



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

P-PARTICIONES CONVEXAS EN UNA FAMILIA DE GRAFOS CONSTRUIDOS
MEDIANTE REEMPLAZOS

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

FELIPE GUILLERMO CONTRERAS SALINAS

PROFESOR GUÍA:
MARTÍN MATAMALA VÁSQUEZ

MIEMBROS DE LA COMISIÓN:
JOSÉ SOTO SAN MARTÍN
MAYA STEIN

Este trabajo ha sido parcialmente financiado por CONICYT mediante Proyecto Basal PFB

03

SANTIAGO DE CHILE

2016

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO
POR: FELIPE GUILLERMO CONTRERAS SALINAS
FECHA: 2016
PROF. GUÍA: SR. MARTÍN MATAMALA VÁSQUEZ

P-PARTICIONES CONVEXAS EN UNA FAMILIA DE GRAFOS CONSTRUIDOS
MEDIANTE REEMPLAZOS

Un conjunto de vértices de un grafo se dice convexo si contiene a los vértices de todos los caminos mínimos entre sus vértices. El problema de determinar si un grafo tiene una partición en p conjuntos convexos es NP-completo, pero hay diversas familias de grafos en las que se puede resolver en tiempo polinomial.

El foco principal de este trabajo es estudiar el problema de la p -partición convexa en una familia de grafos definida recursivamente al reemplazar los vértices de un bosque por grafos de ésta. Esta familia resulta ser cerrada para subgrafos inducidos. Sin embargo, no queda totalmente determinada la familia de subgrafos prohibidos que determina a esta familia. Además, estos grafos resultan ser perfectos, por lo que varios problemas combinatoriales resultan tener soluciones polinomiales en esta familia. Además, se entrega un algoritmo polinomial para reconocer si un grafo pertenece a esta familia.

Para atacar el problema de las particiones convexas en este contexto, combinaremos, mediante programación dinámica, particiones en subgrafos tales que sus particiones convexas son de tres tipos: todos los vértices, particiones en cliques y particiones en cliques más un conjunto localmente convexo. En el caso de las particiones en cliques, se entrega un algoritmo polinomial que lo resuelve.

Never give up. Trust your instincts.
— James McCloud

Agradecimientos

Esta será una lista latera de leer, así que el lector (¿por qué alguien leería esto?) queda advertido.

Parto por agradecer a mi familia, quienes me han apoyado en todas las tonteras que me da por hacer, incluida la de estudiar esta carrera. También a la gente del A-0, que, aunque se suele decir no aporta mucho en lo humano, aún tiene (o tenía) gente destacable. En particular, van saludos para Pingo, Choley, René, Astromédico, Zhorí, Santana, Aránguiz y más gente que se me va en este momento.

Agradezco a la comunidad de Melee, por mantener vivo este juego tan bacán y tan fallado. A la gente con la que compartimos la comunidad de la U: RoZu, Isol, Jano, Alvarado, Nacho y demases. Al Fake por inyectarle aire nuevo a la escena y al resto de los cabros, por formar una comunidad de la que da gusto ser parte. En especial a Carlos, Danny, Gino, Verce, Dark, Crono, Mistgun, Touché, Blaky, Rodrillas, Bou, Roche.

Agradezco a la gente con la que compartí la experiencia de la programación competitiva, la cual me recordó que sin consistencia no se mejora en nada. A los compañeros de equipo: Nico, Rodolfo, Ronie y Paulsen. A los rivales, de los que se aprendió harto (salvo a Montoto, porque no sé quién es): Vonbo, Nico Sanhueza, Barbieri, Ralonso. Y a los coach, por el *coaching* para la competencia y para la vida: Wax, Camilo, Miguel y @perez.

Agradezco a la gente de la rama de Taekwondo, por recordarme que los límites los pone uno.

Agradezco a la gente con la que he compartido en la U. A la gente motivada con diversas iniciativas (¡grandes los grupos organizados!), salvo a los colectivos políticos y a Canal 6, que me caen mal. A Memo, por brígido. A la gente del DIM, por compartir y fomentar tanta estupidez. A Sebastián (*a.k.a* Mario) por recomendarme *mangas* y hacerme perder el tiempo leyéndolos. Al Beto y al Enzo por ser viciados pal Melee. A Karina, Javiera, Ivana y Valentina por por ser bacanes :3. A los funcionarios del DIM, que para algunos son invisibles, pero le dan vida y humanidad al departamento.

Agradezco a los profesores que hacen que la facultad valga la pena. Parto por mi comisión de tesis, en especial a Martín Matamala, por ser la voz de la experiencia. También a José Soto, Maya Stein, Marcos Kiwi, Cristián Reyes y Jorge Pérez, por dar los cursos que valió la pena tomar. Además a Nico y a Vonbo por ser los buenos auxiliares.

En resumen, agradezco al RNG por hacer que este *run* fuera lo que fue.

Y mentí, porque falta Edgardo a quien no sé por donde empezar a dar las gracias, así que no lo haré. Saludos.

Eso, muéranse todos >:(

Tabla de Contenido

Tabla de Contenido	ix
Introducción	1
1. Preliminares	3
1.1. Notación	3
1.2. Convexidad	4
1.3. Algunas operaciones en grafos	7
1.3.1. Producto cartesiano	7
1.3.2. Suma	10
2. Familia \mathcal{T}^*	14
2.1. Propiedades básicas	16
2.2. Reconocimiento	19
2.2.1. $\text{diam}(G) \geq 3$	20
2.2.2. $\text{diam}(G) \leq 2$	22
2.2.3. Complejidad de REC	25
2.3. Estructura	26
2.3.1. Subgrafos prohibidos	26
2.3.2. Relación con familias conocidas	30
3. Convexidad	40
3.1. Convexidad en \mathcal{T}^*	42
3.1.1. Particiones en cliques	45
3.1.2. Particiones localmente convexas	48
3.1.3. Programación Dinámica	49
Conclusión	51
Bibliografía	53

Introducción

En teoría de grafos, una clase de problemas de interés es la de particionar los vértices de un grafo en conjuntos que cumplan una determinada propiedad. Por ejemplo, el problema de coloreo de vértices consiste en particionar los vértices en conjuntos independientes. Considerando el complemento de un grafo, toman también importancia las particiones en cliques. En este trabajo, la propiedad que nos interesa está ligada a esta última y es la convexidad, que corresponde a ser cerrado para caminos mínimos.

Sin embargo, encontrar estas particiones o determinar si un grafo dado tiene una partición de un tamaño específico suele ser un problema difícil de resolver computacionalmente. Un enfoque común en estos casos es restringirse a familias de grafos en las que resolver estos problemas sea más simple. Generalmente, una de las familias en las que es fácil resolver estos problemas corresponde a los árboles. Una vez atacado el problema en los árboles, se busca alguna manera de generalizar el resultado a una familia más grande de grafos, tratando de recuperar alguna propiedad de los árboles relevante al problema. Por ejemplo, los árboles tienen la propiedad de que es posible ir eliminando vértices para mantener alguna invariante interesante, como suele ser el caso de sus hojas. Esta propiedad se recupera en los grafos cordados, los cuales se pueden desagregar mediante sus vértices simpliciales, esto es, vértices cuyos vecinos conforman un clique. Esta propiedad es base de muchos de los algoritmos que para esta familia resuelven fácilmente problemas que son difíciles en el caso general.

En este trabajo, nos enfocaremos en otra posible generalización, que recupera el hecho de que en los árboles se puede realizar programación dinámica. Esto es, al retirar un vértice del árbol, se pueden resolver los subproblemas que aparecen en los subárboles obtenidos al borrar el vértice y luego combinar estas soluciones para resolver el problema en el árbol original.

La familia de grafos estudiada en este trabajo se construye a partir de un bosque, expandiendo sus vértices en grafos de ésta misma. En el primer nivel de esta recurrencia están los bosques. Una primera pregunta a resolver cuando se define una familia de grafos es tratar de caracterizarla de alguna forma o relacionarla con otras familias ya conocidas. En este contexto, se demuestra que la familia de este trabajo es cerrada para subgrafos inducidos, por lo que se puede caracterizar por un conjunto de subgrafos prohibidos. Sin embargo, no se logra determinar en su totalidad tal conjunto. Desde el punto de vista algorítmico, se desea poder reconocer a los grafos de la familia, es decir, dado un grafo arbitrario, queremos responder mediante un algoritmo si éste pertenece a la familia en cuestión. En este trabajo se entrega un algoritmo de reconocimiento polinomial, el que además será base de algoritmos para otros problemas sobre estos grafos.

Como se mencionó antes, el enfoque para encontrar particiones convexas en esta familia será dividir el grafo en subgrafos en donde sea más fácil particionar en convexas y luego combinar estas particiones para encontrar una partición del grafo original. Así, nos interesaremos en dos tipos de objetos en nuestra familia. Los primeros son las particiones en cliques, para el cual se entrega un algoritmo polinomial que las encuentra. El segundo tipo de objetos involucra una noción relacionada con la convexidad, mas no muy estudiada en la literatura: la convexidad local. Este nuevo tipo de conjuntos presenta un nuevo problema, el que queda abierto en este trabajo.

Capítulo 1

Preliminares

1.1. Notación

En esta sección haremos algunas definiciones básicas, pero que en la literatura no necesariamente siguen una notación estándar.

Definición 1.1 Sea $k \in \mathbf{N}$. Definimos $[k] = \{1, \dots, k\}$.

Definición 1.2 Sea G un grafo. Denotaremos por $V(G)$ al conjunto de sus vértices y por $E(G)$ al conjunto de sus aristas. Dado un vértice $v \in V(G)$, denotaremos por $N_G(v)$ al conjunto de sus vecinos, es decir

$$N_G(v) = \{u \in V(G) : uv \in E(G)\}.$$

Si no hay ambigüedad, omitiremos el subíndice. Como trabajaremos con grafos simples, se tiene que $v \notin N(v)$.

Definición 1.3 Sea $U \subseteq V(G)$, donde G es un grafo. Diremos que U es un clique en G si todo par de vértices de U está en $E(G)$. Diremos que U es independiente si ningún par de vértices de U está en $E(G)$.

Definición 1.4 Dado un grafo G , su complemento, denotado por \overline{G} , es el grafo tal que $V(G) = V(\overline{G})$ y $E(\overline{G}) = \{uv : u, v \in V(G), u \neq v, uv \notin E(G)\}$.

Definición 1.5 Sean G un grafo y $S \subseteq V(G)$. El subgrafo de G inducido por S , $G[S]$, está dado por $V(G[S]) = S$, $E(G[S]) = \{uv \in E(G) : u, v \in S\}$. Si H es subgrafo inducido de G , lo denotaremos por $H \sqsubseteq G$.

Definición 1.6 Sean G un grafo y $S \subseteq V(G)$. La operación de borrar los vértices de S a G consiste en tomar el grafo $G \setminus S = G[V(G) \setminus S]$.

Definición 1.7 Sean G_1, G_2 grafos. La **unión disjunta**, o simplemente **unión**, de ellos,

$G_1 \cup G_2$, corresponde al grafo dado por $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$ y $E(G_1 \cup G_2)$, donde supondremos que $V(G_1)$ y $V(G_2)$ son disjuntos.

Definición 1.8 Dado un grafo G , llamaremos un (u, v) -paseo en G , o simplemente paseo, a una secuencia de vértices $u = w_0, w_1, \dots, w_n = v$ tal que $v_i v_{i+1} \in E(G)$, $\forall i \in \{0, \dots, n-1\}$. Si todos los vértices w_i , $i \in \{0, \dots, n\}$ son distintos, entonces diremos que es un (u, v) -camino.

Definición 1.9 Sea $P = (u_0, u_1, \dots, u_n)$ un paseo en un grafo G . Entonces denotamos por $\|P\|$ a su largo, es decir, $\|P\| = n$.

Definición 1.10 Dados u, v vértices de G , su **distancia** será

$$d_G(u, v) = \min\{\|P\| : P \text{ es un } (u, v)\text{-camino}\}.$$

En el caso en que no hayan (u, v) -caminos en G , entonces $d_G(u, v) = \infty$. Cuando no haya ambigüedad sobre G , solo escribiremos $d(\cdot, \cdot)$ en vez de $d_G(\cdot, \cdot)$. Si P es un (u, v) -camino tal que $\|P\| = d(u, v)$, entonces diremos que P es un (u, v) -camino mínimo.

