntos por aristas, entre la fuente y el sumidero es igual al mínimo número de aristas que separan a ambos [19]. Si se intercambian los roles entre los caminos y las aristas de corte se obtiene el siguiente resultado: el máximo número de aristas disjuntas, por vértices, que separan la fuente y el sumidero es igual al mínimo número de aristas de un camino entre ambos [36]. A continuación algunos resultados, donde se relacionan los problemas de optimización y los pares blocking y anti-blocking. Las demostraciones de estos resultados pueden encontrarse en ([21],[23]), un resumen completo sobre estos resultados y su relación con los problemas de optimización duales en [22].

Supongamos que $P = \{x \in \mathbb{R}_+^n \mid Ax \geq 1\}$ donde A no tiene filas nulas. El blocker $B(P)$ de P es el poliedro:

$$B(P) = \{\pi \in \mathbb{R}_+^n \mid \pi x \geq 1 \forall x \in P\}$$

Donde $B \in \mathbb{R}^{r \times n}$ es una matriz cuyas filas son los vértices $\{x^k\}_{k=1}^r$ de P .

Proposición 2.23 Sea $P = \{x \in \mathbb{R}_+^n \mid Ax \geq 1\}$, $A \in \mathbb{R}^{m \times n}$ no negativa y sin filas nulas. Entonces:

- (1) $B(P) = \{\pi \in \mathbb{R}_+^n \mid B\pi \geq 1\}$ y
- (2) $B(B(P)) = P$.

Ahora supongamos que $P = \{x \in \mathbb{R}_+^n \mid Ax \leq 1\}$ donde A no tiene columnas nulas. El antiblocker $A(P)$ de P es el poliedro:

$$A(P) = \{\pi \in \mathbb{R}_+^n \mid \pi x \leq 1 \forall x \in P\}$$

Donde $C \in \mathbb{R}^{s \times n}$ es una matriz cuyas filas son los vértices $\{y^k\}_{k=1}^s$ de P .

Proposición 2.24 Sea $P = \{x \in \mathbb{R}_+^n \mid Ax \leq 1\}$, $A \in \mathbb{R}^{m \times n}$ no negativa y sin columnas nulas. Entonces:

- (1) $A(P) = \{\pi \in \mathbb{R}_+^m \mid C\pi \leq 1\}$ y
- (2) $A(A(P)) = P$.

Definición 2.25 Sean $A \in \mathbb{R}^{m \times n}$ y $C \in \mathbb{R}^{s \times n}$ matrices no negativas, con la propiedad que $\{\pi \in \mathbb{R}_+^m \mid C\pi \leq 1\}$ es el antiblocker de $\{x \in \mathbb{R}_+^n \mid Ax \leq 1\}$. En ese caso el par ordenado A, C es llamado un par anti-blocking.

Equivalentemente se define un par blocking. A continuación daremos un serie de resultados referentes a pares anti-blocking, que tienen su equivalente en pares blocking.

Definición 2.26 Diremos que la igualdad min-max está dada para el par de matrices $A \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{s \times n}$, si para todo $w \in \mathbb{R}_+^n$ se cumple:

$$\min\{1y \mid yA \geq w, y \in \mathbb{R}_+^m\} = \max_{1 \leq j \leq s} c^j w.$$

Definición 2.27 Diremos que la inecuación max-max se cumple para un par de matrices $A \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{s \times n}$ si y solo si para todo $l \in \mathbb{R}_+^n$ y $w \in \mathbb{R}_+^n$ se cumple:

$$\left(\max_{1 \leq i \leq n} a^i l \right) \left(\max_{1 \leq j \leq s} c^j w \right) \geq lw$$

Los siguientes teoremas dan una caracterización de los pares antiblocking, mediante la igualdad min-max y la desigualdad max-max.

Teorema 2.28 La igualdad min-max para el par ordenado A, C de matrices se cumple si y sólo si A y C forman un par antiblocking.

Teorema 2.29 El par ordenado A, C forman un par antiblocking si y sólo si se cumple:

- (i) $a^i c^i \leq 1$ para todas las filas a^i de A y c^i de C , y
- (ii) la desigualdad max-max se cumple para el par ordenado A, C .

Observar que la vuelta de este teorema nos dice que los roles de A y C son intercambiables, finalmente si el par ordenado A, C es anti-blocking también lo es el par ordenado C, A .

Para finalizar este capítulo daremos la segunda demostración del teorema débil.

2.2.6. Teorema débil

Teorema 2.30 *G es perfecto si y sólo si \overline{G} es perfecto.*

Demostración Si G es perfecto entonces, por el lema 2.21, se sigue que $P = \{x \in \mathbb{R}_+^n \mid Kx \leq 1\}$ es un poliedro entero, cuyos vértices son los vectores asociados a los conjuntos independientes máximos, como se vio anteriormente. Como hay una biyección entre los cliques de \overline{G} (los conjuntos independientes de G) y los vértices de P , las filas de $K_{\overline{G}}$ resultan ser los vértices de P . Luego por la proposición 2.24 K_G y $K_{\overline{G}}$ resultan ser un par anti-blocking (observar que las columnas de K_G son no nulas) $A(P) = \{x \in \mathbb{R}_+^n \mid K_{\overline{G}}x \leq 1\} = \overline{P}$ y $A(\overline{P}) = P = \{x \in \mathbb{R}_+^n \mid K_Gx \leq 1, x \geq 0\}$. Luego por el teorema 2.28:

$$\max\{wx \mid K_{\overline{G}}x \leq 1, x \in \mathbb{R}_+^n\} = \min\{1y \mid yK_{\overline{G}} \geq w, y \in \mathbb{R}_+^n\} = \max_j k^j w$$

para todo $w \in \{0, 1\}^n$. Entonces, se ve fácilmente que existe un k_0^j solución óptima entera de $\max\{wx \mid K_{\overline{G}}x \leq 1, x \in \mathbb{R}_+^n\}$. Por lo tanto por el corolario 2.22, \overline{G} resulta pluperfecto y conecuentemente es perfecto. La vuelta se prueba analógamente tomando complemento. \square

Capítulo 3

Polinomialidad de algunos problemas algorítmicos clásicos en grafos perfectos.

3.1. El método del elipsoide aplicado a PL.

Hasta 1979 no se conocía la complejidad del problema de programación lineal (PL): determinar si existe una solución $x \in \mathbb{R}^n$ tal que $Ax \leq 1$. Fue en ese año que L. G. Kachiyan [28], adaptó un algoritmo propuesto por Shor para problemas de optimización no lineales [42], desarrolló el algoritmo del elipsoide y probó que este problema puede ser resuelto en tiempo polinomial. Este algoritmo se convirtió en una herramienta eficiente para probar la polinomialidad de algunos problemas algorítmicos clásicos, aunque no es eficiente desde el punto de vista de la implementación. En cambio, el conocido algoritmo Simplex desarrollado por Dantzig [18] funciona bien en la mayoría de los casos prácticos; pero lamentablemente este no es un algoritmo polinomial. Klee y Minty [29] fabricaron un ejemplo donde este algoritmo tiene un tiempo de ejecución no polinomial. En este capítulo daremos una descripción del algoritmo del elipsoide y veremos como se utilizó para demostrar la polinomialidad de algunos problemas algorítmicos en grafos perfectos [26].

3.1.1. Nociones Geométricas

En este capítulo notaremos el tamaño de entrada de un número racional, un vector o una matriz encerrándolo entre signos de mayor y menor $\langle - \rangle$.

Definición 3.1 *Un conjunto $E \subseteq \mathbb{R}^n$ es un elipsoide si existen $a \in \mathbb{R}^n$ y una matriz $A \in \mathbb{R}^{n \times n}$ simétrica definida positiva tal que:*

$$E = E(A, a) = \{x \in \mathbb{R}^n \mid (x - a)^t A^{-1} (x - a) \leq 1\}.$$

Observar que $E(I, 0)$ es la esfera de radio 1 centrada en el origen $B(0, 1)$, donde $I \in \mathbb{R}^{n \times n}$ es la matriz identidad y 0 es el vector de \mathbb{R}^n con todas sus coordenadas nulas.

Dado que A es una matriz simétrica definida positiva existe una única matriz Q donde $A = QQ^t$ y tal que:

$$E(A, a) = QS(0, 1) + a$$

También existe una conexión entre los autovectores de A y los ejes del elipsoide, los autovalores de A (reales y positivos) y la longitud de los ejes. La longitud del eje mayor es $2\sqrt{\Lambda}$ donde Λ es el máximo autovalor de A , el eje mayor de E tiene la dirección de un autovector asociado a la matriz A . Análogamente la longitud del eje menor es $2\sqrt{\lambda}$, donde λ representa al menor autovalor de A , y el eje menor tiene la dirección de una autovector asociado a λ . Como consecuencia $B(a, \Lambda)$ es la esfera de menor radio que contiene a la elipse E , y $B(a, \lambda)$ es la esfera de mayor radio contenida en la elipse E .

El volúmen de la elipse $E = E(A, a)$ lo notaremos por $vol(E)$, depende solamente del determinante de A y la dimensión del espacio. Más exactamente:

$$vol(E(A, a)) = \sqrt{\det A} \cdot V_n$$

donde V_n es el volúmen de la esfera de radio 1 en \mathbb{R}^n ,

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)}$$

siendo

$$\Gamma(x) := \int_0^{\infty} e^{-t} t^{x-1} dt, \quad x > 0$$

es la función gamma.

Si $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una transformación afín $T(x) = Dx + b$ y $E(a, A)$, $E(b, B)$ son elipsoides entonces se cumple la relación:

$$\frac{\text{vol}(E(A, a))}{\text{vol}(E(B, b))} = \frac{\text{vol}(T(E(A, a)))}{\text{vol}(T(E(B, b)))}$$

siendo $\text{vol}(T(E(A, a))) = \det D \sqrt{\det A} \cdot V_n$.

Es muy fácil ver que $\text{máx}\{cx \mid x \in E(A, a)\} = c^t a + \sqrt{c^t A c}$.

Si definimos:

$$b := \frac{1}{\sqrt{c^t A c}} A c$$

$$z_{\max} := a + b$$

$$z_{\min} := a - b$$

tenemos las igualdades:

$$c^t z_{\max} = \text{máx}\{c^t x \mid x \in E(A, a)\} = c^t a + \sqrt{c^t A c}$$

$$c^t z_{\min} = \text{mín}\{c^t x \mid x \in E(A, a)\} = c^t a - \sqrt{c^t A c}$$

Como veremos más adelante, el método del elipsoide requiere del cálculo de un elipsoide de volumen mínimo que contiene una sección de un elipsoide dada; obtenida por la intersección de esta con un semiespacio, cuya frontera es un hiperplano que pasa por el centro del elipsoide. Sea $E'(A, a, c) = E(A, a) \cap H$ donde $c \in \mathbb{R}^n - \{0\}$ y $H = \{x \in \mathbb{R}^n \mid c^t x \leq c^t a\}$, donde $\partial H = \{x \in \mathbb{R}^n \mid c^t x = c^t a\}$ es el hiperplano frontera de H . El elipsoide de mínimo volumen $E(A', a')$ que contiene medio elipsoide $E'(A, a, c)$ se calcula de la siguiente manera (*):

$$a' := a - \frac{1}{n+1} b$$

$$A' := \frac{n^2}{n^2 - 1} \left(A - \frac{2}{n + 1} bb^t \right)$$

Se puede probar que A' así definida resulta simétrica definida positiva.

3.1.2. Descripción de la versión básica del algoritmo del elipsoide

El algoritmo decide si el poliedro

$$P = \{x \in \mathbb{R}^n \mid c_i^t x \leq \gamma_i, i = 1, \dots, m\} = \{x \in \mathbb{R}^n \mid Cx \leq b\}$$

es vacío o no. En caso de no ser vacío exhibe un $x \in \mathbb{R}^n$ tal que $x \in P$. Para que el algoritmo funcione correctamente se tiene que garantizar que P cumple las siguientes condiciones:

1. P es acotado.
2. Si $P \neq \emptyset$, entonces P es de dimensión n .

La idea del algoritmo es la siguiente: inicialmente tenemos nuestro poliedro encerrado en un elipsoide, si el centro del elipsoide llegara a ser un punto de nuestro poliedro, habremos terminado nuestra búsqueda. En caso contrario, cortamos el elipsoide a la mitad, y nos fabricamos un elipsoide que contenga a la mitad que debería contener al poliedro (esto se preprecisará más adelante); si el centro de este nueva elipsoide es un punto de nuestro poliedro terminamos. Si no, continuamos este proceso de manera iterativa. La pregunta que uno podría hacerse es: **¿Hasta cuando deberíamos iterar?** Esta respuesta la daremos más adelante, pero básicamente se continúa hasta que el volumen del elipsoide sea tal que hace imposible que contenga al poliedro dado. Precisemos un poco estos conceptos. Enunciaremos sin demostrar una serie de lemas, que juntos prueban el correcto funcionamiento del algoritmo. Inicialmente necesitamos un elipsoide que contenga a P .

Lema 3.2 *Si $P = \{x \in \mathbb{R}^n \mid Cx \leq d\}$ es un polítopo, y tanto C como d tienen sus coordenadas enteras. Entonces $P \subseteq B(0, R)$, donde $R := \sqrt{n} 2^{\langle C, d \rangle - n^2}$.*

Este lema nos da la esfera con que el algoritmo inicia su iteración, lo llamaremos E_0 queda claro que $E_0 = E(A_0, a_0)$; donde $A_0 = R^2 I$ y $a_0 = 0$.

Describiremos el k - ésimo paso de la iteración. Por construcción el elipsoide del paso k

$$E_k := E(A_k, a_k)$$

contiene al poliedro P . El centro de estos elipsoides juegan un papel fundamental en el algoritmo, en cada paso se chequea si $a_k \in P$ y en ese caso paramos el algoritmo habiendo encontrado una solución factible del sistema $Cx \leq d$. En caso contrario ($a_k \notin P$) existirá una ecuación del tipo $c^i x \leq d^i$ que no lo cumpla, o sea $c^i a_k \geq d^i$. El hiperplano $H = \{x \in E(A_k, a_k) \mid c^i x = d^i\}$ pasa por el centro a_k del elipsoide E_k cortándolo en dos mitades, se sabe por construcción que el polítopo P está contenido en la mitad

$$E'(A_k, a_k, c^i) = \{x \in E_k \mid c^i x \leq a_k\}$$

Y construimos el elipsoide E_{k+1} que contenga a la mitad $E'(A_k, a_k, c^i)$ usando las fórmulas dadas en (*). De esta forma nos construimos una sucesión de elipsoides de volúmen cada vez menor que contienen al poliedro P ; como dejamos vislumbrar anteriormente, el problema es cuándo parar, si se da el caso que los respectivos centros no están en P . Los volúmenes de los respectivos elipsoides cumplen la relación presentada en el siguiente lema.

Lema 3.3 $\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} = \left(\left(\frac{n}{n+1}\right)^{n+1} \left(\frac{n}{n-1}\right)^{n-1}\right)^{1/2} < e^{-1/(2n)} < 1$

Las condiciones impuestas sobre el poliedro P garantizan, en el caso que $P \neq \emptyset$, que tenga volúmen positivo (distinto de 0). Usando que el sistema que describe P es de coeficientes enteros y que P es un polítopo, se puede probar el siguiente lema:

Lema 3.4 *Si el polítopo P tiene dimensión n , entonces;*

$$\text{vol}(P) \geq 2^{-(n+1)\langle C \rangle + n^3}.$$

Ahora podemos finalizar nuestro análisis. Dado que el lema anterior nos da una cota inferior del volúmen del polítopo n -dimensional P , y dado que por el lema 3.3:

$$\lim_{k \rightarrow +\infty} \text{vol}(E_k) = 0$$

existirá algún $N \in \mathbb{N}$ tal que

$$P \subseteq E_N$$

y

$$\text{vol}(E_N) < \text{vol}(P)$$

dado que esto es imposible, tenemos garantizado que $P = \emptyset$. Si elegimos $N := 2n((2n + 1) \langle C \rangle + n \langle d \rangle - n^3)$, se ve fácilmente que $\text{vol}(E_N) < 2^{-(n+1)\langle C \rangle + n^3}$ (ver el lema 3.6) quedando así resuelto nuestro problema. Alcanza con iterar hasta N para poder afirmar que el poliedro P es vacío, o dicho de otro modo, el sistema $Cx \leq d$ no tiene una solución factible.

3.1.3. Versión básica

Ahora daremos una descripción pormenorizada del algoritmo que describimos en la sección anterior.