Definición 1.11 Sea S un conjunto de vértices de un grafo G . Definimos

$$I_G[S] = \{w \in V(G) : w \text{ está en un } (u, v)\text{-camino mínimo para algún par } u, v \in S\}.$$

También denotaremos $I_G[u, v] = I_G[\{u, v\}]$.

Definición 1.12 Sea G grafo. Dados dos vértices u, v de G , llamaremos $\mathcal{G}_G(u, v)$ al conjunto de (u, v) -caminos mínimos en G . Si no hay ambigüedad, omitiremos el subíndice.

Definición 1.13 Sean $S, T \subseteq V(G)$ no vacíos. Definiremos su distancia como

$$d(S, T) = \min\{d(u, v) : u \in S, v \in T\}.$$

Definición 1.14 Sea G un grafo. Su **diámetro**, $\text{diam}(G)$, se define como la máxima distancia entre un par de vértices de G .

Definición 1.15 Sean G un grafo y M un conjunto de vértices de G . Diremos que M es un **módulo** de G si $\forall x \notin M : M \subseteq N(x) \vee N(x) \cap M = \emptyset$.

Algunos ejemplos triviales de módulos de G son $V(G)$, los singleton, las componentes conexas de G y las componentes conexas de su complemento.

1.2. Convexidad

En esta sección, revisaremos algunos resultados generales sobre la noción de convexidad, los cuales usaremos en capítulos posteriores.

Definición 1.16 Sea G un grafo. Un conjunto S de vértices de G se dice **convexo** si ningún vértice en un (u, v) -camino mínimo está fuera de S , para cualquier par de vértices $u, v \in S$. Equivalentemente, si $I_G[S] = S$.

El primer ejemplo de conjuntos convexos en un grafo son los vértices. Más aún, si S es clique de G , entonces S es convexo en G .

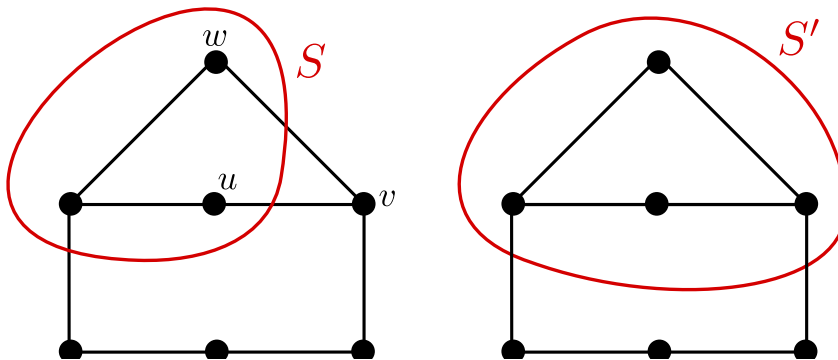


Figura 1.1: S no es convexo, pues el camino uvw es mínimo. En cambio, S' sí es convexo.

Definición 1.17 Sea $p \in \mathbf{N}$. Una partición $\{V_1, \dots, V_p\}$ de $V(G)$ tal que los conjuntos V_1, \dots, V_p son convexos se dirá **p -partición convexa** de G . A la 1-partición convexa $\{V(G)\}$ la llamaremos la **partición convexa trivial**.

Si S es un convexo de G , entonces siempre existe una partición convexa de G que lo contiene, pues basta considerar

$$\mathcal{S} = \{S\} \cup \{\{v\}\}_{v \in V(G) \setminus S}.$$

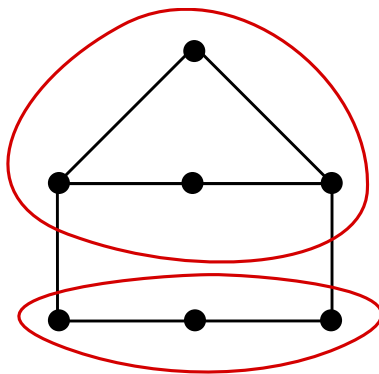


Figura 1.2: Una 2-partición convexa.

Otro ejemplo sencillo de partición convexa surge de considerar la partición de los vértices dada por un emparejamiento, la que consiste en el conjunto de aristas del emparejamiento y los singleton de cada vértice no emparejado. Más aún, cualquier partición en cliques de G es una partición convexa de G .

El problema que se abordará en este trabajo es decidir si un grafo G dado tiene una p -partición convexa, donde p es un número fijo. Para grafos arbitrarios este problema es NP-completo [1].

Como anteriormente mencionamos, los problemas de p -partición convexa y de p -partición en cliques están relacionados, pues los cliques siempre son convexos. Sin embargo, la relación no acaba allí, pues para $p \geq 3$, el problema de p -partición convexa se reduce al de p -partición en

cliques [1]. Aún teniendo tal relación, estos problemas mantienen sus diferencias, siendo la principal que el problema de p -partición convexa no tiene la propiedad de monotonía que p -partición en clique sí tiene: si G tiene una p -partición convexa, con $p < n$, no necesariamente G tiene una $(p + 1)$ -partición convexa.

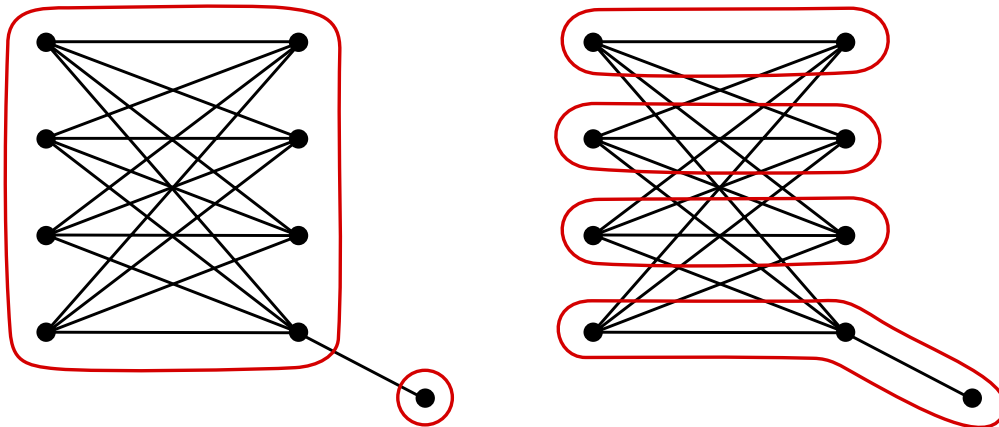


Figura 1.3: En la figura se muestra un ejemplo, presentado en [1], de un grafo con 2 y 4-partición convexa, pero sin 3-partición convexa.

Aun si el problema es el problema es NP-completo, hay casos donde el problema se resuelve fácilmente. Uno de ellos es el de los grafos cordados, en el cual el problema de p -partición convexa es fácil, pues siempre existe una p -partición convexa.

Definición 1.18 G es un grafo cordado si no tiene ciclos inducidos de largo 4 o más.

Teorema 1.19 [1] Si G es un grafo cordado de n vértices, entonces tiene p -partición convexa, $\forall p \in [n]$.

Hay otras familias de grafos en los que el problema se puede resolver en tiempo polinomial, pero mediante técnicas más sofisticadas, como los cografos [1] y los grafos bipartitos [13].

Para atacar el problema de p -partición convexa, basta poder resolverlo en el caso conexo, pues para grafos no conexos, el problema se reduce a sus componentes conexas mediante el siguiente lema:

Lema 1.20 [1] Sea G un grafo con componentes conexas G_1, \dots, G_r . Entonces, G tiene p -partición convexa si y sólo si para cada G_i existe $p_i \leq p$ tal que

1. G_i tiene p_i -partición convexa y
2. $\sum_{i=1}^r p_i \geq p$.

La segunda condición permite que $\sum_{i=1}^r p_i \geq p$, pues la unión de dos convexas en componentes conexas distintas es un conjunto conexo. Así, por cada componente conexas de G , basta encontrar el $p_i \leq p$ máximo tal que G_i tiene p_i -partición convexa.

Lema 1.24 Sean G_1, G_2 grafos y $a = (u, x), b = (v, y) \in V(G_1 \times G_2)$. Entonces

$$d((u, x), (v, y)) = d_{G_1}(u, v) + d_{G_2}(x, y)$$

Demostración:

Sean $(u, w_1, \dots, w_k, v) \in \mathcal{G}_{G_1}(u, v)$ y $(x, z_1, \dots, z_l, y) \in \mathcal{G}_{G_2}(x, y)$. Consideremos el (a, b) -camino dado por seguir primero el camino mínimo en G_2 y después el camino mínimo en G_1 :

$$((u, x), (u, z_1), \dots, (u, y), (w_1, y), \dots, (v, y)).$$

Luego, $d((u, x), (v, y)) \leq d_{G_1}(u, v) + d_{G_2}(x, y)$. Por otra parte, sea P un (a, b) -camino mínimo. Tenemos que P_1 es un (u, v) -paseo y P_2 es un (x, y) -paseo, por lo que $\|P_1\| \geq d_{G_1}(u, v)$ y $\|P_2\| \geq d_{G_2}(x, y)$. Además, tenemos que $\|P_1\| + \|P_2\| = \|P\|$, pues cada arista de P está incluida en solo uno de los paseos P_1, P_2 . Luego,

$$d((u, x), (v, y)) \geq d_{G_1}(u, v) + d_{G_2}(x, y).$$

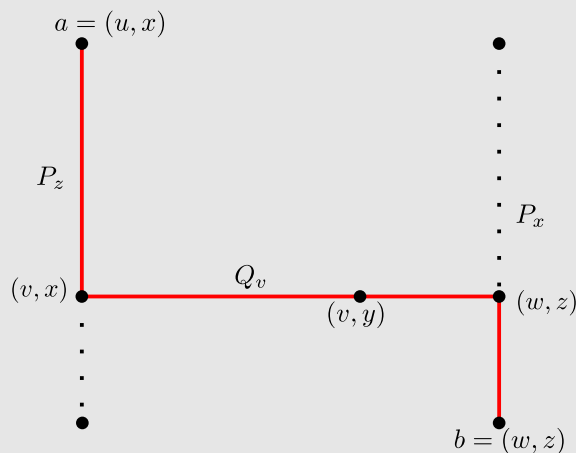
Del lema anterior, tenemos que si $P \in \mathcal{G}((u, x), (v, y))$, entonces P_1 y P_2 son (u, v) y (x, y) -caminos mínimos, respectivamente.

Lema 1.25 Sean $a = (u, x), b = (v, y), c = (w, z)$, vértices de $G_1 \times G_2$. Entonces,

$$b \in I_{G_1 \times G_2}[a, c] \text{ si y sólo si } v \in I_{G_1}[u, w] \wedge y \in I_{G_2}[x, z].$$

Demostración:

(\Leftarrow) Sean $P \in \mathcal{G}_{G_1}(u, w)$ que contiene a v y $Q \in \mathcal{G}_{G_2}(x, z)$. Sean P_x, P_z las copias de P en $G_1 \times G_2$ con segunda coordenada x y z , respectivamente. De igual modo, definimos Q_v como la copia de Q con primera coordenada v . Sea R el (a, c) -camino dado por seguir P_x hasta (v, x) , luego seguir Q_v y finalmente seguir P_z desde (v, z) hasta (w, z) . Tenemos que, por el lema anterior, R es un (a, c) -camino mínimo y además pasa por b .



(\Rightarrow) Sea $P \in \mathcal{G}(a, c)$ que pasa por b . Tenemos que P_1 es un (u, w) -camino mínimo que pasa por v . De igual manera, P_2 es un (x, z) -camino mínimo que pasa por y .

Tenemos que si S es un convexo en $G_1 \times G_2$, entonces el lema anterior implica que S_1, S_2 , sus proyecciones en cada coordenada, son convexos en G_1, G_2 , respectivamente.

Teorema 1.26 *Los conjuntos convexos de $G_1 \times G_2$ son de la forma $S_1 \times S_2$, donde S_1 y S_2 son convexos en G_1 y G_2 , respectivamente.*

Demostración:

- Sean S_1, S_2 convexos de G_1, G_2 , respectivamente. Por el lema anterior, $S = S_1 \times S_2$ es convexo en $G_1 \times G_2$.
- Sea S convexo en $G_1 \times G_2$. Consideremos S_1, S_2 las proyecciones de S a cada coordenada. Claramente, $S \subseteq S_1 \times S_2$. Veamos ahora la otra inclusión. Sea $(u, x) \in S_1 \times S_2$. Entonces, existen $v \in S_1, y \in S_2$ tales que $(u, y), (v, x) \in S$. Sean P_u una copia de un (x, y) -camino mínimo con primera coordenada u y Q_x una copia de un (u, v) -camino mínimo con segunda coordenada x . Entonces, $P_u Q_x \in \mathcal{G}((u, y), (v, x))$ y contiene a (u, x) . Como S es convexo, entonces $(u, x) \in S$. En consecuencia, $S = S_1 \times S_2$.

Lema 1.27 *Sean $v \in V(G_1)$, \mathcal{S} partición convexa de $G_1 \times G_2$. Entonces,*

$$\mathcal{S}_v = \{S_2 : \text{existe } x \in V(G_2) \text{ tal que } (v, x) \in S, S \in \mathcal{S}\}$$

es partición convexa de G_2 . Análogamente, se tiene para la proyección en la primera coordenada.

Demostración:

Como \mathcal{S} es partición convexa de $G_1 \times G_2$, entonces \mathcal{S}_v es partición de $V(G_2)$, pues la primera coordenada está fija. Por otra parte, como las proyecciones de convexos en $G_1 \times G_2$ son convexas, entonces \mathcal{S}_v es partición convexa.

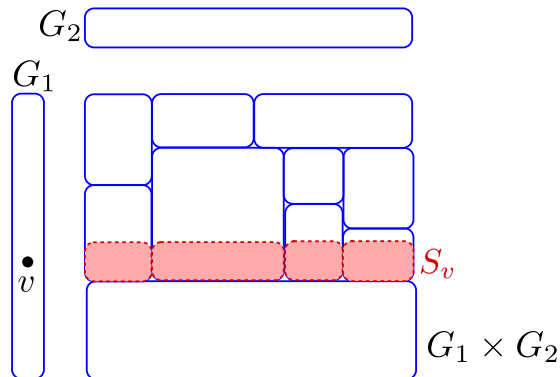


Figura 1.5: Esquema de la partición \mathcal{S}_v definida en el lema 1.27.

Teorema 1.28 *Sean G_1, G_2 grafos tales que G_2 tiene p vértices y contiene una m -partición*

convexa, $\forall m \in [p]$. Entonces $G_1 \times G_2$ contiene una t -partición convexa si y solo si G_1 tiene una q -partición convexa tal que $t \in [q, pq]$.

Demostración:

(\Leftarrow) Sea $\mathcal{T} = (T_1, \dots, T_q)$ una q -partición convexa de G_1 , lo que haremos para particionar $G_1 \times G_2$ será elegir q particiones convexas de G_2 , $\{\mathcal{S}_i\}_{i=1}^q$, y consideraremos la partición

$$\mathfrak{S} = \{T_i \times S : S \in \mathcal{S}_i, i \in [q]\}.$$

Como \mathfrak{S} contiene productos de convexas, entonces es partición convexa. Además, como los tamaños de \mathcal{S}_i se pueden elegir libremente en $[p]$, entonces \mathfrak{S} puede tener cualquier tamaño en $[q, pq]$.

(\Rightarrow) Supongamos que existe t tal que $G_1 \times G_2$ tiene una t -partición convexa, pero donde t no está en ningún intervalo de la forma $[q, pq]$. Como G_1 siempre tiene una 1-partición convexa y una $|V(G_1)|$ -partición convexa, entonces para que se dé lo anterior, t debe estar entre dos de tales intervalos. Es decir, deben existir $q < q'$ tamaños de particiones convexas de G_1 tales que no hay r -particiones convexas con $r \in (q, q')$, los intervalos $[q, pq]$, $[q', pq']$ son disjuntos y $t \in (pq, q)$. Entonces, debe existir $v \in V(G_2)$ tal que $r = |\mathcal{S}_v| > q$, pues si no

$$t = |\mathcal{S}| \leq \sum_{v \in V(G_2)} |\mathcal{S}_v| \leq pq.$$

Por otra parte, $r \leq t < q'$. Es decir, existe una r -partición convexa de G_1 con $r \in (q, q')$
 $\rightarrow \leftarrow$.

Un ejemplo importante de grafos que cumplen la propiedad de tener particiones convexas de todos los tamaños son los grafos cordados [1]. Del teorema anterior, tenemos que podemos resolver el problema de partición convexa para un grafo de la forma $G_1 \times G_2$ cuando G_2 es cordado, mediante $O(|V(G_1)|)$ llamados a un oráculo para partición convexa en G_1 . Esto, pues nos basta calcular cuáles son los valores $q \in [|V(G_1)|]$ tales que G_1 tiene una q -partición convexa, pues entonces las particiones convexas de $G_1 \times G_2$ solo pueden tener tamaños que estén en alguno de los intervalos $[q, pq]$.

Notemos que el resultado anterior, básicamente nos dice que el conjunto de tamaños de las particiones convexas de $G_1 \times G_2$ es una unión de intervalos que dependen de los tamaños de las particiones convexas de G_1 . Entonces, una pregunta relevante que surge de lo anterior es cuánto se puede debilitar las condición sobre las particiones convexas de G_2 para tener un resultado análogo.

1.3.2. Suma

A diferencia de la operación anterior, la suma de grafos no es compatible con las distancias en los grafos a operar. De hecho, la suma de dos grafos siempre tiene diámetro 2.

Definición 1.29 Sean G_1, G_2 grafos. Su **suma**, $G_1 + G_2$, consiste en agregarle todas las

aristas entre $V(G_1)$ y $V(G_2)$ a $G_1 \cup G_2$.

Proposición 1.30 *Si G_1, G_2 son ambos cliques o ninguno es clique, entonces una partición convexa de $G_1 + G_2$ es una partición en cliques o es la trivial.*

Demostración:

Si G_1 y G_2 son cliques, entonces G es un clique y se tiene lo pedido. Si ni G_1 ni G_2 es clique, consideremos S convexo en $G_1 + G_2$. Supongamos que no es un clique. Veremos entonces que $S = V(G)$. Sean $u, v \in S$ tales que $uv \notin E(G)$. Entonces, u, v deben pertenecer al mismo G_i , que supondremos que es G_1 . Luego, como todos los vértices de G_2 están en (u, v) -camino de largo 2, $V(G_2) \subseteq S$. Como G_2 no es clique, el mismo argumento anterior implica que $V(G_1) \subseteq S$ y, por lo tanto, $S = V(G)$.

El caso restante es cuando G_1 es clique y G_2 no lo es. En tal caso, nos aparecerá un nuevo tipo de objeto, que definimos a continuación

Definición 1.31 *Sea $S \subseteq V(G)$. Diremos que S es **localmente convexo** si*

$$\forall u, v \in S, \forall w \in V(G): d(u, v) = 2, uw, vw \in E(G) \Rightarrow w \in S.$$

Es decir, S es cerrado para caminos mínimos de largo a lo más 2. Esto corresponde a un caso particular de la noción general de convexidad local definida en la literatura.

Definición 1.32 [11] *Un conjunto se dice g_k -convexo si es cerrado para caminos mínimos de largo a lo más k .*

Notemos que todo conjunto convexo es localmente convexo. Sin embargo, la recíproca no es cierta, como muestra el siguiente ejemplo:

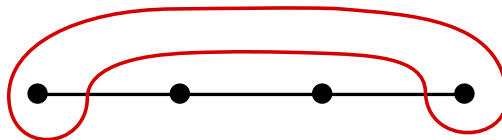


Figura 1.6: Un conjunto localmente convexo que no es convexo.

Proposición 1.33 *Si G_1 es clique, pero G_2 no lo es, entonces una partición convexa de $G_1 + G_2$ que no es la trivial ni es una partición en cliques está compuesta por $V(G_1) \cup S$ y cliques, donde S es localmente convexo en G_2 .*

Demostración:

Primero, veremos que los convexos de $G_1 + G_2$ que no son cliques son de la forma $V(G_1) \cup S$, donde S es localmente convexo en G_2 . Un convexo que no es clique, debe tener dos vértices en $V(G_2)$ que no son adyacentes, por lo que debe contener a $V(G_1)$. Luego, nos basta verificar que si $S \subseteq V(G_2)$, entonces $S' = V(G_1) \cup S$ es convexo en $G_1 + G_2$ si y sólo si S es localmente convexo en G_2 .

(\Rightarrow) Sean $u, v \in S$ tales que $d_{G_2}(u, v) = 2$. Tenemos que como $uv \notin E(G)$ y S' es convexo, entonces

$$\forall w \in V(G_2): uw, vw \in E(G) \Rightarrow w \in S.$$

(\Leftarrow) Solo debemos verificar que $I[u, v] \subseteq S'$ para $u, v \in S'$ no adyacentes. Como G_1 es clique, entonces consideraremos $u, v \in S$. Ahora, como S es localmente convexo en G_2 , tenemos que $I[u, v] \cap V(G_2) \subseteq S$. Como además $V_1 \subseteq S'$, entonces $I[u, v] \subseteq S'$.

Si S' está en una partición convexas, entonces el resto de los conjuntos que la componen deben ser cliques contenidos en G_2 , pues no pueden contener vértices de G_1 .

Así, al estudiar las particiones convexas en la suma, se nos presenta un nuevo problema a resolver: determinar si un grafo G tiene una partición tal que uno de sus elementos es localmente convexo y el resto son cliques.

Definición 1.34 Sea $\{S_1, \dots, S_p\}$ una partición de G tal que S_1 es localmente convexo en G y S_2, \dots, S_p son cliques. Entonces a $\{S_1, \dots, S_p\}$ la llamaremos **p -partición localmente convexa** de G .

Una observación importante es que una p -partición localmente convexa de un grafo no es necesariamente una p -partición en conjuntos localmente convexas, pero usaremos ese nombre de todos modos por comodidad.

De la **proposición 1.33**, tenemos que si G_1 es clique y G_2 tiene una p -partición localmente convexa, entonces $G_1 + G_2$ tiene una partición localmente convexa.

La primera dificultad al estudiar los conjuntos localmente convexas, aparece con la combinatoria que introduce el siguiente resultado

Proposición 1.35 Sea G un grafo y sean S_1, S_2 conjuntos localmente convexas en G . Si $d(S_1, S_2) > 2$, entonces $S_1 \cup S_2$ es localmente convexo en G .

Demostración:

Sea u, v un par de vértices de $S_1 \cup S_2$ tales que $d(u, v) = 2$. Como $d(S_1, S_2) > 2$, entonces ambos vértices deben estar en el mismo S_i , que supondremos es S_1 . Luego, como S_1 es localmente convexo

$$uw, vw \in E(G) \Rightarrow w \in S_1 \subseteq S_1 \cup S_2.$$

Por lo tanto, $S_1 \cup S_2$ es localmente convexo.

Del resultado anterior, tenemos que dada una familia de conjuntos convexas que están suficientemente lejos, la unión de cualquier subconjunto de esta familia será un conjunto localmente convexo. Así, la cantidad de conjuntos localmente convexas puede ser mucho mayor que la de los conjuntos convexas.

Computacionalmente, el problema de encontrar una partición convexa también es difícil.

Teorema 1.36 *El problema de determinar si un grafo tiene una p -partición localmente convexa es NP -completo para $p \geq 3$.*

Demostración:

Dada una partición, se puede verificar en tiempo polinomial que es localmente convexa.

Para ver que el problema es NP -completo, usaremos la construcción dada en [1] para demostrar que el problema de p -partición convexa es NP -completo. En ésta, dado un grafo G se construye G' agregándole a G dos vértices u, v tales que $N_{G'}(u) = N_{G'}(v) = V(G)$. Además, demuestran que los conjuntos convexos propios de G' son cliques. Finalmente, concluyen al ver que G tiene una p -partición en cliques si y solo si G' tiene una p -partición en convexa, que por lo anterior es una partición en cliques.