Algoritmo 3.5 *Input: Un sistema de m ecuaciones con n incógnitas de coeficientes enteros. Garantizando que $P = \{x \in \mathbb{R}^n \mid Cx \leq d\}$ es un polítopo.*

Inicialización:

- (a) $k := 0$,
- (b) $N := 2n((2n + 1) \langle C \rangle + n \langle d \rangle - n^3)$
- (c) $A_0 := R^2 I$ con $N := 2n((2n + 1) \langle C \rangle + n \langle d \rangle - n^3)$
 $a_0 := 0$

Iteración:

- (d) Si $k = N$, parar. (Declarar $P = \emptyset$)
- (e) Si $a_k \in P$, parar. (Declarar que a_k es una solución factible del problema)
- (f) Si $a_k \notin P$, elegir una inecuación del tipo $c^i x \leq d^i$ del sistema $Cx \leq d$ que no cumpla a_k .

$$\begin{aligned}
(g) b &:= \frac{1}{\sqrt{c^t A_k c}} A_k c \\
(h) a_{k+1} &:= a_k - \frac{1}{n+1} b \\
(i) A_{k+1} &= \frac{n^2}{n^2-1} (A_k - \frac{2}{n+1} b b^t) \text{ e ir } a(d)
\end{aligned}$$

A continuación demostraremos un lema que prueba el correcto funcionamiento del algoritmo 3.5.

Lema 3.6 *Si $P = \{x \in \mathbb{R}^n \mid Cx \leq d\}$ es un polítopo de dimensión n y $Cx \leq d$ es un sistema de ecuaciones con coeficientes enteros. Si $E_0 = E(A_0, a_0)$ con $a_0 = 0$, $A_0 = R^2 I$ y $R = \sqrt{n} 2^{\langle C, d \rangle - n^2}$ es el elipsoide inicial, y si aplicamos la iteración de el algoritmo 3.5 $N := 2n((2n+1) \langle C \rangle + n \langle d \rangle - n^3)$ veces, entonces:*

$$\text{vol}(E_N) < 2^{-(n+1)\langle C \rangle + n^3} \leq \text{vol}(P)$$

Demostración Dado que $E_0 \subseteq \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq R\}$, se tiene que

$$\text{vol}(E_0) \leq 2^n R^n = n^{n/2} 2^{n\langle C, d \rangle - n^3 + n} = 2^{n(\langle C, d \rangle - n^2 + 1 + \log_2(n)/2)} < 2^{n\langle C, d \rangle}.$$

Por el lema 3.3 en cada iteración el volúmen del nuevo elipsoide disminuye en un factor de por lo menos $e^{-1/(2n)}$. Por lo tanto después de N iteraciones

$$\text{vol}(E_N) < e^{-N/(2n)} \text{vol}(E_0) < 2^{-N/(2n) + n\langle C, d \rangle} = 2^{-(n+1)\langle C \rangle + n^3} \leq \text{vol}(P),$$

donde la última desigualdad se sigue del lema 3.4. De hecho el N fue elegido para que la última igualdad se cumpla. \square

3.1.4. Complejidad y detalles de implementación.

Complejidad

Para estimar la complejidad del algoritmo 3.5 nos pondremos en el peor de los casos, esto quiere decir que el número de iteraciones es $N = O(n^2 \langle C, d \rangle)$. La inicialización hecha en los pasos (a), (b) y (c) pueden obviamente ser realizada en $O(mn)$ operaciones elementales. Claramente (h) requiere $O(n)$ operaciones elementales mientras que (i) requiere $O(n^2)$. Dado que P es un polítopo $m \geq n$ y la complejidad resulta ser $O(mn^3 \langle C, d \rangle)$.

Detalles de implementación

En la descripción del algoritmo 3.5 hemos obviado algunos detalles de implementación. Por ejemplo en el paso (g) el vector b calculado podría tener alguna coordenada irracional y por lo tanto lo sería a_{k+1} , y dado que los números irracionales no tienen representación finita, el centro a_{k+1} del elipsoide no podría ser calculado con exactitud. Con lo cual uno podría aproximar las coordenadas de a_{k+1} por números racionales. Esto quiere decir aproximar a_{k+1} por $\tilde{a}_{k+1} \in \mathbb{Q}^n$, y trasladar el elipsoide E_{k+1} a \tilde{a}_{k+1} , $\tilde{E}_{k+1} = E(A_{k+1}, \tilde{a}_{k+1})$. Probablemente \tilde{E}_{k+1} ya no contiene el semielipsoide $E'(A_k, a_k, c)$ y tampoco al polígono P . Por lo tanto, el argumento de la sucesión de elipses decrecientes en volumen que contienen a P ya no es válido.

En la sección siguiente veremos como puede adaptarse el algoritmo, en el caso del problema de optimización, para resolver estos inconvenientes.

3.2. El método del elipsoide aplicado a problemas de optimización.

Dado $K \subseteq \mathbb{R}^n$ un conjunto convexo y compacto, definiremos dos problemas algorítmicos en conexión con K .

Definición 3.7 *Problema de optimización (fuerte):* Dado $c \in \mathbb{R}^n$, hallar $x \in K$ tal que $c^t x = \max_{z \in K} c^t z$.

Definición 3.8 *Problema de separación (fuerte):* Dado $y \in \mathbb{R}^n$ decidir si $y \in K$ y si así no fuera hallar un hiperplano que separe y de K . Es decir encontrar un $c \in \mathbb{R}^n$ tal que $c^t y > \max\{c^t x \mid x \in K\}$.

Ejemplos 3.9 *Sea K el conjunto solución del sistema $Ax \leq b$, $x \in \mathbb{R}^n$ y $A \in \mathbb{R}^{m \times n}$. El problema de separación es simplemente un problema de programación lineal. Y el problema de optimización se resuelve fácilmente reemplazando y en la ecuación.*

Ejemplos 3.10 *Sean v_1, \dots, v_n vectores de \mathbb{R}^n y $K = \text{conv}\{v_1, \dots, v_n\}$. El problema de optimización se resuelve fácilmente encontrando el máximo valor que toma $c^t x$ entre v_1, \dots, v_n . Y resolver el problema de separación es hallar $y \in \mathbb{R}^n$ tal que*

$$c^t y > c^t v_i \quad 1 \leq i \leq n.$$

nuevamente se trata de resolver un problema de programación lineal.

Por motivos que tienen que ver con la imposibilidad de dar una representación finita de un número irracional, se define la versión débil de estos problemas.

Definición 3.11 *Problema de optimización (débil):* Dado $c \in \mathbb{Q}^n$ y $\epsilon \in \mathbb{Q}_{>0}$ hallar $y \in \mathbb{Q}^n$ tal que $\min_{x \in K} \|y - x\| \leq \epsilon$ y $c^t x \leq c^t y + \epsilon$.

Definición 3.12 *Problema de separación (débil):* Dado $y \in \mathbb{Q}^n$ y $\epsilon \in \mathbb{Q}_{>0}$ decidir si $\min_{x \in K} \|y - x\| \leq \epsilon$ y en caso contrario hallar $c \in \mathbb{Q}^n$ tal que $\|c\| \geq 1$ y $c^t x \leq c^t y + \epsilon \forall x \in K$.

Asumiremos que existe $a_0 \in K$ y $0 < r \leq R$ tal que:

$$B(a_0, r) \subseteq K \subseteq B(a_0, R) \quad (*).$$

La segunda inclusión implica que K está acotado (en la sección anterior vimos como se calcula R en el caso que K sea un polígono), mientras que la primera inclusión indica que K es de dimensión n .

Definición 3.13 *Definiremos un cuerpo convexo a la 5-upla (K, n, a_0, r, R) donde $n \geq 2$, $K \subseteq \mathbb{R}^n$ es un convexo compacto que satisface $(*)$. Y notaremos por \mathcal{K} a la familia de cuerpos convexos.*

Asumiremos que cada $K \in \mathcal{K}$ tiene una representación. Y un input para el problema de optimización será algún $K \in \mathcal{K}$, un vector $c \in \mathbb{Q}^n$ y $\epsilon \in \mathbb{Q}_{>0}$. El tamaño del input será $\langle K \rangle + \langle c \rangle + \langle \epsilon \rangle$. Un algoritmo para el problema de optimización en \mathcal{K} diremos que es polinomial si el tiempo de ejecución es menor que un polinomio en el tamaño del input. Equivalentemente se define para el problema de separación.

Descripción del algoritmo

Supondremos que existe un algoritmo que llamaremos **SEP** que resuelve el problema de separación (débil).

El algoritmo genera una sucesión de $x_k \in \mathbb{R}^n$, que son los centros de los elipsoides E_k , y en cada paso se le aplica el algoritmo **SEP** a $y = x_k$ con un δ conveniente

(**SEP**(δ, y)), si el algoritmo determina que $\min_{x \in K} \|x - y\| < \delta$ llamaremos a k índice factible; en caso contrario nos devuelve un $d \in \mathbb{Q}^n$ tal que $\|d\| \geq 1$ y

$$\max\{d^t x \mid x \in K\} \leq d^t x_k + \delta.$$

Algoritmo 3.14 *Input:* $K \in \mathcal{K}$, $c \in \mathbb{Q}^n$, $\epsilon \in \mathbb{Q}_{>0}$.

Inicialización:

(a) $k := 0$,

(b) $N := 4n^{2\lceil \log_2 \frac{2R^2 \|c\|}{r\epsilon} \rceil}$

(c) $\delta := \frac{R^2 4^{-N}}{300n}$

(d) $p = 5N$

(e) $x_0 := a_0$

(f) $A_0 = R^2 I$

Iteración:

(g) Si $k = N$, parar. (Declarar $x_{opt} = x_j$, tal que $\max\{c^t x_k \mid 0 \leq k < N, k \text{ factible}\}$)

$$c^t x_j =$$

(h) $y = x_k$

(i) $SEP(\delta, y)$

(j) Si $\min_{x \in K} \|x - y\| < \delta \Rightarrow a = c$
si no $a = -d$

(k) $b_k := \frac{1}{\sqrt{a^t A_k a}} A_k a$

(l) $x_k^* := x_k + \frac{1}{n+1} b$

(m) $A_k^* := \frac{2n^2+3}{2n^2} (A_k - \frac{2}{n+1} b b^t)$

(n) $x_{k+1} \approx x_k^*$

(ñ) $A_{k+1} \approx A_k^*$, ir a (g)

El símbolo \approx en (j) y (k) indican que el lado derecho se obtiene haciendo un redondeo de p dígitos binarios a la derecha de la coma.

Funcionamiento del algoritmo.

En cada iteración del algoritmo aparece un nuevo elipsoide $E_k = \{x \in \mathbb{R}^n \mid (x - x_k)^t A_k^{-1} (x - x_k)\}$, pero para que efectivamente se trate de un elipsoide se necesita garantizar que la matriz A_k sea simétrica definida positiva. El próximo lema nos da esa garantía.

Lema 3.15 *Las matrices A_0, A_1, \dots, A_N definidas en el algoritmo 3.14 son simétricas definidas positivas. Y cumplen:*

$$\|x_k\| \leq \|a_0\| + R2^k, \quad \|A_k\| \leq R^2 2^k, \quad y \quad \|A_k^{-1}\| \leq R^{-2} 4^k \quad (3.1)$$

Por otro lado y al igual que en el algoritmo 3.5 el volumen de los elipsoides forman una sucesión decreciente. Y cumplen la siguiente relación.

Lema 3.16 $\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} < e^{-1/5n}$

El siguiente lema nos garantiza la buena construcción de los elipsoides. Dados

$$\xi_k = \{c^t x_j \mid 0 \leq j < k, j \text{ factible}\}, \quad (3.2)$$

$$K_k = \{x \in K \mid c^t x \geq \xi_k\} \quad (3.3)$$

Lema 3.17 $E_k \supseteq K_k$, para todo $k = 1, \dots, N$.

Estos lemas técnicos permiten probar, mediante el siguiente teorema, el correcto funcionamiento del algoritmo 3.14. Pero antes necesitaremos un resultado sobre conos en \mathbb{R}^n .

Proposición 3.18 *Sea C un cono con vértice y , y H un hiperplano que interseca C , entonces el cono truncado $CT = \text{conv}(\{y\}, C \cap H)$ con base $C \cap H$ y vértice y tiene un volumen*

$$\text{vol}(TC) = \frac{1}{n} \text{vol}(C \cap H) \min_{z \in H} \|y - z\|$$

Teorema 3.19 *Sea j un índice factible tal que*

$$c^t x_j = \text{máx}\{c^t x_k \mid 0 \leq k < N, k \text{ factible}\}.$$

Entonces $cx_j \geq \text{máx}\{c^t x \mid x \in K\} - \epsilon$.

Demostración Por los lemas 3.16 y 3.17 tenemos que

$$\text{vol}(K_N) \leq \text{vol}(E_N) \leq e^{-N/4n} \text{vol}(E_0) = e^{-N/4n} R^n V_n,$$

Se $\xi = \max\{c^t x \mid x \in K\}$ e y una solución óptima ($c^t y = \xi$). Sea CT un cono truncado cuya base es $B(a_0, r) \cap \{x \in \mathbb{R}^n \mid c^t x = c^t x_0\}$ y vértice y , se ve facilmente que $CT \subseteq K$. Por lo tanto la región factible K_N definida en 3.2 y 3.3 contiene al cono truncado $CT' = CT \cap \{x \in \mathbb{R}^n \mid cx \geq \xi_N\}$.

Como la distancia de y al hiperplano $c^t x = c^t x_0$ es $c^t \frac{(y-x_0)}{\|c\|}$ se obtiene por la proposición 3.18 que

$$\text{vol}(CT) = \frac{1}{n} r^{n-1} V_{n-1} c^t \frac{(y - a_0)}{\|c\|}$$

Y dado que la altura de CT' es $\frac{\xi - \xi_N}{\xi - c^t x_0}$ veces la altura de CT tenemos que

$$\text{vol}(CT') = \frac{1}{n} r^{n-1} V_{n-1} c^t \frac{(y - a_0)}{\|c\|} \left(\frac{\xi - c^t x_j}{\xi - c^t x_0} \right)^n$$

Por lo tanto

$$\text{vol}(CT') \leq \text{vol}(K_N) \leq e^{-n/4n} R^n V_n$$

Y luego

$$\xi - c^t x_j \leq e^{-N/4n^2} R \left(\frac{\xi - c^t x_0}{r} \right)^{\frac{n-1}{n}} \left(\frac{nV_n}{V_{n-1}} \right)^{1/n} \|c\|^{1/n}$$

y como

$$|\xi - c^t x_0| \leq |c^t(y - x_0)| \leq \|c\| \cdot \|y - x_0\| \leq R \|c\|$$

finalmente probamos

$$\xi - c^t x_j < 2e^{-n/4n^2} \frac{R^2}{r} \|c\| \leq \epsilon$$

□

3.3. Equivalencia entre los problemas de optimización y separación.

3.3.1. Problemas de optimización y separación

En esta sección probaremos la equivalencia entre los problemas de optimización (débil) y separación (débil) para un convexo cualquiera K . Más precisamente pro-

baremos el siguiente teorema.

Teorema 3.20 *Sea \mathcal{K} la familia de cuerpos convexos, existe un algoritmo polinomial para resolver el problema de separación para los elementos de \mathcal{K} , si y sólo si existe un algoritmo polinomial para resolver el problema de optimización para los elementos de \mathcal{K} .*

A una familia \mathcal{K} para la cual existe un algoritmo polinomial que resuelve el problema de optimización (o separación) la llamaremos resoluble.

Definición 3.21 *Sea $K \in \mathcal{K}$ definimos $K^* = \{u \in \mathbb{R}^n \mid u^t x \leq 1\}$*

es fácil ver que $K^* \in \mathcal{K}$, $(K^*)^* = K$ y si $a_0 = 0$

$$B(0, 1/R) \subseteq K^* \subseteq B(0, 1/r)$$

Si \mathcal{K} es la familia de cuerpos convexos tales que $a_0 = 0$, definimos $K^* = \{K^* \mid K \in \mathcal{K}\}$ y vale el siguiente lema:

Lema 3.22 *El problema de separación para la familia \mathcal{K}^* de cuerpos convexos con $a_0 = 0$ es resoluble en tiempo polinomial si y sólo si el problema de optimización es resoluble en tiempo polinomial para la familia \mathcal{K} .*

Demostración (\Rightarrow) Se sigue de la vuelta del teorema 3.20 que se demostrará luego.

(\Leftarrow) Sea $K^* \in \mathcal{K}$, $v \in \mathbb{R}^n$ y $\epsilon \in \mathbb{Q}_{>0}$. Aplicaremos el algoritmo para resolver el problema de optimización a K , con v como función objetivo y un error ϵr , con lo cual obtenemos un vector $z \in \mathbb{Q}^n$ tal que $\min_{x \in K} \|z - x\| \leq \epsilon$, y

$$v^t z \geq \max\{v^t x \mid x \in K\} - \epsilon r.$$

Si $v^t z \leq 1$ entonces $v^t x \leq 1 + \epsilon r$ y por lo tanto $v_0 = \frac{1}{1+\epsilon r} v \in K^*$. Luego $\|v_0\| \leq 1/r$, y resulta que $\min_{x \in K^*} \|v - v_0\| \leq \|v - v_0\| \leq \epsilon$.

Por otro lado si $v^t z > 1$ entonces z es una solución del problema de separación para K^* . De hecho si $z_0 \in K$ y $\|z - z_0\| \leq \epsilon r$, entonces para todo $u \in K^*$ ($\|u\| \leq 1/r$) y por lo tanto

$$z^t u = (z - z_0)^t u + z_0^t u \leq \|u\| \cdot \|z - z_0\| + 1 \leq \epsilon + z_0^t u,$$

quedando así probado que z es una solución del problema separación para K^* . □

Demostración del teorema 3.20 (\Rightarrow) Sin pérdida de generalidad podemos suponer que $a_0 = 0$ y dado que $(\mathcal{K}^*)^* = \mathcal{K}$, si el problema de optimización es polinomialmente resoluble para \mathcal{K} entonces por el lema 3.22 se sigue que el problema de separación es polinomialmente resoluble para \mathcal{K}^* . Luego por la vuelta del teorema 3.20, que probaremos a continuación, se deduce que el problema de optimización es polinomialmente resoluble para \mathcal{K}^* . Y usando nuevamente el lema 3.22 probamos finalmente que el problema de separación es resoluble polinomialmente para \mathcal{K} .

(\Leftarrow) Consideremos el algoritmo 3.14 y probemos que se trata de un algoritmo polinomial, si el algoritmo **SEP** allí indicado es polinomial y por lo tanto el número de dígitos de las coordenadas de a es polinomial; luego el número de pasos que requiere el cálculo de x_{k+1} y A_{k+1} es polinomial; además se puede ver por el lema 3.15 que el número de dígitos de sus coordenadas es polinomial. Probando así que el algoritmo descripto es polinomial. \square

3.3.2. Conjuntos independientes en grafos perfectos.

Aplicaremos el teorema 3.20 para demostrar que existe un algoritmo polinomial para resolver los problemas de máximo conjunto independiente (pesado) y coloreo, en un grafo perfecto G . Aplicando este algoritmo a su complemento \overline{G} se resolverá en tiempo polinomial los problema de clique máximo (pesado) y mínimo cubrimiento por cliques (ver teorema2.1).

Nosotros sabemos por el lema 2.21 y el corolario 1.22 que un grafo G es perfecto si y solo si $\alpha(G, c) = z(G, c)$ para todo $c \in \{0, 1\}^n$. Se podría tratar de calcular $z(G, 1)$, pero el tamaño de este problema no es polinomial en $|V(G)|$ ya que el tamaño de la matriz A_G depende del número de cliques de G , que puede ser exponencial

Sin embargo el número $\vartheta(G)$ introducido por Lovász en 1979 [32] resuelve este problema.

Definición 3.23 Sea $V = \{1, \dots, n\}$. $\vartheta(G)$ es el máximo valor de $\sum_{i,j=1}^n b_{ij}$, donde los b_{ij} son los coeficientes de un matriz B en el cuerpo convexo

$$\{B \in \mathbb{R}_{n \times n} | B \in \mathcal{D} \text{ tr}(B) \leq 1 \text{ y } b_{i,j} = 0 \forall i \neq j \text{ con } (i, j) \in E(G)\}$$

donde \mathcal{D} es el conjunto de las matrices simétricas, definidas positivas. Si B pertenece a este conjunto, se dice que B es un representante de G .

Está probado que $\alpha(G) \leq \vartheta(G) \leq z(G, 1)$, por lo tanto encontraremos un algoritmo polinomial en $|V|$ para los grafos que cumplen la condición $\alpha(G) = \vartheta(G)$, que en particular es cumplida por los grafos perfectos. Más precisamente hallaremos un máximo conjunto independiente pesado del grafo perfecto G , para un $w \in \mathbb{R}^n$ peso dado, que es equivalente a hallar un conjunto independiente máximo en el grafo perfecto $G \circ w$ (ver definición 2.3 y lema 2.4).

Lema 3.24 *Sea G un grafo no dirigido y $B \in \mathbb{R}^{n \times n}$ una matriz que representa G vale que:*

$$\vartheta(G \circ w) = \sum_{i,j=1}^n \sqrt{w_i w_j} \cdot b_{ij} \quad (3.4)$$

Demostración Si B representa G construimos B' un representante de $G \circ w$ de la siguiente forma, reemplazamos cada b_{ij} por un matriz de tamaño $w_i \times w_j$ con coeficientes que valen constantemente $\frac{b_{ij}}{\sqrt{w_i w_j}}$. La matriz B' así obtenida resulta ser una representante de $G \circ w$ y

$$\sum_{i,j=1}^n \sqrt{w_i w_j} \cdot b_{ij} = \sum_{i,j=1}^{n'} b'_{ij} \quad (3.5)$$

Recíprocamente, si B' representa $G \circ w$, entonces reemplazando cada submatriz de $w_i \times w_j$ inducida por las copias de los vértices i y j , por la suma de sus coordenadas dividido por $\sqrt{w_i w_j}$, obtenemos un representante de G que también cumple la ecuación 3.5. \square

Para aproximar $\vartheta(G \circ w)$ con error a lo sumo $\epsilon \in \mathbb{Q}_{>0}$, podemos reemplazar los $\sqrt{w_i w_j}$ de 3.4 por $\omega_{ij} \in \mathbb{Q}$ tales que:

$$|\omega_{ij} - \sqrt{w_i w_j}| < \epsilon / 2n^2$$

donde los denominadores de los ω_{ij} es a lo sumo $2n^2/\epsilon$. En ese caso, se puede ver que $\vartheta(G \circ w)$ difiere del máximo de $\sum_{i,j} \omega_{ij} b_{ij}$, sobre las matrices B que representan G , en a lo sumo $\frac{1}{2}\epsilon$. Por lo tanto se necesita aproximar este último número con precisión $\frac{1}{2}\epsilon$, y ésto puede ser hecho aplicando el método del elipsoide.

Aplicaremos el método del elipsoide al cuerpo convexo en $\mathbb{R}^{n + \binom{n}{2}}$ que se obtiene del conjunto definido en 3.23 dándole un orden a las coordenadas que están

en la diagonal principal y por debajo de ella, sin considerar los $(i, j) \in E(G)$. Se probará que el problema de separación se puede resolver mediante un algoritmo polinomial.

Dado un $b \in \mathbb{R}^{n+2 \binom{n}{2} - |E(G)|}$ consideremos la matriz B que este vector representa. Si $\text{tr}(B) > 1$ el problema de separación se resuelve trivialmente, la solución sería el vector que tiene un 1 en todas sus coordenadas. Supongamos que $\text{tr}(B) \leq 1$. Encontremos una base del espacio vectorial generado por sus columnas; sin pérdida de generalidad podemos suponer que son las columnas $1, \dots, k$. Entonces la submatriz principal $B' \in \mathbb{R}^{k \times k}$ es no singular y $\text{rg} B' = \text{rg} B$. Se puede ver que B es semidefinida positiva si y sólo si B' es definida positiva, que por el criterio de Silver es equivalente a probar que:

$$\det B_t = \det(b_{ij})_{i=1, \dots, t}^{j=1, \dots, t} > 0,$$

para todo $t = 1, \dots, k$. Dado que estos determinantes pueden ser calculados en tiempo polinomial, eso quiere decir que podemos determinar en tiempo polinomial si B pertenece o no al conjunto definido en 2,17. Por lo tanto si encontramos que B no es semidefinida positiva, consideramos el natural t más pequeño tal que $\det B_t \leq 0$. Y sea $\varphi_i = (-1)^i B_t^{it}$ para $i = 1, \dots, t$ (donde B_t^{it} representa el menor (i, t) de la matriz B_t), y $\varphi_i = 0$ si $i > t$. Luego

$$\sum_{i,j=1}^n \varphi_i \varphi_j \beta_{ij} \geq 0 \tag{3.6}$$

para toda matriz semidefinida positiva β . Y por otro lado,

$$\sum_{i=1}^n \sum_{j=1}^n \varphi_i \varphi_j b_{ij} = \det B_t \cdot \det B_{t-1} \leq 0, \tag{3.7}$$

(si $t = 0$, consideraremos $\det B_0 = 1$). Luego se deduce de 3.6 y 3.7 que $(\varphi_i \varphi_j)_{i,j=1}^n$ es una solución del problema de separación.

Luego por el teorema 3.20 se sigue que el máximo valor de $\sum \omega_{ij} b_{ij}$ sobre las matrices B sobre el conjunto definido en 3.23 con precisión $\frac{1}{2}\epsilon$ y respectivamente $\vartheta(G \circ w)$ con precisión ϵ , en un tiempo acotado por un polinomio $|V|, |\log \epsilon|$ y $\log T$; donde T es el máximo valor absoluto de los numeradores y denominadores de los ω_{ij} . Si se sabe que $\alpha(G \circ w) = \vartheta(G \circ w)$, luego el parámetro $\vartheta(G \circ w)$ es entero y lo podemos

calcular considerando $\epsilon = \frac{1}{2}$. Por lo tanto existe un algoritmo polinomial en $|V|$ y $\log \|w\|$ para hallar $\alpha(G \circ w)$ en un grafo perfecto.

Y se puede hallar explícitamente un máximo conjunto independiente en un grafo perfecto como sigue. Comparamos $\alpha(G \circ w)$ con $\alpha(G' \circ w')$ donde G' y w' se obtiene de G y w suprimiendo el vértice 1 de G y su respectiva coordenada de w (esto se puede hacer en tiempo polinomial ya que $G' \circ w'$ resulta perfecto por el teorema 2.4), en caso de que coincidan reemplazamos G por G' y w por w' , en caso contrario, no se realizan cambios. Se continua este procedimiento con el vértice 2 y se sigue así. Al finalizar nos queda un conjunto de vértices que forman un conjunto independiente máximo (pesado) en G .

Queda claro que utilizando esta misma técnica en \overline{G} (que es perfecto debido a 2.1) obtenemos en tiempo polinomial un clique máximo (pesado) en G .

Usando el resultado que acabamos de mostrar se puede hallar un mínimo cubrimientos por cliques o coloreo para un grafo perfecto G , en tiempo polinomial. Esto puede verse en [26].

Si bien, como acabamos de ver, el método del elipsoide permite probar la polinomialidad de estos problemas algorítmicos sobre los grafos perfectos, no nos da un algoritmo combinatorio (implementable computacionalmente) que nos permita resolver estos problemas. Este problema continua actualmente abierto.

Capítulo 4

Teorema fuerte.

Introducción.

El teorema fuerte de los grafos perfectos fue demostrado en el año 2002 [14]. Este resultado, conjeturado por Berge [2], que resistió ser demostrado durante 40 años, fue probado en un paper de 160 páginas. En este capítulo se intentará dar un resumen de esta demostración esbozando las ideas principales de la demostración.

En la primera sección daremos una muy breve descripción de la demostración del teorema fuerte en los grafos sin C_4 [10], que dio origen a la demostración en el caso general.

Recordemos el enunciado del teorema:

Teorema 4.1 *Un grafo es perfecto si y sólo si es Berge.*

La ida es trivial, como ya se mencionó antes. Si bien durante esos cuarenta años no se pudo encontrar una demostración, pudo ser probado para muchos casos particulares, la forma más usual de encontrar estos grafos era considerando grafos que no contuvieran un subgrafo inducido determinado. Por ejemplo:

Grafos prohibidos de cuatro vértices.

- Grafos sin P_4 , [40].
- Sin K_4 ([45], [46], [47]).
- Sin $K_4 \setminus e$, [48].
- Sin (3-*pan* o *paw*), ([34],[27]).

- Sin C_4 , [10].

Grafos prohibidos de cinco vértices.

- Sin *bulls*, [16].
- Sin *darts*, [43].
- Sin *chairs*, [39].

Grafos derivados.

- Grafos de Línea de un grafo $G = (V, E)$ [34], $L(G)$: es un grafo que se obtiene de G , donde los vértices son las aristas $E(G)$, y dos vértices son adyacentes si sus correspondientes aristas tienen un vértice en común.
- Grafo total de un grafo $G = (V, E)$ [34], $T(G)$: es un grafo que se obtiene de G , donde los vértices son los elementos de $V \cup E$ y dos vértices en $T(G)$ son adyacentes si son adyacentes en G , si el vértice es un extremo de la correspondiente arista o si ambas aristas comparten uno de sus extremos.
- Grafo triangulado de un grafo $G = (V, E)$ [30], $L_3(G)$: donde los vértices son los triángulos de G y dos vértices son adyacentes si sus correspondientes triángulos tienen una arista en común.

Grafos especiales

- Grafos circulares ([6],[7]), grafos de intersección de cuerdas en un círculo.
- Grafos planares [44].
- Grafos pretty [33] (grafos donde todo subgrafo inducido tienen una vecindad (conjunto de vecinos) sin P_4 ni $2K_2$).
- Grafos con grado máximo menor que 7 [24] (el grado de un vértice es el número de aristas que lo tienen por extremo).

4.1. Grafos Perfectos sin C_4 .

Definición 4.2 *Un grafo G se dice que no contiene a C_4 , si no contiene como subgrafo inducido a los ciclos de longitud cuatro.*

Definición 4.3 Un subconjunto $S \subseteq V(G)$ se dice que es un conjunto de corte de G , si el número de componentes conexas del grafo $G - S$ es mayor que las de G .

Definición 4.4 Diremos que un subconjunto $S \subseteq V(G)$ es estrellado si consiste de un vértice x y todos sus vecinos $N(x)$.

Lema 4.5 (Chvátal [15]) Un grafo mínimamente imperfecto no contiene un conjunto de corte estrellado.

Definición 4.6 Un grafo G es una 2-join si existe una partición (H_1, H_2) de $V(G)$ tales que existen $A_i, B_i \subseteq H_i$ para $i = 1, 2$, que cumplen:

- Todos los nodos de A_1 son adyacentes a todos los nodos de A_2 y lo mismo pasa con B_1 y B_2 . Y estas son las únicas adyacencias entre H_1 y H_2 .
- Para $i = 1, 2$ H_i tiene al menos un camino que une a A_i con B_i . Si ambos conjuntos fueran de cardinal 1, entonces el grafo inducido por H_i no es un camino sin cuerdas.

Lema 4.7 (Cornuejols y Cunningham [11]) Un grafo mínimamente imperfecto no contiene una 2-join.

Los lemas 4.5 y 4.7 nos dicen que ambas descomposiciones están prohibidas en todo grafo mínimamente imperfecto.

El siguiente teorema de descomposición, demostrado en [14], tiene como consecuencia el teorema fuerte sobre los grafos sin C_4 .

Teorema 4.8 Sea G un grafo Berge sin C_4 entonces es bipartito o el grafo de línea de un grafo bipartito o tiene un conjunto de corte estrellado o una 2-join.

A partir de este resultado se demuestra fácilmente que:

Teorema 4.9 Sea G un grafo Berge sin C_4 entonces G es perfecto.

Demostración Supongamos que G es un grafo Berge sin C_4 e imperfecto. Luego G contiene un subgrafo mínimamente imperfecto H . Claramente H es Berge y no contiene a C_4 como subgrafo inducido. Entonces por el teorema 4.8 se deduce que H debería tener un conjunto de corte estrellado o una 2-join (ya que los grafos bipartitos y sus grafos de línea son perfectos); pero ésto es imposible por los lemas 4.5 y 4.7. Luego G debe ser perfecto como queríamos probar. \square

El teorema 4.8 fue extendido al caso general [14], pero para conseguir el objetivo se introdujeron algunas modificaciones.

4.2. Estrategia de la demostración en el caso general.

Daremos una descripción somera de las estrategias utilizadas en la demostración del teorema fuerte. Para ello daremos previamente las definiciones necesarias.

4.2.1. Definiciones previas y demostración del Teorema Fuerte asumiendo el teorema de descomposición.

Definición 4.10 Diremos que un grafo G es un *double split* si dados $m, n \geq 2$ enteros sus vértices están particionados en los conjuntos $\{a_1, \dots, a_m\}$, $\{b_1, \dots, b_m\}$, $\{c_1, \dots, c_n\}$, $\{d_1, \dots, d_n\}$. Y sus aristas $E(G)$ cumplen las siguientes condiciones:

- $(a_i, b_i) \in E(G) \forall 1 \leq i \leq m$ y $(c_j, d_j) \notin E(G) \forall 1 \leq j \leq n$.
- No existen aristas entre $\{a_i, b_i\}$ y $\{a_{i'}, b_{i'}\}$ para $1 \leq i < i' \leq m$ y existen las cuatro aristas correspondientes entre $\{c_j, d_j\}$ y $\{c_{j'}, d_{j'}\}$ para $1 \leq j < j' \leq n$.
- Hay exactamente dos aristas entre $\{a_i, b_i\}$ y $\{c_j, d_j\}$ para $1 \leq i \leq m$ y $1 \leq j \leq n$, y estas dos aristas son disjuntas.

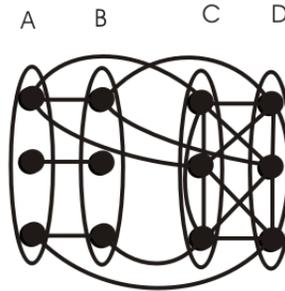


Figura 4.1: Grafo double-split.

Definición 4.11 Diremos que un grafo G es *básico* si G o \overline{G} es un grafo bipartito, el grafo de línea de un grafo bipartito o un grafo double split.