Como G' es de diámetro 2, entonces los conjuntos localmente convexos son los conjuntos convexos. Por otra parte, como los convexos propios de G' son cliques, entonces las particiones localmente convexas de G' son las particiones en cliques. Por lo tanto, la reducción anterior también prueba que el problema de p -partición localmente convexa es NP -completo para $p \geq 3$.

Capítulo 2

Familia \mathcal{T}^*

Como dijimos en un principio, la idea de esta familia es recuperar algunas propiedades de los árboles. Veremos que para los grafos de diámetro 3 o más, su métrica es básicamente la de un árbol. Por otra parte, para el problema de las particiones convexas, nos interesará cortarlo por cliques, encontrar particiones en los subgrafos que quedan de esto y luego combinarlas mediante programación dinámica.

Esta familia queda definida de manera recursiva, reemplazando sus vértices por grafos de ésta misma.

Definición 2.1 Sean G, H grafos y $v \in V(G)$. Definimos la operación de reemplazo $G_{v \rightarrow H}$ por

- $V(G_{v \rightarrow H}) = V(G) \setminus \{v\} \cup V(H)$
- $E(G_{v \rightarrow H}) = E(G) \setminus \{uv \in E(G) : u \in V(G)\} \cup E(H) \cup \{uw : uv \in E(G), w \in V(H)\}$

Definición 2.2 Definiremos la siguiente familia de grafos de manera recursiva: los grafos en \mathcal{T}_k ($k \geq 1$) se obtienen de un bosque F reemplazando cada vértice por un grafo en \mathcal{T}_{k-1} , donde \mathcal{T}_0 son los bosques.

En la definición anterior, siempre se puede considerar F como el bosque de un solo vértice, por lo que tenemos que $\mathcal{T}_{k-1} \subseteq \mathcal{T}_k$.

Definición 2.3 Si $G \in \mathcal{T}_k$ se genera a partir de un bosque F , reemplazando sus vértices V_1, \dots, V_r por los grafos $G_1, \dots, G_r \in \mathcal{T}_{k-1}$, diremos que F es un bosque subyacente de G y que $V(G_1), \dots, V(G_r)$ es su partición asociada. Usualmente, nos referiremos a los conjuntos $V(G_1), \dots, V(G_r)$ por los vértices en F , V_1, \dots, V_r .

Notemos que como los conjuntos V_1, \dots, V_r se construyeron mediante reemplazos, entonces éstos son módulos de G . De este modo, si V_i, V_j son vértices adyacentes de F , entonces el grafo $G[V_i \cup V_j]$ corresponde a la suma $G_i + G_j$.

Por otra parte, tenemos que se pueden unir los grafos asociados a dos hojas con el mismo

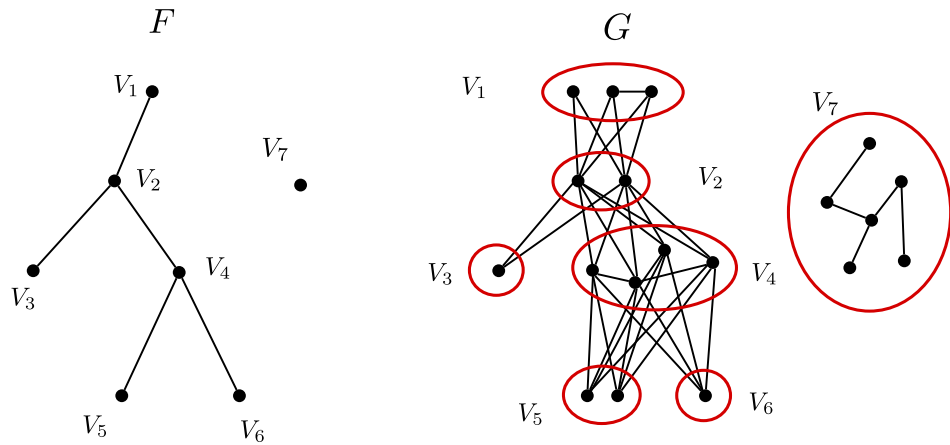


Figura 2.1: Un grafo G en \mathcal{T}_1 junto a su bosque subyacente F .

padre en el bosque subyacente. Luego, podemos suponer sin pérdida de generalidad, que cada vértice interno de los bosques subyacentes tiene a lo más una hoja adyacente a él.

Definición 2.4 *A un bosque subyacente que tiene a lo más una hoja adyacente a cada vértice interno lo llamaremos **bosque subyacente minimal**.*

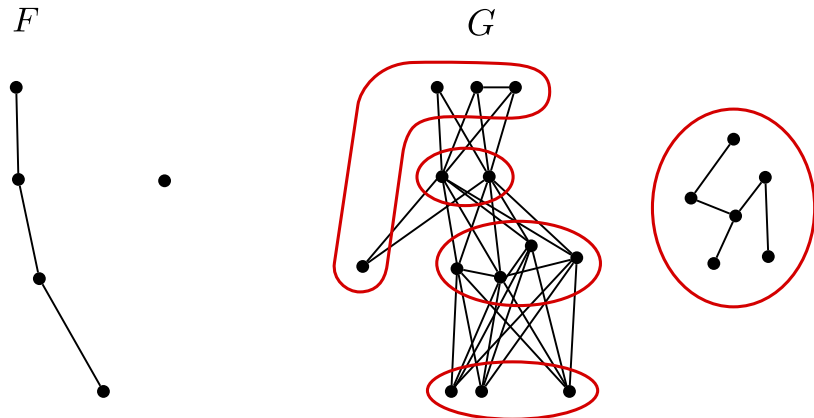


Figura 2.2: El resultado de unir las hojas asociadas al mismo padre en el grafo de la figura anterior.

Definición 2.5 *Definimos la familia $\mathcal{T}^* = \bigcup_{k \in \mathbb{N}} \mathcal{T}_k$.*

Como muchas de las demostraciones o algoritmos en \mathcal{T}^* serán recursivos, nos interesará saber cuántos niveles de reemplazo se ocuparon para construir un grafo en \mathcal{T}^* , lo que motiva la siguiente definición:

Definición 2.6 *Para $G \in \mathcal{T}^*$, definiremos su **nivel**, $\text{lev}(G)$, como el mínimo k tal que $G \in \mathcal{T}_k$.*

Notemos que dado $G \in \mathcal{T}^*$, entonces el solo hecho de tener el bosque subyacente T junto a su partición asociada V_1, \dots, V_r no basta como certificado de pertenencia de G en \mathcal{T}^* ,

pues bastaría reemplazar grafos cualesquiera en T . Para ello, necesitamos tener los bosques subyacentes y sus particiones asociadas de forma recursiva. Entonces, la descripción completa de $G \in \mathcal{T}^*$ puede ocupar espacio $O(|V(G)|^2)$, pues veremos que el nivel de reemplazos para construir G puede ser lineal.

2.1. Propiedades básicas

Dado que para estudiar el nivel de un grafo en \mathcal{T}^* no nos interesan los reemplazos triviales, es decir, usar un solo vértice como bosque subyacente o reemplazar un vértice por otro, pues estos dos casos no aumentan el nivel del grafo construido. Entonces, tendremos que la cantidad de niveles que se puede subir en la jerarquía de \mathcal{T}^* es limitada, dada una cantidad fija de vértices.

Lema 2.7 *Para $G \in \mathcal{T}^*$, $\text{lev}(G) \leq |V(G)|$.*

Demostración:

Lo probaremos inductivamente para los grafos de \mathcal{T}_k . Claramente se tiene lo pedido para $G \in \mathcal{T}_0$. Sea ahora $G \in \mathcal{T}_k \setminus \mathcal{T}_{k-1}$, $k \geq 1$. Entonces, existen $G_1, \dots, G_r \in \mathcal{T}_{k-1}$, $r > 1$, y F un bosque tales que G se obtiene a partir de F reemplazando sus vértices por los grafos G_1, \dots, G_r . Además, como $G \in \mathcal{T}_k \setminus \mathcal{T}_{k-1}$, entonces existe $j \in [r]$ tal que $\text{lev}(G_j) = k - 1$. Luego,

$$|V(G)| = \sum_{i=1}^r |V(G_i)| \geq \sum_{i \in [r] \setminus \{j\}} |V(G_i)| + \text{lev}(G_j) \geq 1 + k - 1 = k.$$

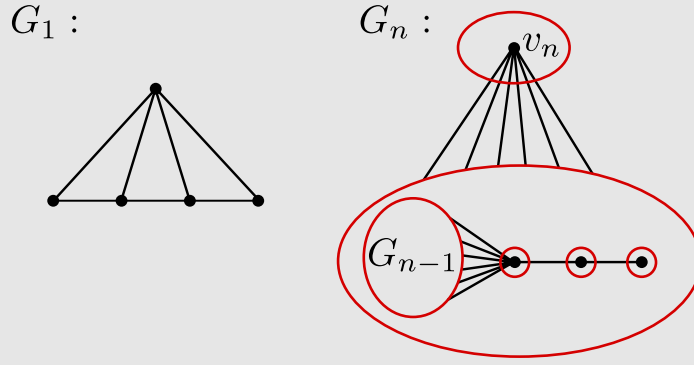
Además, la cota anterior es ajustada salvo por una constante, como veremos a continuación. Para construir este ejemplo, usaremos el hecho de que los árboles subyacentes de un grafo de diámetro 3 o más están dados por la métrica del grafo, lo que implica unicidad si estos son minimales. Esto último será revisado en la **sección 2.2**.

Proposición 2.8 *Existe una familia de grafos $\{G_n\}_{n \in \mathbb{N}}$ en \mathcal{T}^* tales que*

$$\frac{\text{lev}(G_n)}{|V(G_n)|} = \frac{1}{2} - o(1).$$

Demostración:

Consideremos la construcción inductiva, dada por el siguiente esquema:



Es decir, G_1 se obtiene de sumarle un vértice a un P_4 y G_n se obtiene de reemplazar un extremo de P_4 por G_{n-1} y luego sumarle un vértice a tal grafo.

Primero, veamos que $\text{lev}(G_n) = \text{lev}(G_{n-1}) + 2, \forall n \geq 1$. Llamemos $H_n = G_n \setminus \{v_n\}$. Tenemos que \overline{G}_n tiene solo dos componentes conexas: v_n y H_n . Entonces G_n solo tiene un árbol subyacente posible, la arista en la que uno de sus extremos es v_n y el otro se reemplaza por H_n . Luego, $\text{lev}(G_n) = \text{lev}(H_n) + 1$. Por otra parte, como H_n tiene diámetro 3, entonces su árbol subyacente también es único y, por lo tanto, $\text{lev}(H_n) = \text{lev}(G_{n-1}) + 1$. Como $\text{lev}(G_1) = 1$, entonces $\text{lev}(G_n) = 2n - 1, \forall n \in \mathbf{N}$.

Por otra parte, $|V(G_n)| = |V(G_{n-1})| + 4$ y $|V(G_1)| = 5$, por lo que $|V(G_n)| = 4n + 1, \forall n \in \mathbf{N}$.

Por lo tanto,

$$\frac{\text{lev}(G)}{|V(G)|} = \frac{2n - 1}{4n + 1} = \frac{1}{2} - \frac{1}{8n + 2}.$$

Proposición 2.9 \mathcal{T}^* es cerrada para unión y suma.

Demostración:

Sean $G_1, G_2 \in \mathcal{T}^*$.

- Sea T_1 un bosque subyacente de G_1 y sea $\mathcal{V}^1 = (V_1^1, \dots, V_q^1)$ su partición asociada. Análogamente, consideremos T_2 y $\mathcal{V}^2 = (V_1^2, \dots, V_r^2)$. Entonces, $T_1 \cup T_2$ es bosque subyacente de $G_1 \cup G_2$ con partición asociada $\mathcal{V}^1 \cup \mathcal{V}^2$, por lo que $G_1 \cup G_2 \in \mathcal{T}^*$.
- Notemos que $G_1 + G_2$ se puede construir a partir de una arista, reemplazando sus extremos por G_1 y G_2 . Luego, como $G_1, G_2 \in \mathcal{T}^*$, entonces $G_1 + G_2 \in \mathcal{T}^*$.