Observación 4.12 Los grafos básicos son perfectos.

A continuación introduciremos una variante de la descomposición *2-join* dada en 4.6, introducida por Cornuejols y Cunningham en [11].

Definición 4.13 *Una proper 2-join de un grafo G es una partición (X_1, X_2) de $V(G)$ tales que existen dos subconjuntos disjuntos $A_i, B_i \subseteq X_i$ ($i = 1, 2$), que satisfacen:*

- *Todo vértice de A_1 es adyacente a todo vértice de A_2 y lo mismo para B_1 y B_2 .*
- *No existen otras aristas entre X_1 y X_2 .*
- *Toda componente conexa de $G[X_i]$ para $i = 1, 2$ interseca a A_i y B_i , y*
- *Si $|A_i| = |B_i| = 1$ y $G[X_i]$ es un camino de A_i a B_i entonces su longitud es ≥ 3 para $i = 1, 2$.*

Sea $X, Y \subseteq V(G)$ diremos que el par (X, Y) es completo si todo vértice de X es adyacente a todo vértice de Y , y se lo llamará anticompleto si no existe ninguna arista entre ellos. Ahora definiremos otra posible descomposición de un grafo G .

Definición 4.14 *Una M -join de un grafo G es una partición en seis subconjuntos no vacíos (A, B, C, D, E, F) , tales que:*

- *Todo vértice de A tiene un vecino y un no-vecino en B y viceversa.*
- *Los pares (C, A) , (A, F) , (F, B) , (B, D) son completos, y*
- *Los pares (D, A) , (A, E) , (E, B) , (B, C) son anticompletos.*

En este capítulo llamaremos un camino en un grafo G a un subgrafo inducido, acíclico, tal que el grado de sus vértices es ≤ 2 , y un anticamino es un subgrafo inducido cuyo complemento es un camino. La longitud de un camino es el número de sus aristas en G y la longitud de un anticamino es el número de aristas en su complemento \overline{G} . Si P es un camino en G notaremos sus vértices internos (aquellos cuyo grado es 2) por P^* , análogamente para un anticamino. Diremos que un par (A, B) de subconjuntos disjuntos de $V(G)$ es balanceado, si no hay un camino de longitud impar, con interior en A , entre dos vértices no adyacentes en B , y no hay un anticamino de longitud impar, con interior en B , entre dos vértices adyacentes de A . Finalmente diremos que un conjunto $X \subseteq V(G)$ es conexo si $G[X]$ resulta conexo, y anticonexo si $\overline{G}[X]$ resulta conexo.

Definición 4.15 (Chvátal) Una skew partition de un grafo G es una partición (A, B) de $V(G)$ tal que A no es conexo y B no es anticonexo.

Esta descomposición posee una dificultad con respecto a las descomposiciones definidas en 4.13 y 4.14, no está probado que no exista una tal descomposición para un grafo mínimamente imperfecto. Éste problema se resuelve si se le pide a tal descomposición que sea balanceada.

Definición 4.16 Un subconjunto conexo maximal de un conjunto no vacío $A \subseteq V(G)$ es llamado una componente de A , y a un subconjunto anticonexo maximal anticomponente.

Definición 4.17 Diremos que una skew partition (A, B) de G es loose si algún vértice de B no tiene vecinos en alguna componente de A , ó algún vértice de A es completo respecto a alguna anticomponente de B .

Se puede ver que:

Lema 4.18 Si G es Berge y admite un loose skew partition entonces admite una skew partition balanceada.

Teorema 4.19 Sea G es un grafo mínimamente imperfecto. Entonces G no admite una skew partition balanceada, y conecuentemente ninguna skew parttion de G es loose.

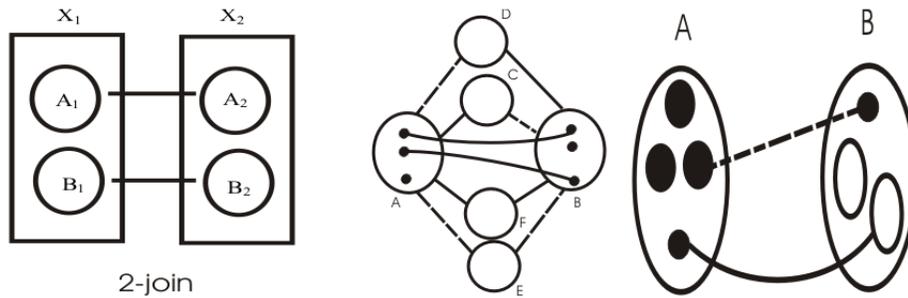


Figura 4.2: 2-join, M-join y skewpartition loose

Teorema 4.20 *Sea G un grafo Berge, G es básico ó G o \overline{G} admiten una proper 2-join, o G admite una M -join, o G admite una skew partition balanceada.*

Daremos la demostración de el teorema 4.1 asumiendo la veracidad del teorema anterior.

Demostración (\Rightarrow) Como ya se vio esta implicación es trivial.

(\Leftarrow) Se probará por el absurdo. Asumamos que G es Berge y *Mínimamente Imperfecto*. Como todo grafo *básico* es perfecto se sigue que G no es *básico*. En [11] se demostró que los grafos *Mínimamente Imperfecto* no admiten una *proper 2-join*. Dado que \overline{G} resulta *Mínimamente Imperfecto* (ver teorema 2.1) y por lo tanto \overline{G} no admite una *proper 2-join*. Se muestra en [16] que un grafo *Mínimamente Imperfecto* no admite una *M-join*, y se probó en [14] que un grafo *Mínimamente Imperfecto* no admite una *skew partition* balanceada. Pero eso contradice el teorema 4.20, luego G debe ser *perfecto*. □

4.2.2. ¿Cómo se puede demostrar el teorema 4.20?

Hay diversos teoremas de esta características en teoría de grafos (ver: [41], [13] y [37]). Estos teoremas básicamente dicen que “dado un grafo que no contiene un objeto del tipo X cae dentro de una clase llamada *básica* o admite algún tipo de *descomposición*”. Las demostraciones de estos teoremas permiten en algunos casos encontrar un construcción para ese conjunto de grafos, pero lo más importante es que todas siguen el mismo patrón.

La demostración de este tipo de teoremas consiste básicamente en una combinación de los siguientes dos métodos:

- Elegimos explícitamente un grafo H apropiado sin X (esto quiere decir que no contiene como subgrafo inducido a X) que no sea *básico*, y se prueba que posee alguna *descomposición* y que ésta puede extenderse a todo grafo más *grande* que contenga a H . Ésto prueba el teorema para todo grafo sin X que contenga a H , por lo tanto, bastará con demostrar que el teorema es válido en aquellos grafos sin X que no contengan a H .

- Se elige un grafo J *básico* y “apropiadamente conectado” (el significado depende del contexto). Y consideramos un grafo G sin X que contenga a J como subgrafo inducido. Luego extendemos a J a un subgrafo maximal K “apropiadamente conectado” y que sea *básico* del mismo tipo que J . Se puede asumir que $K \neq G$. En caso contrario el teorema estaría demostrado. Luego se prueba que el remanente de G junto a K , haciendo uso de la maximalidad de K , induce una descomposición de G . Y ahora nos podemos concentrar en los grafos G sin X que no contengan a J .

El teorema 4.20 se demostró aplicando estas técnicas doce veces. A continuación definiremos unas clases de grafos *Berge* $\mathcal{F}_1, \dots, \mathcal{F}_{11}$ (donde cada clase es subclase del anterior). Pero antes necesitamos algunas definiciones.

Definición 4.21 *Llamaremos subdivisión bipartita de un grafo G , a una subdivisión que da como resultado un grafo bipartito.*

Definición 4.22 *Diremos que $L(H)$ es degenerada, donde H es una subdivisión de K_4 , si existe algún ciclo C de longitud 4 en K_4 que se corresponde con un ciclo en H de la misma longitud. Es decir, hay un ciclo de longitud 4 de K_4 , que no fue subdividido.*

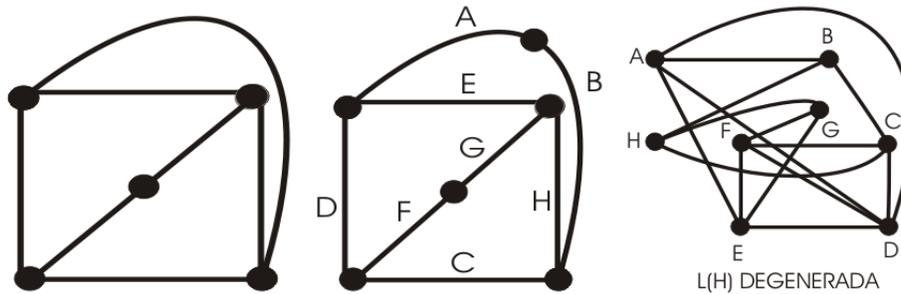


Figura 4.3: K_4 , una subdivisión y una aparición

Definición 4.23 *Un prisma es un grafo compuesto por dos triángulos disjuntos a_1, a_2, a_3 y b_1, b_2, b_3 y tres caminos $P_1 P_2 P_3$ disjuntos por aristas, donde cada P_i tiene por extremos a los vértices a_i y b_i .*

Observación 4.24 Si G es Berge entonces la paridad de los tres caminos de un prisma coincide. Diremos que un prisma es par o impar, si la longitud de estos caminos es par o impar respectivamente.

Definición 4.25 Un prisma largo es un prisma tal que uno de sus caminos tiene longitud > 1 .

Definición 4.26 Diremos que un grafo G es un doble diamante si $|V(G)| = 8$ ($V(G) = a_1, \dots, a_4, b_1, \dots, b_4$) y sus vértices cumplen las siguientes condiciones:

- Todos los a_i 's son adyacentes dos a dos excepto por a_3 y a_4 .
- Todos los b_i 's son adyacentes dos a dos excepto por b_3 y b_4 .
- $a_i b_i \in E(G)$ para todo $1 \leq i \leq 4$.

Definición 4.27 Una wheel (ó rueda) en un grafo G es un par (C, Y) que satisface:

- C es un hole de longitud ≥ 6 .
- Y es un conjunto de vértices anticonexo no vacío disjunto de C .
- C contiene dos aristas disjuntas e Y -completas.

Llamaremos a C la llanta (ó rim) y a Y el centro (hub).

Definición 4.28 Un camino maximal en un camino ó un hole H cuyos vértices son todos Y -completos es llamado un segmento ó Y -segmento en H . Diremos que una rueda (C, Y) es impar si algunos de sus segmentos tiene longitud impar.

Definición 4.29 Diremos que una 3-upla (X, Y, P) en un grafo G es una pseudowheel si satisface las siguientes condiciones:

- X e Y son subconjuntos no vacíos anticonexos de $V(G)$, completo cada uno respecto al otro,
- $P = p_1 - \dots - p_n$ es un camino en $G \setminus (X \cup Y)$ $n \geq 5$.
- p_1 y p_n son los únicos vértices X -completos de P y
- P_1 es Y -completo, p_2 y p_n no lo son.

Observación 4.30 Se puede ver que si (X, Y, P) es una pseudowheel en un grafo $G \in \mathcal{F}_7$ entonces la longitud de P es ≥ 6 y contiene un número impar de aristas Y -completas.

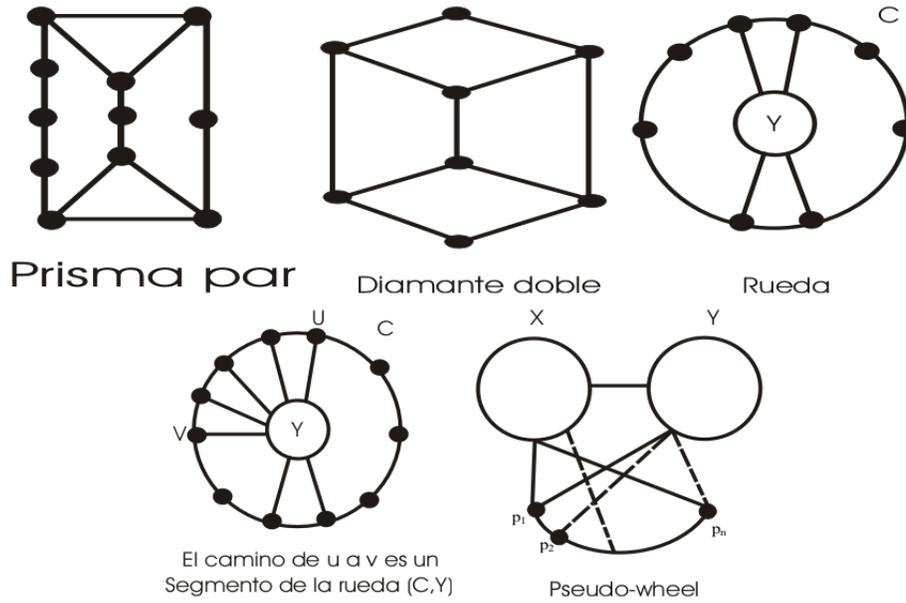


Figura 4.4:

- \mathcal{F}_1 es la clase de todos los grafos *Berge* tales que para toda subdivisión bipartita H de K_4 , todo subgrafo inducido de G isomorfo a $L(H)$ es degenerado.
- \mathcal{F}_2 es la clase de todos los grafos G tales que $G, \overline{G} \in \mathcal{F}_1$ y ningún subgrafo inducido de G es isomorfo a $L(K_{3,3})$.
- \mathcal{F}_3 es la clase de todos los grafos *Berge* G tales que, para toda subdivisión H de K_4 , ningún subgrafo inducido de G o de \overline{G} es isomorfo a $L(H)$.
- \mathcal{F}_4 es la clase de todos los grafos $G \in \mathcal{F}_3$ tales que ningún subgrafo inducido de G es un prisma par.
- \mathcal{F}_5 es la clase de todos los $G \in \mathcal{F}_3$ tales que ningún subgrafo inducido de G o de \overline{G} es un prisma largo.
- \mathcal{F}_6 es la clase de todos los grafos $G \in \mathcal{F}_5$ tales que ningún subgrafo inducido de G es isomorfo a un diamante doble.
- \mathcal{F}_7 es la clase de todos los grafos $G \in \mathcal{F}_6$ tales que G y \overline{G} no contienen una rueda impar.

- \mathcal{F}_8 es la clase de todos los grafos $G \in \mathcal{F}_7$ tales que G y \overline{G} no contienen pseudoruedas.
- \mathcal{F}_9 es la clase de todos los $G \in \mathcal{F}_8$ tales que G y \overline{G} no contienen ruedas.
- \mathcal{F}_{10} es la clase de todos los $G \in \mathcal{F}_9$ tales que, para todo *hole* C en G de longitud ≥ 6 , ningún vértice de G tiene tres vecinos consecutivos en C ; y lo mismo debe valer para \overline{G} .
- \mathcal{F}_{11} es la clase de todos los $G \in \mathcal{F}_{10}$ tales que todo *antihole* en G tiene longitud 4.

Teorema 4.31 (*Pasos de la demostración de el teorema 4.20*)

1. Para todo grafo G Berge, G es un grafo de línea de un grafo bipartito, o G admite una *proper 2-join* o una *skew partition balanceada*, o $G \in \mathcal{F}_1$.
2. Para todo grafo G con $G, \overline{G} \in \mathcal{F}_1$, ó bien $G = L(K_{3,3})$ o bien G o \overline{G} admite una *proper 2-join* ó G admite una *skew partition balanceada*, o $G \in \mathcal{F}_2$.
3. Para todo grafo $G \in \mathcal{F}_2$ o bien G es un grafo *double split*, o G o \overline{G} admiten una *proper 2-join* o G admite una *balanced skew partition* o $G \in \mathcal{F}_3$.
4. Para todo grafo $G \in \mathcal{F}_1$ ó bien G es un *prisma par* con $|V(G)| = 9$ ó G admite una *proper 2-join* ó G admite una *balanced skew partition* ó $G \in \mathcal{F}_4$.
5. Para todo grafo $G \in \mathcal{F}_3$, G ó \overline{G} admiten una *proper 2-join*, ó G admite una *M-join* ó G admite una *skew partiton balanceada* ó $G \in \mathcal{F}_5$.
6. Para todo grafo $G \in \mathcal{F}_5$, G ó \overline{G} admiten una *proper 2-join* ó G admite una *skew partiton balanceada* ó $G \in \mathcal{F}_6$.
7. Para todo grafo $G \in \mathcal{F}_6$, G admite una *skew partition balanceada* ó $G \in \mathcal{F}_7$.
8. Para todo grafo $G \in \mathcal{F}_7$, G admite una *skew partition balanceada* ó $G \in \mathcal{F}_8$.
9. Para todo grafo $G \in \mathcal{F}_8$, G admite una *skew partition balanceada* ó $G \in \mathcal{F}_9$.
10. Para todo grafo $G \in \mathcal{F}_9$, G admite una *skew partition balanceada* ó $G \in \mathcal{F}_{10}$.
11. Para todo grafo $G \in \mathcal{F}_{10}$ ó $G \in \mathcal{F}_{11}$ ó $\overline{G} \in \mathcal{F}_{11}$

12. Para todo grafo $G \in \mathcal{F}_{11}$ ó G admite una skew partition balanceada ó G es completo ó G es bipartito.

En las próximas secciones ilustaremos las técnicas utilizadas, describiendo los dos primeros pasos de la demostración y dando una idea del último.

4.3. Apariciones de K_4 en un grafo Berge

Ahora daremos la idea de como se demostraron los pasos 1. y 2. del teorema 4.31. Empecemos dando algunas definiciones.

Definición 4.32 *Un track es un camino no necesariamente inducido.*

Definición 4.33 *Un vértice de un grafo G se dice que es un branch si tiene grado mayor o igual a 3.*

Definición 4.34 *Un branch de un grafo G es un track P tal que ninguno de sus vértices internos es un vértice branch.*

Definición 4.35 *Diremos que un grafo de línea $L(H)$ es sustancial, si H es una subdivisión de un grafo J 3-conexo. Si $J = K_4$, $L(H)$ es no degenerada.*

Observación 4.36 *Observar que $V(J)$ son los vértices branch de H y los branches de H están en biyección con las aristas de J .*

Definición 4.37 *Sean F y K dos subgrafos inducidos de G , disjuntos en vértices. Diremos que un vértice de K es un attachment de F si tiene un vecino en F .*

Teorema 4.38 *Sea G un grafo Berge, y tiene como subgrafo inducido a $L(H)$ no degenerado donde H es una subdivisión bipartita de K_4 . Entonces G es un grafo de línea, o G admite una proper 2-join, ó G admite una skew partition balanceada. En resumen, los grafos Berge sin $L(H)$ no degenerada, admiten una partición o es un grafo de línea.*

Claramente el lema anterior prueba el paso 1 del teorema 4.20. Dado que si $G \notin \mathcal{F}_1$, debe contener como subgrafo inducido un $L(H)$ no degenerado con H una subdivisión de K_4 . Luego o bien $G = L(H)$, o bien (como consecuencia del lema anterior) G admite una skew partition balanceada, o G admite una proper 2-join.

Definición 4.39 *Un grafo G es cíclicamente 3-conexo si se obtiene por subdivisión de un grafo H 3-conexo.*

Teorema 4.40 *Sea H bipartito y cíclicamente 3-conexo. Entonces $H = K_{3,3}$, ó H es una subdivisión de K_4 , ó H contiene un subgrafo inducido H' tal que H' es una subdivisión de K_4 y $L(H)$ es no degenerada.*

Teorema 4.41 *Sea G un grafo Berge, que contiene un subgrafo inducido isomorfo a $L(K_{3,3})$. Entonces G ó \overline{G} es un grafo de línea, o bien G ó \overline{G} admite una proper 2-join, ó bien G admite una skew partiton balanceada.*

Definición 4.42 *Dados J y G dos grafos. Diremos que J es una aparición en G si existe una subdivisión bipartita H de J , tal que $L(H)$ es isomorfo a un subgrafo inducido de G . Si $J \neq K_4$ diremos $L(H)$ que es una aparición no degenerada en G si $J = H = K_{3,3}$.*

Ambos lemas se pueden fusionar para dar el siguiente resultado, del cual se desprende facilmente el paso dos del teorema 4.31.

Teorema 4.43 *Sea G un grafo Berge, que contiene a $L(K_{3,3})$ como subgrafo inducido. Entonces se cumple alguna de las siguientes condiciones:*

- $G = L(K_{3,3})$, ó
- hay una aparición no degenerada de K_4 en G o \overline{G} ó
- G o \overline{G} admite una proper 2-join, o G admite una skew partition balanceada.

A partir de este lema se prueba fácilmente el paso 2 del teorema 4.20. Si $G \notin \mathcal{F}_2$ entonces G contiene a $L(K_{3,3})$ como subgrafo inducido. Y dado que por hipótesis $G, \overline{G} \in \mathcal{F}_1$, no hay ninguna aparición no degenerada de K_4 en G ni en \overline{G} , luego se desprende por el lema anterior la tesis de 2.4.20.

Definición 4.44 *Diremos que el grafo J' es un alargamiento de el grafo J si J' es 3-conexo y contiene un subgrafo inducido propio, isomorfo a J .*

Se verá que los lemas 4.38 y 4.41 se prueban fácilmente a partir del próximo teorema

Teorema 4.45 *Sea G Berge y J un grafo 3-conexo que cumple:*

- *no hay una aparición no degenerada $L(H)$ de J en G , y no hay un alargamiento de J con una aparición no degenerada en G , o*
- *hay una aparición $L(H)$ de $J = K_{3,3}$ en G , y no hay una aparición de un alargamiento de J en G o \overline{G} .*

Entonces $G = L(H)$, ó G admite una proper 2-join o una skew partition balanceada.

Para ver los detalles de la demostración de este teorema se puede consultar [14].

Demostración del teorema 4.38 asumiendo la veracidad del teorema 4.45 .

Sea G Berge y supongamos que hay una aparición no degenerada de K_4 en G . Luego elijamos un grafo 3-conexo J maximal (en el sentido de los alargamientos de J) tal que hay una aparición no degenerada de J en G ; luego se satisfacen las hipótesis del teorema 4.45, y como consecuencia el lema 4.38 queda demostrado. \square

Demostración del teorema 4.41 asumiendo la veracidad del teorema 4.45. Sea G Berge, y supongamos que contiene un subgrafo inducido isomorfo a $L(K_{3,3})$. Por lo tanto podemos elegir un grafo 3-conexo J maximal (en el sentido de los alargamientos de $K_{3,3}$), tal que haya una aparición de J en G . Si esta aparición es no degenerada el lema queda demostrado, como consecuencia del teorema 4.45. Supongamos ahora que tal aparición resulta ser degenerada ($J = K_{3,3}$), si existe un alargamiento de J en \overline{G} , lo elegimos maximal, entonces, el lema se deduce aplicando el teorema 2.1 sobre \overline{G} . Por lo tanto se puede suponer que no hay un alargamiento de J que aparezca en \overline{G} , cumpliéndose nuevamente las hipótesis del teorema 4.45 y como consecuencia se desprende la tesis del lema 4.41. \square

4.4. Los siguientes pasos

En el paso 3 se ataca el problema en el caso que G no contenga a $L(K(3,3))$

Teorema 4.46 *Si G o \overline{G} contienen una aparición de K_4 que sea degenerada, entonces:*

- G es un grafo double split, o
- G admite una skew partition balanceada.
- G y \overline{G} admiten una proper 2-join.
- No hay apariciones de K_4 en G o \overline{G} .

En el cuarto paso se consideran los grafo *Berge* sin apariciones no degeneradas que contienen un *prisma* par.

Teorema 4.47 *Sea G un grafo Berge sin apariciones no degeneradas. Entonces:*

- G es un prisma par con $|V(G)| = 9$ o
- G admite un proper 2-join, o
- G admite una skew partition balanceada.

En el quinto paso se demuestra el teorema para los grafos *Berge* sin apariciones de K_4 en G o \overline{G} que contienen un prisma largo impar (es decir, sus caminos tienen longitud > 1).

Teorema 4.48 *Sea G un grafo Berge tal que no contiene apariciones de K_4 . Entonces:*

- G y \overline{G} admiten una proper 2-join, o
- G admite una skew partition balanceada.

G admite una M -join.

Teorema 4.49 *Si $G \in \mathcal{F}_5$ y contiene un diamante doble entonces:*

- G y \overline{G} admiten una proper 2-join, o
- G admite una skew partition balanceada.

Definición 4.50 *Diremos que un grafo G es recalitrante si:*

- G es Berge.
- G y \overline{G} no son grafos de línea.
- G no es un double split.
- G y \overline{G} no admiten una proper 2-join .
- G no admite una M -join.
- G no admite una skew partition balanceada.

Teorema 4.51 *Sea $G \in \mathcal{F}_6$ que contiene wheels impares entonces admiten una skew partition balanceada. En particular todos los grafos recalcitrantes pertenecen a \mathcal{F}_7 .*

Recordemos que los grafos en \mathcal{F}_6 son aquellos grafos *Berge* que no contiene *diamantes dobles*, G y \overline{G} no contienen *prismas* largos y tanto G como \overline{G} no contienen apariciones de K_4 .

Teorema 4.52 *Sea $G \in \mathcal{F}_7$ que contiene una pseudowheel entonces admite una skew partition balanceada. En particular todos los grafos recalcitrantes pertenecen a \mathcal{F}_7 .*

La parte final de la demostración se centra en los grafos recalcitrantes. Se ve que:

Teorema 4.53 *Sea $G \in \mathcal{F}_8$ tal que no admite una skew partition balanceada entonces no hay una wheel en G . En particular todo grafo recalcitrante pertenece a \mathcal{F}_9 .*

Por lo tanto consiste en estudiar la familia de grafos que viven en \mathcal{F}_9 . Pero dado que:

Teorema 4.54 *Sea $G \in \mathcal{F}_9$ tal que no admite una skew partition balanceada y C un hole de longitud ≥ 6 . Entonces, no existen vértices en $G \setminus V(C)$ con tres vecinos consecutivos en C . En particular, todo grafo recalcitrante pertenece a \mathcal{F}_{10}*

Pero el estudio se puede reducir aún más estudiando una familia de grafos más pequeña \mathcal{F}_{11} , donde gracias al siguiente lema, se ve que viven los grafos recalcitrantes.

Teorema 4.55 *Sea $G \in \mathcal{F}_{10}$ entonces G no contiene holes y anti-holes de longitud ≥ 6 . En particular, todo grafo recalcitrante pertenece a \mathcal{F}_{11} .*

Par terminar con la demostración queda por probar:

Teorema 4.56 *Sea $G \in \mathcal{F}_{11}$ un grafo entonces:*

- G es completo, o
- G o \overline{G} es bipartito, o
- G admite una skew partition balanceada.

Lema 4.57 Sea G un grafo en \mathcal{F}_6 tal que no admite una skew partition balanceada. Sea $X, Y \subseteq V(G)$ no vacíos disjuntos y completos uno respecto al otro.

- Si $X \cup Y = V(G)$, entonces G es completo o \overline{G} tiene exactamente dos componentes con, a lo sumo, dos vértices cada una.
- Si $X \cup Y \neq V(G)$, entonces $V(G) \setminus (X \cup Y)$ es conexo, y si $|X| > 1$ todos los vértices en X tienen vecinos en $V(G) \setminus (X \cup Y)$.

Idea de la demostración del teorema 4.56 Se supone que existe un grafo G en \mathcal{F}_{11} que no es un grafo bipartito ni que admite una skew partition balanceada. Dado que G no es bipartito contiene un triángulo.

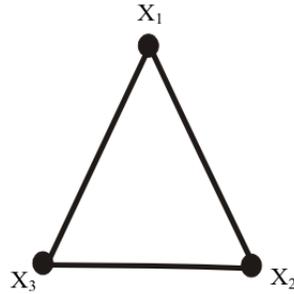


Figura 4.5:

Por lo tanto contiene n ($n \geq 3$) conjuntos de vértices no vacíos, anticonexos y completos entre sí. Se eligen estos vértices de forma tal que su unión sea maximal. Por lo tanto hay dos posibilidades.

$$X_1 \cup X_2 \cup \dots \cup X_n = G$$

o

$$X_1 \cup X_2 \cup \dots \cup X_n \neq G$$

Si se da este último caso, hay vértices de G que no están en $X_1 \cup X_2 \cup \dots \cup X_n$ y se define

$$F = G \setminus X_1 \cup X_2 \cup \dots \cup X_n$$

Se define a $X = X_1 \cup X_2 \cup \dots \cup X_{n-1}$ y $Y = X_n$. Se puede ver que F es conexo y que todo vértice en X tiene un vecino en F . Por lo tanto, todos los vértices de X_1 , X_2 y X_3 tienen un vecino en F . A partir de esto se puede probar que algún vértice de F (llamémoslo v) es completo con respecto a dos conjuntos de X_1, X_2, X_3 .

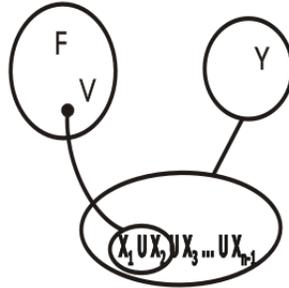


Figura 4.6:

Luego v es completo respecto a algunos X_i y con respecto a otros no. Sean los conjuntos completos respecto v en orden:

$$X_1, \dots, X_i$$

y los no completos:

$$X_i, \dots, X_n$$

Definimos:

$$X'_{i+1} = X_{i+1} \cup \dots \cup X_n \cup v$$

Pero entonces $X_1, \dots, X_i, X'_{i+1}$ viola la optimalidad de la elección de X_1, \dots, X_n , por lo tanto $G = X_1 \cup X_2 \cup \dots \cup X_n$. El lema 4.57 nos permite afirmar que G es completo o \overline{G} tiene exactamente 2 componentes. \overline{G} no puede tener exactamente 2 componentes porque G tiene como mínimo tres conjuntos anticonexos, completos entre sí, por lo tanto \overline{G} tiene tres o más componentes, luego G resulta completo \square

Asi queda probado que todo grafo perfecto es *Berge*

Capítulo 5

Algoritmo de reconocimiento.

Introducción.

Como ya sabemos un grafo es perfecto si y solo si es *Berge*. En este capítulo describiremos un algoritmo polinomial $O(|V(G)|^9)$, que reconoce si un grafo G dado es o no *Berge*. Este algoritmo utiliza una técnica llamada *cleaning*, utilizada anteriormente por Conforti y Rao [9] para reconocer matrices balanceadas.

Básicamente el algoritmo determina si G no es *Berge*, o que G no contiene un *odd hole*. Luego para testear si G es *Berge* aplicamos el algoritmo a G y \overline{G} . Si G no fuera un grafo *Berge*, entonces contiene un *odd hole*, y por lo tanto un *shortest odd hole* C (agujero impar mínimo)). Diremos que un vértice $v \in G \setminus C$ es *C-major*, si sus vecinos en C no pertenecen a un camino de longitud dos de C . A C lo llamaremos *clean* si no contiene vértices *C-major* en G . Los *shortest odd hole (clean)* son fácilmente detectables en un grafo G dado, y básicamente éste será el primer paso del algoritmo. El resto del algoritmo consistirá en reducir el problema al caso “*clean*”.

Un subconjunto $X \subseteq V(G)$ lo llamaremos un *cleaner* para un determinado *shortest odd hole* C en G , si $X \cap V(C) = \emptyset$ y X contiene todos los vértices *C-major*. Observemos que si X es un *cleaner* para C , C resulta *clean* en $G - X$.

La idea central del algoritmo consiste en generar un número polinomial de subconjuntos de $V(G)$, tales que si C es un *shortest odd hole* de G uno de estos subconjuntos resulta ser un *cleaner* para C . Luego para cada uno de estos subconjuntos se testeará si alguno los subgrafos inducidos por $V(G) - X$ contiene un *shortest odd hole clean*. Y ésto será suficiente ya que G contiene un *odd hole* si y

sólo si algunos de los subgrafos inducidos por $V(G) - X$ contiene un *shortest odd hole clean*.

Con la intención de reducir la complejidad del algoritmo, no se hace exactamente lo que se dijo anteriormente. Dado un *shortest odd hole* C en un grafo G , diremos que $X \subseteq V(G)$ es un *near-cleaner* para C si X contiene todos los vértices *C-major*, y $X \cap V(G)$ está incluido dentro un camino de longitud dos en C .

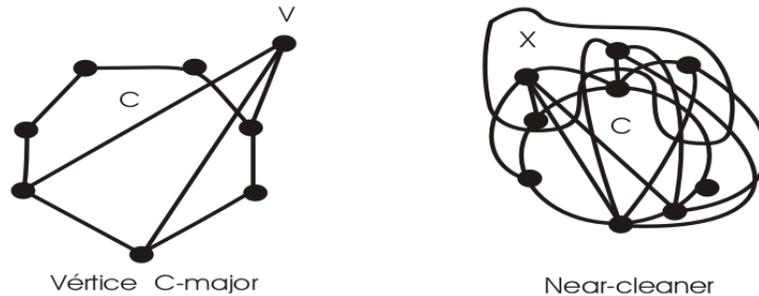


Figura 5.1:

Se sabe como generar un cantidad polinomial de subconjuntos de $V(G)$, de manera tal que uno de ellos sea un *near-cleaner* para un *shortest odd hole* C , sin necesidad de conocerlo explícitamente. En el caso que C sea *amenable*, es decir que cumpla:

- C es un *shortest odd hole* en G , de longitud al menos siete.
- Para todo conjunto anticonexo X de vértices *C-major*, hay una arista de C X -completa.

El algoritmo está dividido en tres rutinas (en orden cronológico) a saber:

Rutina 1: Un algoritmo polinomial que, dado un grafo G y un subconjunto $X \subseteq V(G)$, tal que si X es un *near-cleaner* de un *shortest odd hole* el algoritmo determina la existencia de un *odd hole* en G .