Proposición 2.10 Sean $G_1, G_2 \in \mathcal{T}^*$. Entonces, $\text{lev}(G_1 \cup G_2) = \max\{\text{lev}(G_1), \text{lev}(G_2)\}$.

Demostración:

La construcción realizada en el **lema 2.9**, nos dice que

$$\text{lev}(G_1 \cup G_2) \leq \max\{\text{lev}(G_1), \text{lev}(G_2)\}.$$

Por otro lado, como G_1 y G_2 son subgrafos inducidos de $G_1 \cup G_2$, por **proposición 2.19**,

entonces

$$\text{lev}(G_1 \cup G_2) \geq \text{lev}(G_1), \text{lev}(G_2).$$

Del mismo modo, nos gustaría obtener una propiedad análoga para la suma. De la construcción de la **proposición 2.9**, la tentación es a decir que

$$\text{lev}(G_1 + G_2) = 1 + \text{máx}\{\text{lev}(G_1), \text{lev}(G_2)\}.$$

Sin embargo, si alguno de esos grafos se construyó como una suma, entonces podríamos intercambiar el orden de las operaciones para construir el grafo total usando menos reemplazos. Sean $G_1, G_2, G_3 \in \mathcal{T}^*$ tales que $\text{lev}(G_1) = 1$, $\text{lev}(G_2) = 1$, $\text{lev}(G_3) = 40$. Usando la fórmula anterior, tendríamos que

$$\begin{aligned} \text{lev}(G_1 + (G_2 + G_3)) &= 1 + \text{máx}\{\text{lev}(G_1), 1 + \text{máx}\{\text{lev}(G_2), \text{lev}(G_3)\}\} \\ &= 1 + \text{máx}\{1, 1 + \text{máx}\{1, 40\}\} = 42 \end{aligned}$$

Sin embargo, 42 no es la respuesta, porque al cambiar el orden en el que se suma, tenemos

$$\begin{aligned} \text{lev}((G_1 + G_2) + G_3) &= 1 + \text{máx}\{1 + \text{máx}\{\text{lev}(G_1), \text{lev}(G_2)\}, \text{lev}(G_3)\} \\ &= 1 + \text{máx}\{1 + \text{máx}\{1, 1\}, 40\} = 41 \end{aligned}$$

El problema de cómo considerar la manera óptima para sumar grafos en \mathcal{T}^* volverá a aparecer en la **sección 2.2**.

Proposición 2.11 *Los cliques de $G \in \mathcal{T}_k$ tienen tamaño a lo más 2^{k+1} y esa cota se alcanza.*

Demostración:

Claramente los cliques de tamaño máximo en \mathcal{T}_0 son aristas.

Sea $G \in \mathcal{T}_k$, $k \geq 1$ y F un bosque subyacente. Un clique máximo debe estar conformado por cliques en vértices vecinos de F . Como estos son cliques de grafos en \mathcal{T}_{k-1} , entonces tienen tamaño a lo más 2^k . Luego, un clique máximo en G puede tener tamaño a lo más 2^{k+1} .

Para ver que la cota se alcanza, basta ver que $K_{2^{k+1}} \in \mathcal{T}_k$, mediante la siguiente construcción inductiva: sea P_2 el camino de dos vértices. Reemplazando cada uno de sus vértices por K_{2^k} , se obtiene $K_{2^{k+1}}$. Como $K_{2^k} \in \mathcal{T}_{k-1}$, entonces $K_{2^{k+1}} \in \mathcal{T}_k$. El caso base, $K_2 \in \mathcal{T}_0$, se tiene pues K_2 es un bosque.

En lo que sigue, como los argumentos se pueden realizar por cada componente conexa de $G \in \mathcal{T}^*$, supondremos que G es conexo. Así, hablaremos del árbol subyacente de G en vez de su bosque subyacente.

El siguiente resultado relaciona las distancias de $G \in \mathcal{T}^*$, en particular sus caminos mínimos, con las de sus árboles subyacentes T .

Proposición 2.12 Sean $G \in \mathcal{T}^*$, T su árbol subyacente y V_1, \dots, V_r su partición asociada. Sean $u, v \in V(G)$ tales que $u \in V_i, v \in V_j$ con V_i, V_j no adyacentes, $i \neq j$. Entonces, el conjunto de (u, v) -caminos mínimos es

$$\mathcal{G}(u, v) = \{u\} \times V_{i_1} \times \dots \times V_{i_p} \times \{v\},$$

donde $V_i V_{i_1} \dots V_{i_p} V_j$ es el (V_i, V_j) -camino en T .

Demostración:

Directo de las definiciones.

2.2. Reconocimiento

Dado G un grafo, queremos diseñar un algoritmo de reconocimiento para $G \in \mathcal{T}^*$, es decir, queremos un algoritmo que haga lo siguiente:

1. Si $G \in \mathcal{T}^*$, nos entregue $\text{lev}(G)$, su bosque subyacente y su partición asociada.
2. Si $G \notin \mathcal{T}^*$, lo reporte.

Como \mathcal{T}_0 corresponde a los bosques, lo que se puede determinar en tiempo $O(|V(G)| + |E(G)|)$ [8], entonces queda estudiar el caso $G \in \mathcal{T}^* \setminus \mathcal{T}_0$. Además, si G no es conexo y sus componentes conexas son G_1, \dots, G_r , entonces por la **proposición 2.10**

$$\text{lev}(G) = \text{máx}\{\text{lev}(G_i) : i \in [r]\},$$

su bosque subyacente F se obtiene uniendo los árboles subyacentes de cada G_i, T_i , y la partición asociada a F es la unión de las particiones asociadas a los T_i . Luego, podemos reducirnos a trabajar con G conexo.

Procederemos de modo distinto según el valor de $\text{diam}(G)$.

Algoritmo 1: REC(G)

```

si  $G$  es un bosque entonces
  | devolver  $G \in \mathcal{T}_0$ 
si  $\text{diam}(G) \geq 3$  entonces
  | devolver REC3( $G$ )
en otro caso
  | devolver REC2( $G$ )

```

2.2.1. $\text{diam}(G) \geq 3$

En este caso, veremos que, para $G \in \mathcal{T}^*$ su árbol minimal T es único, pues queda determinado por la métrica de G . Sean $u, v \in V(G)$ tales que $d(u, v) \geq 3$. Entonces, $u \in V_i, v \in V_j, i \neq j$. Sean V_{i_1}, \dots, V_{i_p} los vértices internos del (V_i, V_j) -camino en T . Por la **proposición 2.12**, los (u, v) -caminos mínimos quedan dados por el conjunto

$$\{u\} \times V_{i_1} \times \dots \times V_{i_p} \times \{v\}.$$

Entonces, tenemos que para $l \in [p]$

$$V_{i_l} = \{w \in I[u, v] : d(u, w) = l\}.$$

Además, G contiene todas las aristas entre V_{i_l} y $V_{i_{l+1}}$.

La siguiente observación asegura que, con lo anterior, siempre podemos reconocer todos los nodos internos de T .

Observación Sea T un árbol tal que $\text{diam}(T) \geq 3$. Entonces, todos sus vértices pertenecen a un camino de largo al menos 3.

Demostración:

Primero lo probaremos para los vértices internos de T . Supongamos que existe v tal que todos los caminos que pasan por él tienen largo a lo más 2. Entonces, todos los vértices de T están conectados a v y, luego, $\text{diam}(T) \leq 2$.

Ahora, si v es una hoja, entonces está conectado a un vértice interno de T , u . Sean P, Q caminos con extremo en u tales que PQ tiene largo al menos 3. Entonces, alguno de ellos tiene largo al menos 2, digamos P , y, por lo tanto, vP tiene largo al menos 3.

Queda entonces determinar los vértices que componen las hojas de T , para lo cual basta particionar los vértices todavía no asignados, según cual V_{i_l} tienen como vecino.

Así, el algoritmo REC3 realiza lo pedido para G en este caso.

Algoritmo 2: REC3(G)

```

 $r \leftarrow 0$ 
para  $u, v \in V(G)$ ,  $d(u, v) \geq 3$  hacer
  para  $i \in [d(u, v) - 1]$  hacer
    si existe  $w \in I[u, v]$  con  $d(u, w) = i$  en algún  $V_j$  entonces
      si  $V_j \neq \{s \in I[u, v] : d(u, s) = i\}$  entonces
        devolver  $G \notin \mathcal{T}^*$ 
      en otro caso
         $r \leftarrow r + 1$ 
         $V_r \leftarrow \emptyset$ 
        para  $s \in I[u, v]$ ,  $d(u, s) = i$  hacer
           $V_r \leftarrow V_r \cup \{s\}$ 
  para  $i \in [r]$  hacer
    si  $N(V_i) \setminus \bigcup_{j=1}^{r-1} V_j \neq \emptyset$  entonces
       $V_{r+1} \leftarrow N(V_i) \setminus \bigcup_{j=1}^{r-1} V_j$ 
       $r \leftarrow r + 1$ 
  para  $i \neq j$ ,  $i, j \in [r]$  hacer
    si  $V_i \cap V_j \neq \emptyset$  entonces
      devolver  $G \notin \mathcal{T}^*$ 
    Sean  $u \in V_i, v \in V_j$ 
    si  $uv \in E(G)$  entonces
      si No están todas las aristas entre  $V_i$  y  $V_j$  entonces
        devolver  $G \notin \mathcal{T}^*$ 
        Agrega  $V_i V_j$  a  $T$ 
      en otro caso
        si Existe alguna arista entre  $V_i$  y  $V_j$  entonces
          devolver  $G \notin \mathcal{T}^*$ 
  si  $T$  no es un árbol entonces
    devolver  $G \notin \mathcal{T}^*$ 
  para  $i \in [r]$  hacer
     $G_i \leftarrow G[V_i]$ 
    REC( $G_i$ )
    si  $G_i \notin \mathcal{T}^*$  entonces
      devolver  $G \notin \mathcal{T}^*$ 
     $l_i \leftarrow \text{lev}(G_i)$ 
  devolver  $T, \{V_i\}_{i=1}^r, \max\{l_i\}_{i=1}^r + 1$ 

```

Tenemos que cálculo de la matriz de distancias se puede hacer en $O(|V(G)|^3)$. Una vez hecho esto, se puede chequear si un vértice $w \in I[u, v]$ es tiempo $O(1)$, pues basta saber si $d(u, v) = d(u, w) + d(w, v)$.

Entonces, REC3 toma tiempo $O(|V(G)|^3 + \sum_{i=1}^r \text{REC}(G_i))$.

Usando el algoritmo anterior, podemos ver que el grafo DOMINO no está en \mathcal{T}^* .

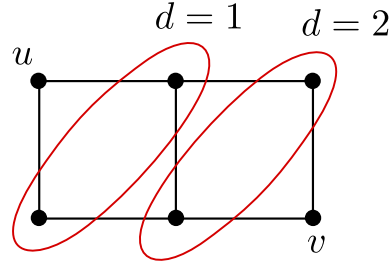


Figura 2.3: En la figura, se muestran el grafo DOMINO, donde se marcan los vértices de $I[u, v]$ según su distancia a u . Como estos conjuntos no tienen todas las aristas entre sí, DOMINO $\notin \mathcal{T}^*$.

2.2.2. $\text{diam}(G) \leq 2$

En este caso, veremos que no hay una forma estándar de representar G , pues no habrá una única forma de elegir la partición asociada al árbol subyacente. Sean $\hat{V}_1, \dots, \hat{V}_r$ las componentes conexas de \overline{G} y sea $\hat{G}_i = G[\hat{V}_i]$, $\forall i \in [r]$.

Proposición 2.13 *Sea $G \in \mathcal{T}^* \setminus \mathcal{T}_0$ con $\text{diam}(G) \leq 2$. Entonces, los únicos árboles subyacentes minimales son las aristas y sus vértices son uniones de componentes conexas de \overline{G} .*

Demostración:

Como $\text{diam}(G) \leq 2$, entonces un árbol subyacente tiene diámetro a lo más 2. Luego, un árbol subyacente minimal debe ser una arista. Luego, los vértices de G se particionan en $\{V_1, V_2\}$ de modo que $G[V_1], G[V_2] \in \mathcal{T}^*$ y que $\{uv : u \in V_1, v \in V_2\} \subseteq E(G)$. Por lo anterior, una componente conexa de \overline{G} debe estar contenida en V_1 o en V_2 , pues éstos son conjuntos disconexos en \overline{G} .