Rutina 2: Un algoritmo polinomial que, dado un grafo G , si contiene algún *shortest odd hole* que no es *amenable*, determina que G no es *Berge*.

Rutina 3: Un algoritmo polinomial que, dado un grafo G , da como resultado una cantidad polinomial de subconjuntos de $V(G)$, tales que si C es un *amenable hole* en G , entonces uno de estos subconjuntos es un *near-cleaner* para C .

Aplicando estas tres rutinas se puede construir un algoritmo polinomial, que determina si un grafo es o no *Berge*. Primero ejecutamos la **rutina 2**, y si no determinara que G no es *Berge*, si hubiera un *shortest odd hole* debería ser *amenable*. Luego ejecutamos la **rutina 3**, y para cada subconjunto de $V(G)$ ejecutamos la **rutina 1**. Si no decidiera que G no es *Berge*, repetimos el procedimiento para \overline{G} . Si tampoco se decidiera que \overline{G} no es *Berge*, entonces se deduce que G es *Berge*.

Este algoritmo es el producto de la investigación de dos grupos de trabajo, Cormuejols, Liu y Vuskovic (**CLV**) y Chudnovsky y Seymour (**ChS**). Ambos grupos obtuvieron simultaneamente la **rutina 1**, utilizando distintos caminos. **ChS** obtuvo las rutinas 2 y 3, **rutina 3** que más tarde fue mejorada por **CLV**. En este capítulo presentaremos las rutinas 1 y 2 de **ChS**, y la **rutina 3** en la versión de **CLV**.

5.1. Rutina 1

5.1.1. Reconocimiento de Pirámides.

Definición 5.1 Sea K una pirámide formada por tres caminos P_1, P_2, P_3 que unen a con b_1, b_2, b_3 respectivamente, diremos que la 10-tupla

$$a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3$$

es un marco de K si:

- Para $i = 1, 2, 3$ s_i es vecino de a en P_i .
- Para $i = 1, 2, 3$ $m_i \in V(P_i)$ satisface $d_{P_i}(a, m_i) - d(m_i, b_i) \in \{0, 1\}$.

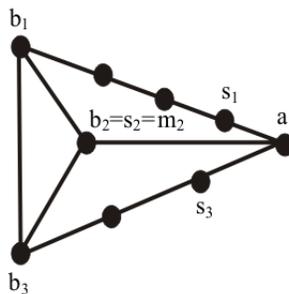


Figura 5.2: Pirámide con marco $a_1, a_2, a_3, s_1, s_2, s_3, m_1, m_2, m_3$

Definición 5.2 Una pirámide K de un grafo G se llama *óptima* si no existe una pirámide K' tal que $|V(K')| < |V(K)|$.

Observación 5.3 Si hay una pirámide en un grafo G entonces hay una pirámide *óptima*. La ventaja de estas últimas es que son más fácilmente detectables.

Lema 5.4 Sea K una pirámide *óptima* en G , con un marco $a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3$. Sean S_1, T_1 los subcaminos de m_1 a s_1 y b_1 respectivamente. Sea F el conjunto de todos los vértices no adyacentes a cada uno de los vértices s_2, s_3, b_2, b_3 . Entonces :

- Sea Q un camino entre s_1 y m_1 con interior en F , y con longitud mínima sobre todos estos caminos. Entonces $P'_1 = a - s_1 - m_1 - T_1 - b_1$ junto con P_2 y P_3 forman una pirámide *óptima*.
- Sea Q un camino entre m_1 y b_1 con interior en F , y con longitud mínima sobre todos estos caminos. Entonces $P'_1 = a - s_1 - S_1 - m_1 - b_1$ junto con P_2 y P_3 forman una pirámide *óptima*.

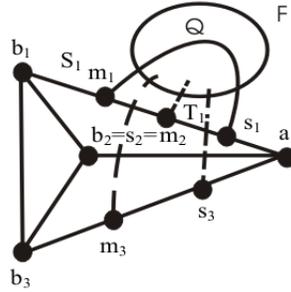


Figura 5.3:

Lema 5.5 Existe un algoritmo que dado un grafo G , encuentra una pirámide (y por lo tanto un odd hole) en G , o determina que G no contiene ninguna pirámide. El algoritmo tiene complejidad $O(|V(G)|^9)$.

Algoritmo • Enumeramos todas las 6-tuplas $b_1, b_2, b_3, s_1, s_2, s_3$ que satisfacen:

1. Para $1 \leq i < j \leq 3$, $\{b_i, s_i\}$ son disjuntos de $\{b_j, s_j\}$ siendo $b_i b_j$ la única arista entre ellos.

2. Hay un vértice a que es adyacente a todos los vértices s_1, s_2, s_3 y a lo sumo a uno de los b_1, b_2, b_3 (si a fuese adyacente a b_i entonces $s_i = b_i$)

Para cada una de las 6-tuplas procedemos como se describe a continuación.

- Sea $M = V(G) \setminus \{b_1, b_2, b_3, s_1, s_2, s_3\}$. Para cada $a, m \in M$ se busca un camino mínimo $S_1(m)$ entre s_1 y m (en el caso que exista) tal que s_2, s_3, b_2, b_3 no tengan vecinos en su interior y un mínimo camino $T_1(m)$ entre b_1 y m (en el caso que exista) tal que s_2, s_3, b_2, b_3 no tengan vecinos en su interior. De manera análoga se buscan $S_2(m), T_2(m), S_3(m), T_3(m)$. Encontrar estos mínimos caminos demanda $O(|V(G)|^2)$ operaciones.

- Para cada $m \in M \cup \{b_1\}$ se definirá un camino $P_1(m)$. Si $s_1 = b_1$, $P_1(b_1)$ es el camino formado por un único vértice b_1 , y si $m \neq b_1$ no está definido. Si $s_1 \neq b_1$ entonces $P_1(b_1)$ no está definido, y para $m \in M$ se testea que cumpla las siguientes condiciones:

1. m no es adyacente a b_2, b_3, s_2, s_3 .
2. Existen $S_1(m)$ y $T_1(m)$.
3. $V(S_1(m) \cap T_1(m)) = \{m\}$
4. No existen aristas entre $V(S_1(m) - m)$ y $V(T_1(m) - m)$.

- Si se cumplieran definimos $P_1(m) := s_1 - S_1(m) - m - T_1(m) - b$ (en caso contrario no está definido). Análogamente se definen $P_2(m)$ y $P_3(m)$. Encontrar $P_i(m)$ par $i = 1, 2, 3$ demanda $O(|V(G)|^3)$ operaciones.

- Se testea para cada 3-upla m_1, m_2, m_3 , si los $P_i(m_i)$ con $i = 1, 2, 3$ forman una pirámide para una elección apropiada de a .

Si $1 \leq i < j \leq 3$, diremos que (m_i, m_j) es un *good* (i, j) -pair si $m_i \in M \cup \{b_i\}$, $m_j \in M \cup \{b_j\}$, existe $P_i(m_i)$ $i = 1, 2$, $V(P_i(m_i)) \cap V(P_j(m_j)) = \emptyset$ y $b_i b_j$ es la única arista entre ellos.

Para cada $m_1 \in M \cup \{b_1\}$ se buscará el conjunto de todos los m_2 tales que (m_1, m_2) es *good*, como describiremos a continuación.

- Si $P_1(m_1)$ no existe entonces no hay ningún *good pair*.
- Si existiera coloreamos de negro todos los vértices $m \in P_1(m_1)$, o que tienen vecinos en $P_1(m_1)$; los otros los coloreamos de blanco (esta operación demanda $O(|V(G)|^2)$ operaciones).
- Para cada $m_2 \in M \cup \{b_2\}$ testeamos si $P_2(m_2)$ existe y no está pintado de negro (para cada m_2 (esto lleva $O(|V(G)|)$ operaciones), por lo tanto para todos $O(|V(G)|^2)$).
- Repitiendo este procedimiento para cada m_1 se puede calcular la lista de todos los $(1,2)$ -*good pairs* en $O(|V(G)|^3)$.

Luego se realiza el mismo procedimiento para calcular los $(1,3)$ y $(2,3)$ -*good pairs*.

- Para cada 3-upla m_1, m_2, m_3 $m_i \in \{b_i\} \cup M$ $i = 1, 2, 3$, testeamos si (m_i, m_j) es un *good (i,j)-pair* $1 \leq i < j \leq 3$. Cada 3-upla lleva un cantidad constante de operaciones, y por lo tanto para todas las 3-uplas $O(|V(G)|^3)$.
- Si se encuentra una 3-upla tal que los tres pares son *good*, paramos y decimos que G contiene una pirámide. Si luego de examinar todas las 6-uplas $b_1, b_2, b_3, s_1, s_2, s_3$, no se encuentra ninguna que cumpla esta condición, diremos que G no contiene pirámides. Dado que por cada 6-tupla se realizan $O(|V(G)|^3)$ operaciones, la complejidad del algoritmo resulta ser $O(|V(G)|^9)$.

□

5.1.2. Reconocimiento de Joyas.

Antes de desarrollar la **rutina 1** hay otras configuraciones que deben ser eliminadas.

Definición 5.6 Diremos que una 5-upla v_1, v_2, v_3, v_4, v_5 de vértices todos distintos es una *joya*, tales que $v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_1$ son aristas, y v_1v_3, v_2v_4, v_1v_4 no lo son, y P es un camino de G entre v_1 y v_4 tal que v_2, v_3 y v_5 no tienen vecinos en P^* .

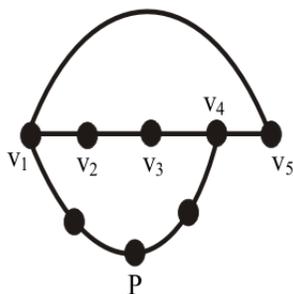


Figura 5.4:

Lema 5.7 *Existe un algoritmo tal que dado G , decide si G contiene una Joya, con complejidad $O(|V(G)|^6)$.*

Algoritmo • Se enumeran todas las 3-uplas v_2, v_3, v_5 de vértices distintos tales que $v_2v_3 \in E(G)$ (este calculo demanda $O(|V(G)|^3)$ operaciones).

- Para cada una de estas 3-uplas calculamos el conjunto F , compuesto por los vértices de G que no son adyacentes a v_2, v_3, v_5 , y se calcula sus componentes.
- Se calculan los conjuntos X_1 de vértices adyacentes a v_2, v_5 y no a v_3 y X_2 adyacentes a v_3, v_5 y no a v_2 .
- Se busca (si existe) una 3-upla $v_1 \in X_1, v_2 \in X_2$ no adyacentes, F' componente de F tales que ambos vértices tengan vecinos en F' .

Estos cálculos demandan $O(|V(G)|^3)$ operaciones.

- Si se encuentra una 3-upla v_1, v_2, F' que cumpla lo pedido G contiene una Joya. Si al finalizar la iteración no se encuentra ninguna, G no contiene una Joya.
- Se obtiene así un algoritmo polinomial con la complejidad del lema. □

Observación 5.8 *Si G contiene una Joya, entonces G contiene un odd hole.*

5.1.3. Detector de shortest *odd hole clean*.

Lema 5.9 *Sea G un grafo que no contiene Joyas ni Pirámides, y sea C un shortest odd hole en G . Dados $u, v \in V(C)$ distintos no adyacentes, y L_1, L_2 los dos posibles subcaminos de C que unen u y v , $|E(L_1)| < |E(L_2)|$. Entonces:*

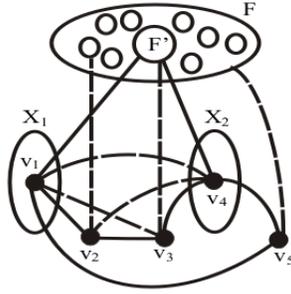


Figura 5.5:

- L_1 es un mínimo camino entre u y v .
- Para todo camino mínimo P entre u , v , $P \cup L_2$ es un shortest odd hole de G , y además clean.

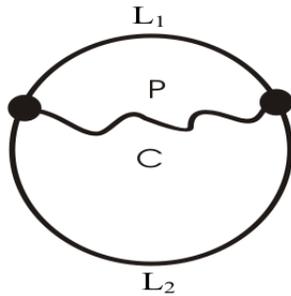


Figura 5.6:

Como consecuencia del lema anterior se desprende.

Lema 5.10 *Existe un algoritmo tal que, dado un grafo G sin Joyas ni Pirámides, determina si G contiene un odd hole, o no hay un clean shortest odd hole. El algoritmo tiene complejidad $O(|V(G)|^4)$.*

Algoritmo • Primeramente calculamos, si existen, los caminos mínimos entre los vértices u y v de G (llamémoslo $P(u, v)$), para cada par de vértices distintos.

- Para cada 3-upla u, v, w se verifica si $P(u, v)$, $P(v, w)$, $P(w, u)$ forman un *odd hole*.

Este algoritmo tiene una complejidad $O(|V(G)|^5)$. Pero esta puede reducirse usando el procedimiento del coloreo blanco y negro usado en 5.5.

□

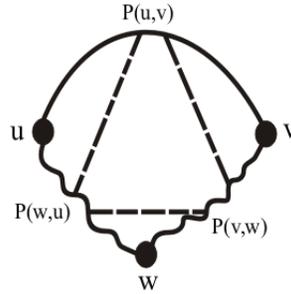


Figura 5.7:

5.1.4. El uso de near-cleaners y final de la rutina 1.

Lema 5.11 *Existe un algoritmo tal que dado un grafo G sin Joyas ni Pirámides y un subconjunto X de $V(G)$, determina si G tiene un odd hole o no hay un shortest odd hole C tal que, X es un near-cleaner para C , con complejidad $O(|V(G)|^4)$*

Demostración Hay un camino simple para construir un algoritmo de estas características: enumerando todos los subconjuntos $Y \subseteq X$ con $|Y| \leq 3$, y aplicando el algoritmo dado en 5.10 a $G \setminus (X \setminus Y)$ para cada Y . Si el algoritmo determina que tiene un *odd hole* entonces uno de estos subgrafos tiene un *clean shortest odd hole*. En caso contrario X no es un *near-cleaner* para ningún *shortest odd hole* de G . Este simple algoritmo cumple lo pedido, pero su complejidad es $O(|V(G)|^7)$ y puede mejorarse la eficiencia como se verá a continuación.

- Para cada par $x, y \in V(G)$ distintos, se busca un mínimo camino $R(x, y)$ entre x e y tal que no contenga vértices internos en X , si tal camino existe notamos la longitud por $r(x, y)$; en caso contrario $r(x, y) = \infty$.

Para cada $y_1 \in V(G) \setminus y_1$ y los caminos de longitud dos $x_1 - x_3 - x_2$ de $G \setminus y_1$, se chequea si las siguientes condiciones se cumplen, donde y_2 es el vecino de y_1 en $R(x_2, y_1)$:

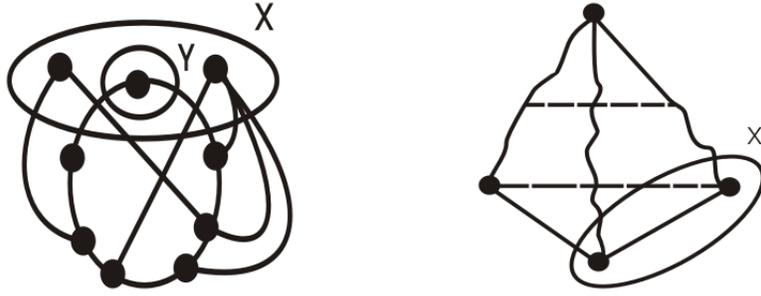


Figura 5.8:

- $r(x_1, y_1), r(x_2, y_2)$ son ambos finitos (y por lo tanto y_2 está definido).
- $r(x_2, y_1) = r(x_1, y_1) + 1 = r(x_1, y_2) := n$.
- $r(x_3, y_1), r(x_3, y_2) \geq n$.

Si se encuentra una 4-upla x_1, x_2, x_3, y_1 , que cumpla estas condiciones el algoritmo reportará que hay un *odd hole*. En caso contrario reportará que no hay un *shortest odd hole* en C tale que X sea un *near cleaner* para C .

Quedaría por ver que el algoritmo funciona correctamente. Supongamos que existe una 4-upla x_1, x_2, x_3, y_1 que satisface las tres condiciones, y sean y_2 y n como antes. Probaremos que en ese caso G contiene un *odd hole*. Sea $R(x_1, y_1) = p_1 - \dots - p_n$, $R(x_2, y_1) = q_1 - \dots - q_{n+1}$, donde $p_1 = x_1, p_n = q_{n+1} = y_1, q_1 = x_2$ y $q_n = y_2$. De la definición de $R(x_1, y_1)$ y $R(x_2, y_1)$ se sigue que los vértices $p_2, \dots, p_{n-1}, q_2, \dots, q_n$ no pertenecen a X , y de la elección de y_1 se sigue que $y_1 \notin X$ (posiblemente x_1, x_2, x_3 pertenezcan a X).

Dado que $r(x_1, y_1) = r(x_2, y_1) - 1$, se sigue que $x_2 \notin R(x_1, y_1)$, por el mismo motivo x_1, x_2 no son adyacentes y $x_1 \notin R(x_2, y_1)$. Dado que $r(x_3, y_1), r(x_3, y_2) \geq n$, se sigue que $x_3 \notin R(x_1, y_1), R(x_2, y_1)$, y no tiene vecinos en $R(x_1, y_1) - \{x_1\}, R(x_2, y_1) - \{x_2\}$. Como consecuencia de que $r(x_1, y_2) = n, y_2 \notin R(x_1, y_1)$. Probaremos a continuación que los vértices p_2, \dots, p_{n-1} son distintos de q_2, \dots, q_n . Supongamos que $p_i = q_j$ para algún $2 \leq i \leq n - 1$ y $2 \leq j \leq n$. Entonces ambos subcaminos de p_i a y_1 resultan ser subcaminos de un camino mínimo, por lo tanto deben tener la misma longitud, es decir $j = i + 1$, por lo tanto $p_1 - \dots - p_i - q_{j+1} \dots - q_n$ contienen un camino

entre x_1, y_2 de longitud $\leq n - 2$, contradiciendo el hecho que $r(x_1 y_2) = n$. Entonces $R(x_1, y_1)$ y $R(x_2, y_1)$ tienen a y_1 como único vértice en común. Si no hubiera aristas entre $R(x_1, y_1) - \{y_1\}$ y $R(x_2, y_1) - \{y_1\}$ la unión de $R(x_1, y_1) - y_1$, $R(x_2, y_1) - y_1$ y $x_1 - x_3 - x_2$ forman un *odd hole* y el algoritmo funciona correctamente. Ahora asumamos que $p_i q_j$ es una arista siendo $1 \leq i \leq n - 1$ y $1 \leq j \leq n$ y probaremos que $i \geq j$. Si $j = 1$ es trivial, asumamos que $j > 1$. En ese caso habría un camino entre x_1, y_2 con vértices en $\{p_1, \dots, p_i, q_j, \dots, q_n\}$, con lo cual su longitud es $\leq n - j + i$ y no tiene ningún vértice interno en X (dado que $j > 1$), y como $r(x_1, y_2) = n$, se sigue que $n - j + i \geq n$ y por lo tanto $i \geq j$ como afirmamos previamente. Consecuentemente $i \geq 2$, dado que x_1, x_2 son no adyacentes. Pero además, $r(x_2, y_1) \geq n$, y por lo tanto $j + n - i \geq n$, esto es $j \geq i$. Y finalmente queda probado que $i = j$, y eligiendo i mínimo resulta que $x_3 - p_1 - \dots - p_i - q_i - \dots - q_1 - x_3$ es un *odd hole*, y nuevamente en este caso el algoritmo funciona correctamente. Por lo tanto cuando el algoritmo determina que G tiene un *odd hole* lo hace correctamente.

Faltaría probar que cuando el algoritmo determina que no hay un *shortest odd hole* C en G tal que X es un *near cleaner* para C , lo hace correctamente. Supongamos que X es un *near cleaner* para un *shortest odd hole* en C . Por lo tanto, para alguna elección de x_1, x_2, x_3, y_1 , todos estos cuatro vértices yacen en C , $d_c(x_2, y_1) = d_c(x_1, y_1) + 1 := n$, y C es un *clean shortest odd hole* de longitud en H , donde $H = G - (X - \{x_1, x_2, x_3\})$. Notar que en particular $V(C) \cap X \subseteq \{x_1, x_2, x_3\}$. Probemos que si $u, v \in V(C)$ entonces $r(u, v) \geq d_H(u, v) = d_C(u, v)$. Si $R(u, v)$ existe entonces alguno de sus vértices internos están en X , por lo tanto es un camino en H ; y como consecuencia $r(u, v) \geq d_H(u, v)$. Por el lema 5.9 resulta $d_H(u, v) = d_C(u, v)$, probando así nuestra afirmación.

Dado que $r(x_1, y_1) \leq d_C(x_1, y_1)$ (porque $V(C) \cap X \subseteq \{x_1, x_2, x_3\}$), se sigue de la afirmación anterior que $r(x_1, y_1) = n - 1$ y $R(x_1, y_1)$ resulta un camino mínimo en H entre x_1 e y_1 . Por el lema 5.9 podemos elegir C tal que $R(x_1, y_1)$ es un camino de C . Similarmente $r(x_1, y_1) = n$, y podemos asumir que $R(x_2, y_1)$ es un camino en C . En particular $y_2 \in V(C)$. Usando el mismo argumento, $r(x_1 y_2) = d_C(x_1, y_2) = n$. Ahora $d_C(x_3, y_1) = n$, y por la afirmación anterior resulta $r(x_3, y_1) \geq n$, y similarmente resulta $r(x_3, y_2) \geq n$. También en este caso el algoritmo corre correctamente, determinando que G tiene un *odd hole*, funcionando correctamente en todos los casos.

En $O(|V(G)|^3)$ busca todos los mínimos caminos, chequear las 4-uplas x_1, x_2, x_3, y_1 lleva $O(1)$ operaciones. Por lo tanto el algoritmo corre en $O(|V(G)|^4)$. Como se afirma en el enunciado del lema. \square

5.2. Rutina 2.

Comencemos definiendo algunas estructuras importantes.

Definición 5.12 Una configuración del tipo \mathcal{T}_1 en un grafo G , es un hole de longitud 5.

Definición 5.13 Una configuración del tipo \mathcal{T}_2 en un grafo G es una sucesión v_1, v_2, v_3, v_4, P, X tal que:

- $v_1 - v_2 - v_3 - v_4$ es un camino en G .
- X es un anticomponente del conjunto formado por todos los vértices $\{v_1, v_2, v_3\}$ -completos.
- P es un camino en $G \setminus (X \cup \{v_2, v_3\})$ entre v_1, v_4 y ningún vértice de P^* es X -completo ó adyacente a v_2 ó adyacente a v_3 .

Definición 5.14 Una configuración del tipo \mathcal{T}_3 en G es una sucesión v_1, \dots, v_6, P, X tal que:

- v_1, \dots, v_6 son vértices distintos de G .
- $v_1v_2, v_3v_4, v_1v_4, v_2v_3, v_3v_5, v_4v_6$ son aristas, y $v_1v_3, v_2v_4, v_1v_5, v_2v_5, v_1v_6, v_2v_6, v_4v_5$ son no adyacentes.
- P es un camino en $G \setminus (X \cup \{v_1, v_2, v_3, v_4\})$ entre v_5, v_6 y ningún vértice de P^* es X -completo ó adyacente a v_1 ó adyacente a v_2 .
- Si v_5v_6 es una arista entonces v_6 no es X completo.

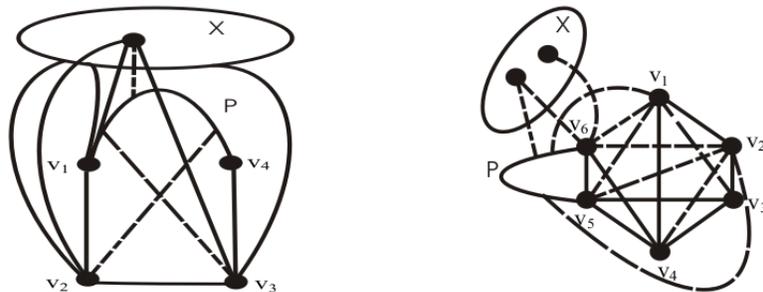


Figura 5.9: Configuraciones \mathcal{T}_2 y \mathcal{T}_3 .

Claramente se puede testear en $O(|V(G)|^5)$ si un grafo G contiene una configuración del tipo \mathcal{T}_1 . A continuación daremos resultados análogos para las otras dos configuraciones.

Lema 5.15 *Existe un algoritmo tal que dado un grafo G , resporta si G tiene una configuración del tipo \mathcal{T}_2 con complejidad $O(|V(G)|^6)$.*

Demostración Enumeramos todos los caminos $v_1 - v_2 - v_3 - v_4$ de G . Para cada uno calculamos el conjunto Y de todos los vértices $\{v_1v_2, v_4\}$ -completos. Luego buscamos todas las anticomponentes de Y . Para cada anticomponente de X testeamos si hay un camino P entre v_1, v_4 en $G - \{v_2, v_3\}$, tal que ningún vértice interno en P sea X -completo. y este algoritmo evidentemente cumple con las condiciones del lema. \square

Lema 5.16 *Si G es Berge y X es un conjunto de vértices anticonexo, y P es un camino impar en $G - X$ tal que ambos extremos de P son X -completos y ninguna arista de P es X -completa. Entonces todo vértice X -completo tiene un vecino en P^* .*

Corolario 5.17 *Si G tiene una configuración \mathcal{T}_2 entonces G no es Berge.*

Demostración Sea v_1, v_2, v_3, v_4, P, X una configuración \mathcal{T}_2 de G . Si P es par entonces $v_1 - v_2 - v_3 - v_4 - P - v_1$ es un *odd hole*, mientras que si P es impar entonces dado que el vértice v_2 es X -completo y no tiene vecinos en P^* , y como los extremos de P son X -completos y los vértices internos no lo son, se sigue del lema 5.16 que G no es Berge. \square

Lema 5.18 *Existe un algoritmo tal que dado un grafo G determina si tiene una configuración \mathcal{T}_3 , en $O(|V(G)|^6)$.*

E numerar todas las 3-uplas v_1, v_2, v_5 de vértices distintos, tales que v_1v_2 es una arista y v_5 no es adyacente a v_1 ni a v_2 . Para cada 3-upla hallada buscar el conjunto Y formado por todos los vértices $\{v_1, v_2, v_5\}$ -completos y buscar sus anticomponentes. Para cada anticomponente X , buscar un subconjunto maximal conexo F' que contiene a v_5 , con la propiedad que v_1v_2 no tiene vecinos en F' y ningún vértice en $F' \setminus \{v_5\}$ es X -completo. Sea F la unión de F' y todos los vértices X -completos que no son adyacentes a v_1, v_2, v_5 y tiene un vecino en F' . Para la misma 3-upla y X enumeramos todos los vértices v_4 que son adyacentes a v_1 y no a v_2, v_5 , y tiene un vecino en F y un no-vecino en X . Para cada elección de v_4 se testea si hay un vértice v_3 adyacente a v_2, v_4, v_5 y no a v_1 , con un no vecino en X . Si encontramos un tal v_3 , dado v_6 un vecino de v_4 en F , y P un camino de v_6 a v_5 con interior en F' , entonces v_1, \dots, v_6, P, X es una configuración del tipo \mathcal{T}_3 . Si luego de chequear todos los v_1, v_2, v_5, X, v_4 no se encuentra un v_3 , entonces no hay una tal configuración.

Este algoritmo es evidentemente un correcto test par la configuración \mathcal{T}_3 . Dado que hay $O(|V(G)|^3)$ 3-uplas v_1, v_2, v_5 que examinar, par cada 3-upla hay $O(|V(G)|)$ elecciones de X , y cada una toma $O(|V(G)|^2)$ ser procesada. Por lo tanto la complejidad de este algoritmo es $O(|V(G)|^6)$ \square

Para probar que si G contiene una configuración tipo \mathcal{T}_3 no es *Berge*. Se necesita el siguiente lema de Russel-Rubio [38].

Lema 5.19 *Sea G Berge, X un conjunto anticonexo de $V(G)$ y $P = p_1 - \dots - p_n$ un camino impar en $G \setminus X$ con longitud ≥ 5 , tal que p_1, p_n son X -completos y p_2, \dots, p_{n-1} no lo son. Entonces existe un par $x, y \in X$ no adyacentes, tal que hay exactamente dos aristas entre x, y y P^* , xp_2 e yp_{n-1} .*

Corolario 5.20 *Sea G Berge, $X \subseteq V(G)$ conexo. Y $v_1, v_2, v_3, v_4, p_1, p_2$ vértices distintos de $G \setminus X$ tales que:*

- v_1v_3 y v_2v_4 son aristas, y no hay aristas entre $\{v_1, v_3\}$ y $\{v_2, v_4\}$.
- v_1, v_4 tienen un vecino en X , ningún vértice de X es adyacente a v_3 y v_4 simultaneamente; y v_1, v_2 no tienen vecinos en X .
- $p_1, p_2 \notin E(G)$, p_1 es adyacente a v_1, v_2, v_4 y no a v_3 , y por último;
- Hay un camino entre v_3, v_4 con interior en X tal que p_1, p_2 no tienen vecinos en X .

Finalmente se deduce a partir de este lema:

Lema 5.21 *Si G contiene una configuración del tipo \mathcal{T}_3 . Entonces G no es Berge.*

Demostración Sea G Berge y supongamos que v_1, \dots, v_6, P, X es una configuración del tipo \mathcal{T}_3 en G . Sea Q un anticamino entre v_3, v_4 con interior en X , puede ser completado a un *antihole* via $v_4 - v_5 - v_1 - v_3$, por lo tanto Q es impar y $v_1 - v_3 - Q - v_4 - v_2$ es un anticamino. Por el lema 5.19, aplicado a \overline{G} al camino $v_1 - v_3 - Q - v_4 - v_2$ y el conjunto anticonexo $V(P)$, se deduce que existe $p_1, p_2 \in V(P)$, adyacentes en G , tales las únicas aristas en \overline{G} entre p_1, p_2 y $V(Q)$ son p_1v_4, p_2v_3 . Aplicando el lema anterior a \overline{G} se deduce que algún $z \in X$ no es adyacente a v_3, v_4 ó p_1, p_2 son ambos X -completos. Lo primero es imposible dado que $z - v_1 - v_4 - v_3 - v_5 - z$ no es un *odd hole*. Lo segundo es imposible dado que ningún vértice interno de P es X -completo, y si v_5, v_6 son adyacentes, entonces por hipótesis v_6 no es X -completo. Por lo tanto G no contiene una tal configuración. Quedando así demostrado el teorema. \square

Recordemos que un *hole* C de G es *amenable* si:

- C es un *shortest odd hole* de G de longitud al menos 7, y
- Para todo conjunto anticonexo X de vértices C -majors hay una arista X -completa en C .

El siguiente resultado da por concluida la **rutina 2**.

Lema 5.22 *Sea G un grafo que no contiene pirámides, configuraciones $\mathcal{T}_1, \mathcal{T}_2$ o \mathcal{T}_3 y G ó \overline{G} no contienen joyas. Entonces todo shortest odd hole en G es amenable.*

5.3. Algoritmo Cleaning.

Comenzaremos ahora con los preliminares de la rutina 3.

Definición 5.23 *Notaremos por $N(a, b)$ el conjunto de todos los vértices $\{a, b\}$ - completos. Y una 3 - upla (a, b, c) la llamaremos relevante si $a \neq b$ y $c \notin N(a, b)$ (posiblemente $c \in \{a, b\}$). Para estas 3 - uplas tenemos que:*

- $r(a, b, c)$ es el cardinal de la anticomponente de mayor cardinal de $N(a, b)$ que contiene un vértice no adyacente a c (ó 0 si c es $N(a, b)$ - completo).
- $Y(a, b, c)$ es la unión de las anticomponentes de $N(a, b)$ que tienen cardinal estrictamente mayor que $r(a, b, c)$.
- $W(a, b, c)$ es la anticomponente de $N(a, b) \cup \{c\}$ que contiene a C .
- $Z(a, b, c)$ es el conjunto de todos los vértices $Y(a, b, c) \cup W(a, b, c)$ - completos.
- $X(a, b, c) = Y(a, b, c) \cup Z(a, b, c)$.

El algoritmo se basa en el siguiente lema:

Lema 5.24 *Sea C un shortest odd hole en G , con longitud a lo sumo 7. Entonces hay una 3 - upla (a, b, c) relevante de vértices tales que:*

- *El conjunto de todos los vértices C -majors que no están en $X(a, b, c)$ es anticonexo.*
- *$X(a, b, c) \cap V(C)$ es un subconjunto de algún conjunto de vértices formado por un camino de longitud 2 en C .*

Lema 5.25 *Existe un algoritmo polinomial, de complejidad $O(|V(G)|^5)$, que genera $O(|V(G)|^5)$ subconjuntos de $V(G)$, tales que si C es un amenable hole en G , entonces uno de estos subconjuntos es un near-cleaner para C .*

Demostración Describiremos el algoritmo. Para cada par de vértices u, v adyacentes en G calculamos $N(u, v)$, y listamos todos tales conjuntos. Para cada 3 - upla relevante (a, b, c) calculamos $X(a, b, c)$ y listamos todos tales conjuntos. La salida del algoritmo serán todos los subconjuntos formados por la unión entre un conjunto de la primera lista y la segunda.

Para probar que el algoritmo funciona correctamente, supongamos que C es un amenable hole en G . Por el lema anterior existe una 3 - upla (a, b, c) que satisface el lema. Dado que el conjunto de todos los vértices C - major, que llamaremos T , que no están en $X(a, b, c)$ es anticonexo y C es amenable, existe una arista uv de C que es T - completa, por lo tanto $T \subseteq N(u, v)$. Pero entonces $N(u, v) \cup X(a, b, c)$ es un near-cleaner de C , y este es uno de los conjuntos de la salida del algoritmo antes descrito. \square

5.4. Algoritmo de Reconocimiento de grafos Berge.

Sólo hay que reunir las piezas desarrolladas en las secciones anteriores.