Notemos además que $G_1 = G[V_1], G_2 = G[V_2]$ están en \mathcal{T}^* y que entonces $G = G_1 + G_2$. Si alguno de los G_i contiene a más de una componente \hat{V}_i , entonces su diámetro es a lo más dos y podemos repetir el argumento anterior. Por lo tanto, G resulta de sumar los grafos \hat{G}_i , $i \in [r]$. Sin embargo, como vimos en la **sección 2.1**, el orden en los que se suman los grafos importa cuando se quiere determinar $\text{lev}(G)$.

Además, tenemos que si $G \notin \mathcal{T}_0$, entonces la cantidad de componentes \hat{V}_i es $r > 1$, pues de otro modo, no hay forma de particionar los vértices en dos conjuntos para reemplazar en los extremos de la arista subyacente. Esto, nos da el siguiente criterio para decidir si un grafo de diámetro 2 no está en \mathcal{T}^* .

Proposición 2.14 *Si G tiene diámetro 2 y \overline{G} es conexo, entonces $G \notin \mathcal{T}^*$.*

Demostración:

Notemos que si G tiene diámetro 2 y es un árbol, entonces debe ser una estrella, por lo que \overline{G} no puede ser conexo. Luego, $G \notin \mathcal{T}_0$.

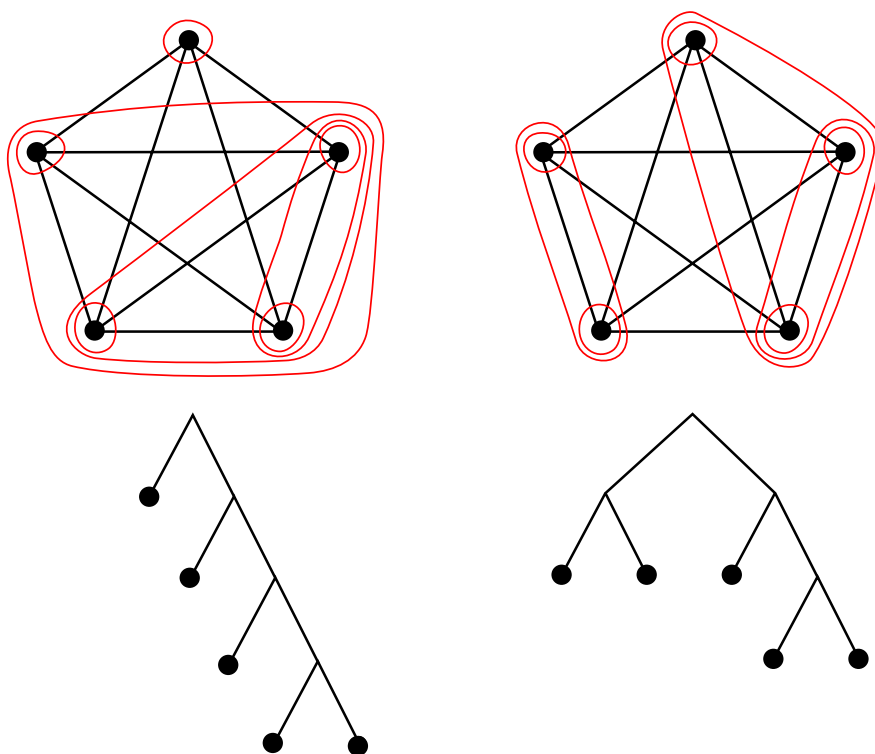


Figura 2.4: Dos formas de construir K_5 en \mathcal{T}^* , junto con los árboles binarios que las representan.

Supongamos ahora que $G \in \mathcal{T}^* \setminus \mathcal{T}_0$. En tal caso, como G tiene diámetro 2, entonces \overline{G} debe tener más de una componente conexas, por el argumento anterior, lo que contradice que \overline{G} es conexo.

Ahora, dados los niveles de \hat{G}_i , nos interesa encontrar la construcción óptima de G , es decir, la que nos permite calcular $\text{lev}(G)$. Para ello, consideraremos a cada componente \hat{G}_i como una hoja de un árbol binario A , sobre el que nos interesa minimizar la siguiente función

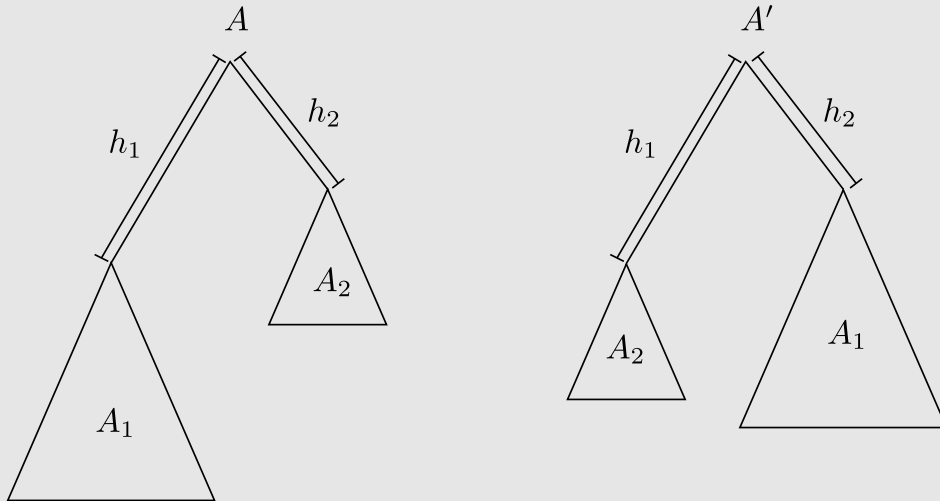
$$f(A) = \max_{i \in [l]} h_A(\hat{G}_i) + \text{lev}(\hat{G}_i),$$

donde $h_A(\hat{G}_i)$ es la profundidad de \hat{G}_i en el árbol A . De igual forma, queda definida la profundidad de un subárbol de A como la profundidad de su raíz. Veremos que podemos encontrar un árbol que minimiza f de manera glotona, para lo cual usaremos el siguiente lema:

Lema 2.15 *Existe un árbol óptimo A tal que para A_1, A_2 subárboles de A , con profundidades h_1, h_2 , respectivamente, si $h_1 > h_2$, entonces $f(A_1) \leq f(A_2)$.*

Demostración:

Sea A árbol óptimo y supongamos que A_1, A_2 son subárboles de A tales que $f(A_1) > f(A_2)$ y sus profundidades satisfacen $h_1 > h_2$. Consideremos ahora A' el árbol obtenido a partir de A intercambiando A_1 y A_2 .



Entonces, los únicos términos que cambian para el cálculo de $f(A')$ son los asociados a tales subárboles.

A	A'
$h_1 + f(T_1)$	$h_1 + f(T_2)$
$h_2 + f(T_2)$	$h_2 + f(T_1)$

Como los nuevos términos en A' son menores que $h_1 + f(A_1)$ y el resto no cambia, entonces $f(A') \leq f(A)$. Repitiendo el argumento en los pares que no cumplen la condición, obtenemos un árbol óptimo que cumple lo pedido.

Del lema anterior, tenemos que podemos construir un árbol óptimo con el siguiente algoritmo glotón, el que es análogo a la construcción del árbol de Huffman [8]:

Algoritmo 3: BINARIOGLOTON(w_1, \dots, w_l)

$Q \leftarrow \emptyset$
para $i \in [l]$ **hacer**
 $T \leftarrow$ hoja i con peso w_i
 $Q \leftarrow Q \cup \{T\}$
mientras $|Q| > 1$ **hacer**
 Sean T_1, T_2 los dos elementos más livianos de Q
 $Q \leftarrow Q \setminus \{T_1, T_2\}$
 $T \leftarrow$ árbol con hijos T_1, T_2 y peso $\max(w(T_1), w(T_2)) + 1$
 $Q \leftarrow Q \cup \{T\}$
devolver T , el único elemento de Q , con su peso.

Usando una cola de prioridad para las operaciones de Q , tenemos que BINARIOGLOTON toma tiempo $O(l \log l)$ [8].

Así, el algoritmo de reconocimiento se ve como sigue en este caso

Algoritmo 4: REC2

$\hat{V}_1, \dots, \hat{V}_r \leftarrow$ componentes conexas de \overline{G}
si $r = 1$ **entonces**
 | **devolver** $G \notin \mathcal{T}^*$
para $i \in [l]$ **hacer**
 | $\hat{G}_i \leftarrow G[\hat{V}_i]$
 | $\text{REC}(\hat{G}_i)$
 | **si** $\hat{G}_i \notin \mathcal{T}^*$ **entonces**
 | | **devolver** $G \notin \mathcal{T}^*$
 | $l_i \leftarrow \text{lev}(\hat{G}_i)$
 $A \leftarrow \text{BINARIOGLOTON}(l_1, \dots, l_r)$
 $V_1 \leftarrow \bigcup_{i \text{ está en la derecha de } A} V(\hat{G}_i)$
 $V_2 \leftarrow \bigcup_{i \text{ está en la izquierda de } A} V(\hat{G}_i)$
devolver $\{V_1 V_2\}, \{V_1, V_2\}, w(A)$

Tenemos que las componentes conexas de \overline{G} se pueden encontrar en tiempo $O(|V(\overline{G})| + |E(\overline{G})|) = O(|V(G)| + |V(G)|^2 - |E(G)|) = O(|V(G)|^2)$, por lo que REC2 toma tiempo

$$\begin{aligned}
\text{REC2}(G) &= O\left(|V(G)|^2 + l \log l + \sum_{i=1}^l \text{REC}(\hat{G}_i)\right) \\
&= O\left(|V(G)|^2 + \sum_{i=1}^l \text{REC}(\hat{G}_i)\right)
\end{aligned}$$

2.2.3. Complejidad de REC

Lema 2.16 REC toma tiempo $O((\text{lev}(G) + 1)|V(G)|^3)$ para $G \in \mathcal{T}^*$.

Demostración:

Lo demostraremos inductivamente en $\text{lev}(G)$. Para $G \in \mathcal{T}_0$, tenemos que REC toma tiempo $O(|V(G)| + |E(G)|) = O(|V(G)|^2) = O(|V(G)|^3)$. Ahora, sea G tal que $\text{lev}(G) \geq 1$. Independiente de $\text{diam}(G)$, tenemos que REC toma tiempo

$$\text{REC}(G) = O(|V(G)|^3 + \sum_{i=1}^l \text{REC}(G_i)),$$

donde los G_i satisfacen que $\sum_{i=1}^l |V(G_i)| = |V(G)|$. Por hipótesis inductiva,

$$\begin{aligned}
\text{REC}(G) &= O(|V(G)|^3 + \sum_{i=1}^l O((\text{lev}(G_i) + 1)|V(G_i)|^3)) \\
&= O\left(|V(G)|^3 + \sum_{i=1}^l \text{lev}(G)|V(G_i)|^3\right) \\
&= O\left(|V(G)|^3 + \text{lev}(G)\left(\sum_{i=1}^l |V(G_i)|\right)^3\right) \\
&= O(|V(G)|^3 + \text{lev}(G)|V(G)|^3) \\
&= O((\text{lev}(G) + 1)|V(G)|^3)
\end{aligned}$$

Teorema 2.17 *REC toma tiempo $O(|V(G)|^4)$ para $G \in \mathcal{T}^*$.*

Demostración:

Es directo de los lemas 2.7, 2.16.

2.3. Estructura

En esta sección, se estudiará la pregunta de cómo se relaciona \mathcal{T}^* con otras familias conocidas de grafos. Además, tratará de acercarse a alguna caracterización de \mathcal{T}^* .

2.3.1. Subgrafos prohibidos

Proposición 2.18 *La familia \mathcal{T}^* es cerrada para subgrafo inducido.*

Demostración:

Sean $G \in \mathcal{T}_k$, T árbol subyacente de G y V_1, \dots, V_r los vértices de T . Sean H subgrafo inducido de G y F el bosque inducido en T por los V_i tales que $V_i \cap V(H) \neq \emptyset$. Definamos los grafos

$$H_i = G[V_i \cap V(H)], \quad V_i \cap V(H) \neq \emptyset.$$

Entonces, H se construye a partir de F reemplazando sus vértices por los grafos H_i . Inductivamente, como H_i es subgrafo inducido de $G[V_i] \in \mathcal{T}_{k-1}$, entonces $H_i \in \mathcal{T}_{k-1}$. Así, $H \in \mathcal{T}_k$.