Lema 5.26 *Existe un algoritmo polinomial, de complejidad $O(|V(G)|^9)$, que dado un grafo G determina si es Berge.*

Demostración Se aplica los algoritmos 5.4, 5.6, 5.16, 5.17 a G y \overline{G} y se analiza la existencia de un configuración \mathcal{T}_1 (observando si cada 5-upla de vértices de G forma o no un hole o antihole. Si se encuentra alguna de estas estructura el algoritmo determina que G no es Berge. En caso contrario por el lema 5.21 sabemos que todo shortest odd hole es amenable. Luego usando el algoritmo 5.24 se obtiene $O(|V(G)|^5)$ subconjuntos. Para cada subconjunto X aplicamos el algoritmo 5.10 al par X, G , si encuentra un odd hole el algoritmo habrá determinado que G no es Berge. Si después de terminar el algoritmo 5.10 no encontró un odd hole, se repite todo el procedimiento para \overline{G} (a partir de la aplicación del algoritmo 5.24). Si nuevamente

no se detecta un *odd hole*, el algoritmo determina que G es *Berge* y por lo tanto es perfecto.

Faltaría probar:

G no es *Berge* \Leftrightarrow el algoritmo determina que G no es *Berge*

(\Leftarrow) Trivial.

(\Rightarrow) Supongamos que G no es *Berge*, podemos suponer (sin pérdida de generalidad) que G contiene un *odd hole*. Luego hay un *shortest odd hole* en G que llamaremos C y supongamos también que en la primer fase del algoritmo no se encuentra ningún *odd hole*, es decir, tanto G como \overline{G} no contiene *joyas*, *piramides* y tampoco configuraciones \mathcal{T}_i con $i = 1, 2, 3$. Como consecuencia del lema 5.21 C es *amenable* y luego sabemos por 5.24 que alguno de los conjuntos X es un *near cleaner* para C . Finalmente el algoritmo 5.10 aplicado al par (G, X) encontrará un *odd hole*. □

Capítulo 6

Variantes de grafos perfectos.

Existen otras familias de grafos, que por sus similitudes con los grafos perfectos describiremos someramente en este capítulo.

6.1. Grafos K -perfectos

Los grafos K -perfectos son un caso particular de grafos de intersección, en el capítulo 1 vimos que relación existe entre los grafos perfectos y su poliedro asociado. en esta sección enunciaremos un resultado similar (ver [25], [3]) para los grafos que son *clique Helly* y K -perfectos.

Definición 6.1 *Sea $K(G)$ el grafo de intersección de las cliques del grafo G , es decir los vértices son las cliques de G y dos vértices son adyacentes en $K(G)$ si y sólo si sus respectivas cliques tienen intersección no vacía. A un grafo G se lo llama K -perfecto si y sólo si $K(G)$ es perfecto.*

Definición 6.2 *Se dice que un familia de conjuntos cumple la propiedad de Helly, si para toda subfamilia tal que sus conjuntos tienen intersección no vacía dos a dos, entonces la intersección de todos los conjuntos de esta subfamilia es no vacía.*

Definición 6.3 *Diremos que un grafo G es clique Helly si la familia de conjuntos compuesta por las cliques de G cumple la propiedad de Helly (CH), y clique-Helly hereditario si todo subgrafo inducido de G es clique-Helly (HCH).*



Figura 6.1: A la derecha vemos el grafo G con sus cliques enumeradas y la izquierda $K(G) = K_4$, G es K -perfecto

Recordemos que un grafo G sea perfecto es equivalente a que G sea *pluperfecto*, es decir el poliedro $P_G = \{x \in \mathbb{R}^n | K_G x \leq 1, x \geq 0\}$ es entero (K_G es la matriz de clique de G). El siguiente teorema relaciona los grafos que son K -perfectos y clique *Helly* con la matriz de clique de G transpuesta .

Teorema 6.4 [25] G es K -perfecto y clique *Helly* si sólo si $P_G = \{x \in \mathbb{R}^n | (K_G)^t x \leq 1, x \geq 0\}$ es entero.

6.2. Grafos Clique Perfectos

Definición 6.5 Se denomina conjunto transversal de cliques del grafo G a un conjunto de vértices que interseca a todas las cliques de G . Un conjunto independiente de cliques de G es un conjunto de cliques de G tal que su intersección es vacía dos a dos. Notaremos:

$\alpha_C(G)$ = cardinal de un conjunto independientes de cliques máximo.

$\tau(G)$ = cardinal de un conjunto transversal de cliques de G mínimo.

Un grafo G es clique perfecto si $\alpha_C(H) = \tau(H) \forall H$ subgrafo inducido de G .

Observación 6.6

$$\alpha_C(G) \leq \tau(G) \forall G$$

Definición 6.7 Un grafo r -sol generalizado es un grafo compuesto por un ciclo impar tal que si dos vértices consecutivos forman una clique con otro vértice del

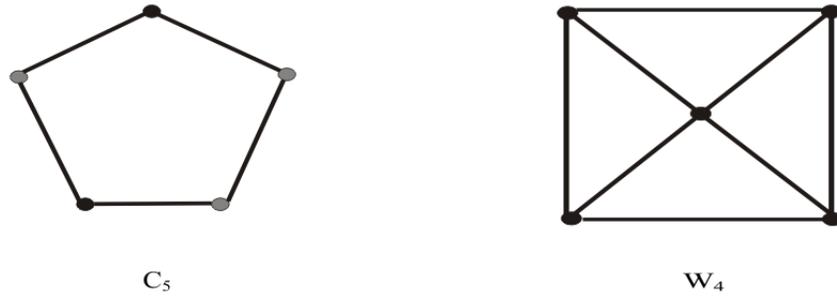


Figura 6.2: C_5 no es clique perfecto ya que $\alpha_C(G) = 2$ y $\tau(G) = 3$, W_4 es un grafo clique perfecto.

ciclo, existe otro vértice que no esta en el ciclo que tiene como vecinos únicamente a estos dos vértices. Por otro lado todo vértice del grafo pertenece al ciclo o forma un clique con dos vértices consecutivos que pertenecen al ciclo.

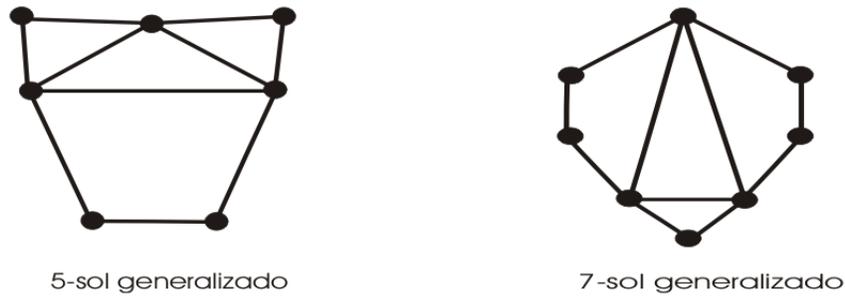


Figura 6.3: s-soles generalizados con $s = 5$ y $s = 7$

Teorema 6.8 ([33], [3]) *Los grafos r -sol generalizado con $r \geq 3$ no son clique perfectos.*

Veamos a continuación que relación existe entre los grafos perfectos y los clique perfectos. Se puede ver que el grafo 5-sol generalizado [25] no es clique perfecto, pero sin embargo es perfecto. Cabe preguntarse si un grafo perfecto es o no clique perfecto. Lamentablemente el siguiente resultado nos muestra que los grafos perfectos no son una subfamilia de los grafos clique perfectos.

Teorema 6.9 [3] *Los grafos $\overline{C_{2r+1}}$ si $2r+1$ no es múltiplo de 3 no es clique perfecto y en caso contrario si lo es.*

Un problema abierto para esta familia de grafos es encontrar una caracterización mediante subgrafos prohibidos (como en el caso de los grafos perfectos). A continuación veremos una serie de resultados parciales obtenidos en esta dirección cuando éstos pertenecen a una determinada clase. Otra problema abierto es si existe un algoritmo polinomial de reconocimiento para esta familia de grafos.

Definición 6.10 *Diremos que un grafo G es interesante si ningún subgrafo inducido es isomorfo a un sol generalizado impar ó un antihole de longitud mayor a 5 e igual a 1,2 (módulo 3).*

Los siguientes resultados pueden verse en: [4], [5].

Teorema 6.11 *Sea G un grafo que no contiene a $K_4 - e$ como subgrafo inducido. Entonces, los siguientes enunciados son equivalentes:*

- G es interesante.
- Ningún subgrafo inducido es isomorfo a un sol generalizado impar.
- G es clique-perfecto.

Teorema 6.12 *Se G un grafo que no contiene a $K_{1,3}$ como subgrafo inducido y es HCH. Entonces, los siguientes enunciados son equivalentes:*

- G es interesante.
- Ningún subgrafo inducido es isomorfo a un odd hole ó $\overline{C_7}$.
- G es clique-perfecto.

Teorema 6.13 *Sea G es un grafo de línea. Entonces, los siguientes enunciados son equivalentes:*

- G es interesante.
- Ningún subgrafo inducido es isomorfo a un odd hole ó a un 3-sol generalizado.
- G es clique-perfecto.

Bibliografía

- [1] C. Berge; *Les problès de colotations en thèorie des graphes*, Publ. Inst. Est. Univ. Paris, 9 (1960) 67-75.
- [2] C. Berge; *Farbung von Graphen, deren samtliche bzw. deren ungerade Kreise starr sind* , Wiss. Z. Marthin-Luther-Univ. Halle Wittenberg Math.-Natur. Reihe 10 (1961), 114.
- [3] F. Bonomo ,G. Durán ,M. Groshaus and J. Szwarcfiter , *On clique- perfect and K-perfect graphs*, Ars Combinatoria (2004), en prensa.
- [4] F. Bonomo,G. Durán , M. Chudnovsky, *Partial characterizations of clique-perfect graphs I: subclasses of claw-free graphs*, enviado a Discrete Applied Mathematics (2005).
- [5] F. Bonomo,G. Durán , M. Chudnovsky, *Partial characterizations of clique-perfect graphs II: diamond-free and Helly circular-arc graphs*, enviado a Discrete Applied Mathematics (2005).
- [6] M. A. Buckyngham M. C. Golumbic ; *Partitionabla graphs, circle graphs an the strong perfect graph conjeture*, Ann. Discrete Math. 44 (1983) ,45-54.
- [7] M. A. Buckyngham M. C. Golumbic; *Recent results on the strong perfect graph conjeture*, Ann. Discrete Math. 20 (1984) ,75-82.
- [8] M. Conforti ; *$K_4 - e$ -free graphs and star cutsets*, Lecture notes in Math. 1403 (1989), 236-253.
- [9] M. Conforti and M.R. Rao; *Testing balancedness and perfection of linear matrices*, Mathematical Programming 61 (1993), 1-18.

- [10] M. Conforti, G. Cornuejols, C. Vuskovic *Square-Free Perfect Graphs*, Journal of Combinatorial Theory B 90 (2004), 257-307.
- [11] G. Cornuejols y W. H. Cunningham; *Composition for perfect graphs*, Discrete Math. 160 (1996), 165-175.
- [12] G. Cornuejols ,M. Chudnovsky, , X. Liu, P. Seymour and K. Vuskovic *Recognizing Berge Graphs*, Combinatorica 25 (2005) 143-186.
- [13] W. Mc. Cuaig; *Pólya's permanent problem*,preprint,1996.
- [14] M. Chudnovsky, N. Robertson, P. Seymour and R. Thomas; *The Strong Perfect Graph Theorem* Annals of Mathematics,.
- [15] V. Chvátal; *Star cutsets and perfect graphs*, J. Combin Theory B 39 (1985) 189-199.
- [16] V. Chvátal y N. Sbihi *Bull-free Berge graphs are perfect*, Graphs and Combinatorics 3 (1987), 127-139.
- [17] V. Chvátal; *On certain polytoped associated with graphs*, Graph Combin. 3 (1987), 127-139
- [18] G. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, Chapter 21 in: T. C. Koopmans (ed.), Activity Analysis of Production and Allocation (Cowles Comission Monograph No. 13), Willey, New York, 1951, 339-347.
- [19] L.R. Ford, Jr., and D.R. Fulkerson; *Flows in networks*, Princeton University Press (1962).
- [20] D. R. Fulkerson; *The graph conjeture and pluperfect graph theorem*, 2nd Chapell Hill Conf. on Combin.Math anda its Apl., (1969) 171-175.
- [21] D. R. Fulkerson; *Blocking polyhedra*,Graph theory end its applications, Ed. B. Harris (Academic Press 1970) p. 93.
- [22] D. R. Fulkerson; *Blocking and antiblocking pairs of polyhedra*,Math. programming 1 (1971) 168-194.
- [23] D. R. Fulkerson; *Antiblocking polyhedra*,J. Combin. Theory, 12 (1972) 50-71.

- [24] C. M. Grinstead ; *The strong perfect graph conjecture for a class of graphs*, Ph. D. thesis ,UCLA 25 (1978).
- [25] Marina E. Groshaus; *Sobre grafos perfectos, clique perfectos, K -perfectos y coordinados*, Tesis de licenciatura (2001).
- [26] M. Grötschel, L. Lovász y A. Schrijver; *The ellipsoid method and its consequences in combinatorial optimization* ,Combinatorica, volumen 1, Nro. 2, 1981.
- [27] W. L. Hsu ; *How to color claw-free perfect graphs* , Ann. Discrete Math. 11 (1981), 189-197.
- [28] L.G. Kachiyan; *A polynomial algorithm in linear programming.*, Doklady Akademii Nauk SSSR 244 (1979), 1093-1096 (English translation: Soviet Math. Dokl. 20, 191-194).
- [29] V. Klee and G. J. Minty, *How good is the simplex algorithm?*, in: O. Shisha (ed.), Inequalities III, Academic Press, New York, 1972, 159-175.
- [30] V. B. Le ; *Perfect k -line graphs and k -total graphs*, J. Graph theory 17 (1993) ,65-73.
- [31] L. Lovász; *A characterization of perfect graphs the perfect graph conjecture*,J. Combin. Theory, B 1 (1972) 95-98.
- [32] L. Lovász; *On the Shannon capacity of a graph.*, IEEE Trans. on Information Theory 25 (1979) 1-7.
- [33] M. Maffray, M. Preissmann ; *Split-neighbourhood graphs and the strong perfect graph conjecture*, J. Combin. Theory B 63 (1995) ,294-309.
- [34] K. R. Parthasarthy, G. Ravindra ; *The strong perfect graph conjecture is true for $K_{1,3}$ -free graphs*, J. Combin. theory, 21 (1976), 212-223.
- [35] H. J. Prömel, A. Steger; *Almost all Berge graphs are perfect.*, Combin. Probab. Comput. 1 (1992), 53-79.
- [36] J.T. Robacker; *Min-max theorems on shortest chains and disjoint cuts of network.*, The Rand Corporation, RM-1660-PR, 1956.

- [37] N. Robertson, P. Seymour and R. Thomas ; *Permanents, Pfaffian orientations, and even directed circuits*, Annals of Math. , 150 (1999), 929-975.
- [38] F. Roussel and P. Rubio, *About skew partition in minimal imperfect graphs*, J. Combinatorial Theory, Ser. B 83 (2001), 171-190.
- [39] A. Sassano ; *Chair free Berge graphs are perfect* ,Graphs Combin. 13 (1997) ,369-f5.
- [40] D. Seinshe ; *On a property of the class of n -coloreable graphs*, J. Combin. Theory B, 16 (1974), 191-193.
- [41] P. Seymour; *Tutte's edge-coloring conjecture*, J. Combinatorial Theory, Ser. B 70 (1997),166- 183 volumen 36 (1990).
- [42] N. Z. Shor *Convergence rate of the gradient descent method with dilatation of the space*, Kibernetika 2 (1970) 80-85 (English translation: Cybernetics 6 (1970) 102-108) .
- [43] L. Sun ; *Two classes of perfect graphs* , J. Combin. Theory B. 42 (1991) , 273-291.
- [44] A. C. Tucker ; *The strong perfect graph conjecture for planar graphs*, Canad. J. Math. 25 (1973) 103-114.
- [45] A. C. Tucker ; *Critical perfect graphs and perfect 3-chromatic*, J. Combin. Theory B. 23 (1977), 143-149.
- [46] A. C. Tucker ; *The validity of the strong prefect graph conjecture for K_4 -free graphs*, Ann. Discrete Math. 21 (1984),149-158.
- [47] A. C. Tucker ; *A reduction procedure for coloring perfect K_4 -free graphs*, J. Combin. Theory B. 43 (1987), 151-172.
- [48] A. C. Tucker ; *Coloring $K_4 - e$ -free graphs* , J. Combin. Theory B. 42 (1987), 313-318.