El caso base, \mathcal{T}_0 , claramente cumple la propiedad.

Tenemos además que si $H \sqsubseteq G$, entonces la construcción presentada en la demostración

anterior acota el nivel de H en función del de G .

Proposición 2.19 Si $H \sqsubseteq G$, entonces $\text{lev}(H) \leq \text{lev}(G)$.

Notemos además que la demostración de la **proposición 2.18** funciona para cualquier familia de grafos definida mediante reemplazos que satisfaga las siguientes dos condiciones:

1. Los grafos subyacentes sean cerrados para subgrafo inducido.
2. La familia base sea cerrada para subgrafo inducido.

Como \mathcal{T}^* resulta ser cerrada para subgrafo inducido, entonces existe una familia de subgrafos prohibidos que la caracteriza. En lo que sigue, trataremos de acercarnos a determinar esta familia.

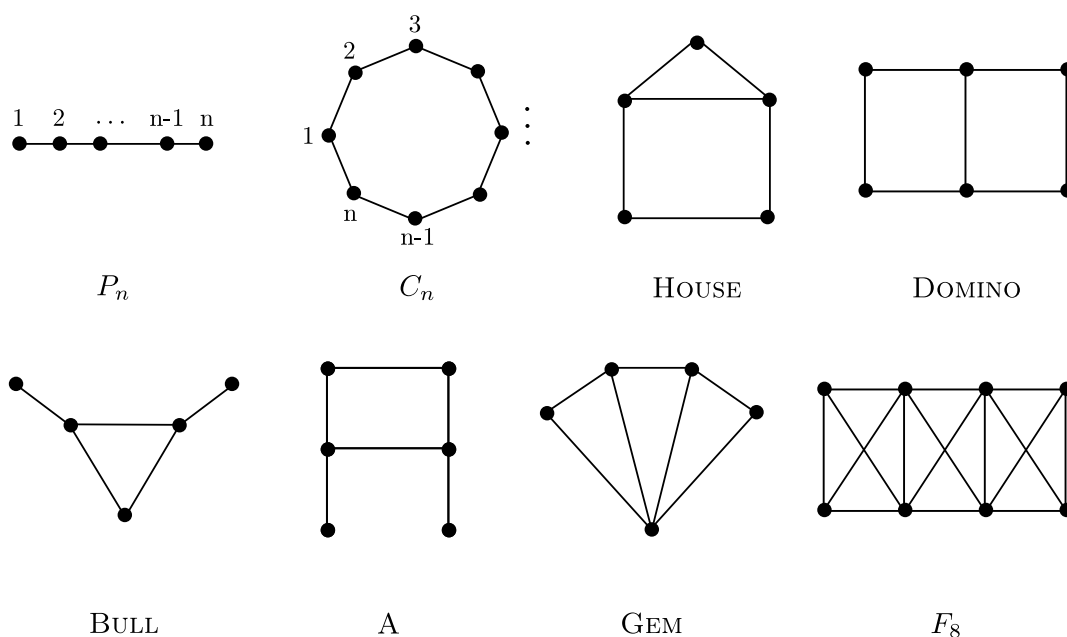


Figura 2.5: Algunos grafos que estudiaremos en esta sección.

Definición 2.20 Un grafo se dirá **primo** si sus únicos módulos son $V(G)$ y los singleton. A estos módulos los llamaremos los **módulos triviales** de G .

Como corolario de la **proposición 2.18**, tenemos el siguiente lema, que nos da una herramienta nueva para encontrar subgrafos prohibidos en \mathcal{T}^* .

Lema 2.21 \mathcal{T}^* es libre de grafos primos, salvo los árboles.

Demostración:

Sea H un grafo primo. Supongamos que $H \sqsubseteq G$, con $G \in \mathcal{T}^*$. Sean entonces T el bosque subyacente de G y V_1, \dots, V_r su partición asociada a T . Si existieran dos vértices de H en el mismo V_i , entonces $V(H) \cap V_i$ es un módulo no trivial de H , pues es módulo de H y $H \sqsubseteq G$. Luego, cada V_i tiene a lo más un vértice de H y, por lo tanto, H es subgrafo de un bosque.

Como un bosque primo debe ser un árbol, entonces H debe ser un árbol.

Notemos que si cambiamos \mathcal{T}_0 por una familia cerrada para subgrafos inducidos, entonces el resultado anterior sigue siendo válido, salvo que los subgrafos primos que pueden aparecer son los que pertenecen a la nueva familia.

Proposición 2.22 C_k , el ciclo de k vértices, es primo para $k \geq 5$.

Demostración:

Sean v_1, \dots, v_k los vértices de C_k y sea M módulo en C_k . Supongamos que $v_1, v_2 \in M$. Como $v_1v_3 \notin E(C_k)$, $v_2v_3 \in E(C_k)$, entonces $v_3 \in M$. Repitiendo el argumento, tenemos que $v_4, \dots, v_k \in M$ y $M = V(C_k)$.

Supongamos ahora que $v_1, v_j \in M$ con $j \neq 2, k$. Sin pérdida de generalidad, $j \leq \lfloor \frac{k}{2} \rfloor$. Como $k \geq 5$, entonces $v_jv_k \notin E(C_k)$, pero $v_1v_k \in E(C_k)$. Luego, $v_k \in M$ y nos reducimos al caso anterior.

De la proposición anterior, tenemos que no puede haber una familia finita de subgrafos prohibidos que defina a \mathcal{T}^* , pues los ciclos de tamaño al menos 5 están prohibidos, pero sus subgrafos propios son bosques, los que sí están en \mathcal{T}^* .

Varios de los grafos que veremos a continuación son primos. Sin embargo, en vez de demostrar que lo son, veremos que no pertenecen a \mathcal{T}^* usando argumentos de la sección anterior, lo que entregará demostraciones más sucintas.

Proposición 2.23 $\text{HOUSE} \notin \mathcal{T}^*$.

Demostración:

Tenemos que $\text{diam}(\text{HOUSE}) = 2$ y $\overline{\text{HOUSE}} = P_5$ es conexo. Por la **proposición 2.14**, $\text{HOUSE} \notin \mathcal{T}^*$.

Corolario 2.24 \mathcal{T}^* no es cerrada para complemento.

Demostración:

Tenemos que $P_5 \in \mathcal{T}^*$, pues es un bosque. Sin embargo, su complemento es HOUSE , que no está en \mathcal{T}^* .

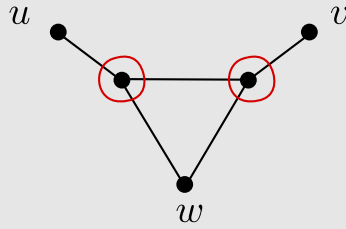
Proposición 2.25 $\overline{C}_k \notin \mathcal{T}^*$, $k \geq 5$.

Demostración:

Tenemos que $\text{diam}(\overline{C}_k) = 2$ y su complemento, C_k , es conexo, por lo se concluye de la **proposición 2.14**.

Proposición 2.26 $BULL \notin \mathcal{T}^*$.

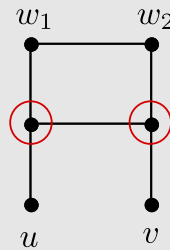
Demostración:



Como u, v son los únicos vértices a distancia al menos 3, entonces REC3 clasificará a los vértices de $I[u, v]$ en su propio conjunto. El resto de los vértices deberá quedar en conjuntos asociados a hojas del árbol subyacente. Sin embargo, w es vecino de más de un vértice interno de éste árbol, por lo que BULL no puede estar en \mathcal{T}^* .

Proposición 2.27 $A \notin \mathcal{T}^*$.

Demostración:



Al igual que antes, u, v son los únicos vértices a distancia 3 o más. Entonces, REC3 clasificará a los vértices de $I[u, v]$, dejando a w_1, w_2 en hojas distintas del bosque subyacente. Pero como w_1, w_2 son vecinos no pueden estar en hojas distintas.

Definición 2.28 A un ciclo inducido C_k , $k \geq 5$, en un grafo G , lo llamaremos un HOLE. De igual manera, a un anticiclo inducido \overline{C}_k , $k \geq 5$, en un grafo G lo llamaremos un ANTIHOLE.

Teorema 2.29 Los grafos en \mathcal{T}^* son HOUSE-HOLE-ANTI-HOLE-BULL-A-DOMINO-free.

Algunos de los subgrafos prohibidos que encontramos resultan ser redundantes. Por ejemplo, $A \sqsubseteq \text{DOMINO}$. Además, $\text{HOUSE} \sqsubseteq \overline{C}_k$, $k \geq 6$, y C_5 es isomorfo a su complemento, por lo que ser HOUSE-HOLE-free implica ser ANTIHOLE-free. Luego, la proposición anterior es equivalente a la siguiente:

Teorema 2.30 Los grafos en \mathcal{T}^* son HOUSE-HOLE-BULL-A-free.

Finalmente, desconocemos si estos grafos son suficientes para caracterizar a \mathcal{T}^* , pero conjeturamos que analizando los casos en los cuales el algoritmo de reconocimiento es negativo, se

pueden encontrar los subgrafos prohibidos faltantes, si es que los hubiera, o demostrar que los anteriores son suficientes.

2.3.2. Relación con familias conocidas

Grafos Perfectos

Definición 2.31 Sea G un grafo. Su **número cromático**, $\chi(G)$, es el tamaño de la mínima partición en independientes de G . Alternativamente, es el menor $n \in \mathbf{N}$ tal que existe una función de coloreo $c: V(G) \rightarrow [n]$ que satisface $uv \in E(G) \Rightarrow c(u) \neq c(v)$.

Definición 2.32 Sea G un grafo. Su **número de clique**, $\omega(G)$, es el máximo tamaño de un clique en G .

Como cada vértice de un clique tiene que estar en un independiente distinto, entonces para todo grafo G se tiene que $\omega(G) \leq \chi(G)$. Esto inspira la siguiente definición:

Definición 2.33 Un grafo se dice **perfecto** si para todo subgrafo inducido en él, su número de clique y su número cromático coinciden.

En el contexto algorítmico, muchos problemas NP-completos se pueden resolver en tiempo polinomial en grafos perfectos [15], como clique máximo, independiente máximo y determinar el número cromático. Sin embargo, tales algoritmos recurren a técnicas de programación lineal. En particular, se basan en el método de la elipsoide [14], por lo que para muchas subfamilias de los grafos perfectos se buscan algoritmos alternativos, que resultan ser más eficientes.

Los grafos perfectos también se pueden caracterizar por subgrafos prohibidos gracias al teorema fuerte de los grafos perfectos [7]:

Teorema 2.34 Un grafo es perfecto si y sólo si es libres de HOLES de tamaño impar y de ANTIHOLES de tamaño impar.

Del teorema anterior y del **teorema 2.30**, tenemos que los grafos de \mathcal{T}^* son perfectos. Sin embargo, es posible demostrar esto sin requerir del teorema fuerte de los grafos perfecto, lo que presentamos a continuación:

Teorema 2.35 Los grafos de \mathcal{T}^* son perfectos.

Demostración:

Como \mathcal{T}^* es cerrada para subgrafo inducido, nos basta probar que $\omega(G) = \chi(G)$, $\forall G \in \mathcal{T}^*$. Sabemos que es cierto en los bosques, por lo que veamos el caso de $G \in \mathcal{T}^* \setminus \mathcal{T}_0$. Además, se tiene que tanto el número cromático como el de clique de un grafo desconexo corresponden al máximo entre las componentes conexas, por lo que supondremos que G es conexo.

Sea T el árbol subyacente de G y V_1, \dots, V_r su partición asociada. Llamemos $G_i = G[V_i]$. Tenemos que como los cliques están contenidos en a lo más dos V_i , los cuales además son vecinos en T , entonces

$$\omega(G) = \text{máx}\{w(G_i) + w(G_j) : V_i V_j \in E(T)\}.$$

Por otra parte, como los vértices de V_i están conectados con todos los vértices de V_j , entonces

$$\chi(G) \geq \text{máx}\{\chi(G_i) + \chi(G_j) : V_i V_j \in E(T)\} = K.$$

Veamos que G es K -coloreable. Consideremos un V_i arbitrario como raíz de T y asignemos colores a V_j , $j \in [r]$, de la siguiente manera: a cada V_j le corresponderán los primeros $\chi(G_j)$ colores que no hayan sido usados por su padre. En el procedimiento anterior, solo se necesitan K colores en total, pues las restricciones solo están dadas por las aristas de T .

En lo que sigue, estudiaremos subfamilias de los grafos perfectos, para ver qué información de \mathcal{T}^* podemos recuperar de éstas.

Grafos débilmente cordados

Definición 2.36 *Un grafo se dice **débilmente cordado** si es HOLE-ANTIHOLE-free.*

Los grafos débilmente cordados son una de las primeras subfamilia de los grafos perfectos en la que ya hay algoritmos polinomiales que evitan la programación lineal. Por ejemplo, se pueden resolver coloreo y clique en tiempo $O(|V(G)|^3)$ y partición en cliques mínima en tiempo $O(|V(G)||E(G)|)$ [17]. En el caso de las versiones con peso de los problemas anteriores, estos se pueden resolver en tiempo $O(|V(G)|^4)$ [20].

Como tenemos que los grafos de \mathcal{T}^* son débilmente cordados, por el **teorema 2.30**, entonces se puede resolver el problema de partición en clique mínima en tiempo $O(|V(G)||E(G)|)$. Este problema tomará relevancia más adelante, cuando estudiemos las particiones convexas en \mathcal{T}^* .

Distance Hereditary

Definición 2.37 *Un grafo G se dice **distance hereditary** si las distancias en todo subgrafo inducido conexo H son iguales a las de G .*

Los grafos distance hereditary son una subfamilia de los grafos perfectos [18]. Algunos parámetros asociados a la convexidad, que en general son NP-completos, se pueden calcular en tiempo polinomial en esta familia [19].

Los siguientes dos teoremas entregan caracterizaciones de los grafos distance-hereditary por dos caminos distintos: una los construye mediante una secuencia de operaciones y el otro los

caracteriza por subgrafos prohibidos. Ambos teoremas están demostrados en [2].

Teorema 2.38 *Un grafo es distance hereditary si y sólo si se puede construir a partir de un vértice usando las siguientes operaciones:*

1. *Agregar una hoja al grafo.*
2. *Reemplazar un vértice por un independiente de dos vértices, los que llamamos **gemelos falsos**.*
3. *Reemplazar un vértice por un clique de dos vértices, los que llamamos **gemelos**.*

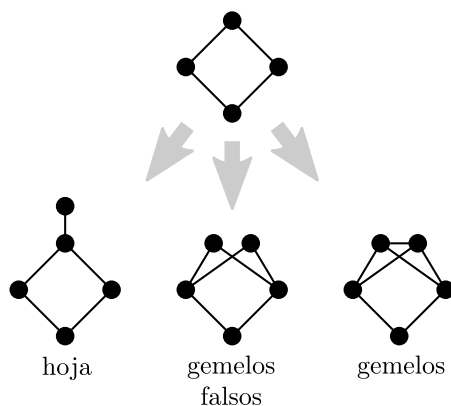


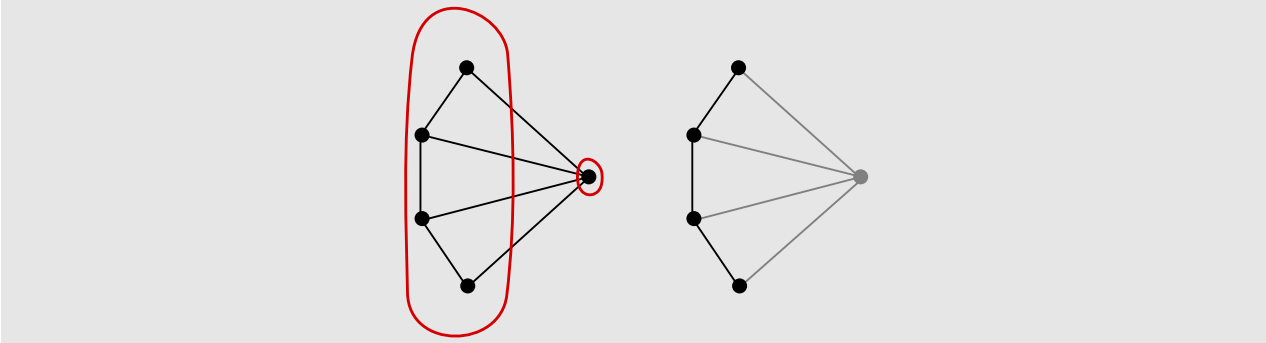
Figura 2.6: Las tres operaciones para construir un grafo distance hereditary.

Teorema 2.39 *Los grafos distance-hereditary corresponden a los grafos HOUSE-HOLE-DOMINO-GEM-free.*

Proposición 2.40 *Los grafos en \mathcal{T}^* no necesariamente son distance hereditary.*

Demostración:

Tenemos que $GEM \in \mathcal{T}_1$, pero contiene un subgrafo inducido que no mantiene las distancias:



Proposición 2.41 *Los grafos distance hereditary no necesariamente están en \mathcal{T}^* .*

Demostración:

Tenemos que BULL es distance hereditary, pues consiste de un triángulo al que se le agregan dos hojas. Además, los cliques se pueden construir mediante secuencias de reemplazos por gemelos. Por otro lado, \mathcal{T}^* es BULL-free, por **teorema 2.30**.

Module-composed

Definición 2.42 *Un grafo G se dice **module-composed** si existe un orden v_1, \dots, v_n de sus vértices tal que $N(v_i) \cap \{v_1, \dots, v_{i-1}\}$ es un módulo en $G[\{v_1, \dots, v_{i-1}\}]$, $\forall i \in \{2, \dots, n\}$.*

Los grafos module-composed también son débilmente cordados y, por lo tanto, perfectos [16]. Como en \mathcal{T}^* también aparecen módulos naturalmente en su construcción, estos grafos parecen ser buenos candidatos a estar relacionados con \mathcal{T}^* , pero veremos que no resultan serlo. Para comparar esta familia con \mathcal{T}^* usaremos el siguiente resultado:

Proposición 2.43 [16] *El resultado de agregarle una hoja a un grafo module-composed también es module-composed.*

Proposición 2.44 *Los grafos module-composed no están necesariamente en \mathcal{T}^* .*

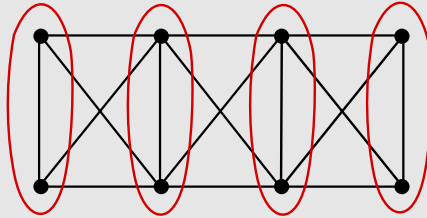
Demostración:

Notemos que BULL es un grafo module-composed, pues se obtiene de agregarle dos hojas a un triángulo, y los cliques son trivialmente module-composed.

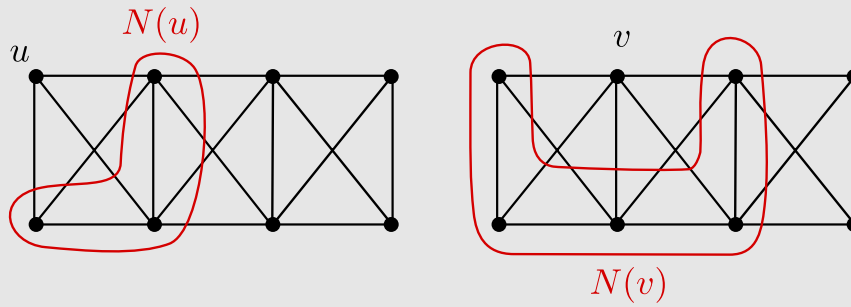
Proposición 2.45 *Los grafos de \mathcal{T}^* no son necesariamente module-composed.*

Demostración:

Tenemos que el grafo $F_8 \in \mathcal{T}_1$



Sin embargo, veremos que no existen ningún vértice de éste tal que su vecindad sea un módulo, el cual debiera existir si F_8 fuera module-composed. Por la simetría del grafo, solo queda verificar dos casos:



Como en ninguno de los casos la vecindad del vértice es un módulo, F_8 no es module-composed.

Cografos

Definición 2.46 *Los cografos son los grafos que no tienen a P_4 como subgrafo inducido.*

Los cografos son otra subfamilia de los grafos perfectos. Estos grafos son interesantes desde el punto de vista de las particiones convexas, pues este problema se puede resolver en tiempo lineal en esta familia [1]. Otros problemas NP-completos que se resuelven en tiempo lineal en los cografos incluyen a como ciclo hamiltoniano [9] y coloración [1].

Éstos grafos también pueden caracterizarse mediante la construcción a través de una secuencia de operaciones.

Proposición 2.47 [10] *Un grafo es un cografo si y sólo si se puede construir a partir de un vértice mediante las siguientes operaciones:*

1. *Unir dos cografos.*
2. *Tomar complemento de un cografo.*

Proposición 2.48 *Los grafos de \mathcal{T}^* no son necesariamente cografos.*

Demostración:

P_4 pertenece a \mathcal{T}_0 , pues es un bosque.

Teorema 2.49 *Los cografos están en \mathcal{T}^* .*

Demostración:

Procederemos por inducción en la cantidad de vértices. Claramente, un vértice aislado está en \mathcal{T}^* . Sea ahora G un cografo de al menos dos vértices. Hay dos casos:

1. G no es conexo. Entonces G se construyó uniendo dos cografos G_1, G_2 . Por hipótesis inductiva, $G_1, G_2 \in \mathcal{T}^*$. Como \mathcal{T}^* es cerrada para uniones, entonces $G \in \mathcal{T}^*$.
2. G es conexo. Entonces, G se construyó como complemento del cografo \bar{G} . A su vez, \bar{G}

se construyó uniendo dos cografos $\overline{G}_1, \overline{G}_2$. Por hipótesis inductiva, estos grafos están en \mathcal{T}^* . Como $G = G_1 + G_2$, y \mathcal{T}^* es cerrada para suma de grafos, entonces $G \in \mathcal{T}^*$.

La demostración anterior inspira la siguiente caracterización de los cografos, en términos de la definición de \mathcal{T}^* .

Definición 2.50 *Sea A el conjunto de bosques tales que todas sus componentes conexas tienen diámetro a lo más 2. La familia \mathcal{A}^* queda definida recursivamente reemplazando los vértices de un bosque de A por grafos de \mathcal{A}^* y usando como caso base a los grafos de A .*

Al igual que el caso de \mathcal{T}^* , para \mathcal{A}^* consideraremos los bosques subyacentes minimales, lo que hace que solo sean relevantes los bosques que son uniones disjuntas de aristas.

Teorema 2.51 *Los cografos corresponden a la familia \mathcal{A}^* .*

Demostración:

La demostración del **teorema 2.49** nos dice que los cografos están en \mathcal{A}^* . Ahora, sea G un grafo de \mathcal{A}^* . Veamos que no puede tener P_4 inducido. Si G es un bosque, entonces no puede tener a P_4 , pues tendría una componente conexas de diámetro al menos 3. Supongamos entonces que G fue construido mediante reemplazos, por lo que sea $\{V_1, V_2\}$ la partición asociada a la arista subyacente de G . Sean v_1, v_2, v_3, v_4 vértices de G , entonces hay tres casos

- $v_1 \in V_1$ y $v_2, v_3, v_4 \in V_2$. Entonces, v_1 es adyacente a los otros tres vértices, por lo que no pueden inducir P_4 .
- $v_1, v_2 \in V_1$ y $v_3, v_4 \in V_2$. Entonces, los vértices inducen al menos a C_4 , por lo que no pueden inducir a P_4 .
- $v_1, v_2, v_3, v_4 \in V_1$. Entonces, el resultado se tiene por inducción, pues $G[V_1] \in \mathcal{A}^*$.

Como mencionamos antes, otros problemas NP-completos se pueden resolver en tiempo polinomial en los cografos. Por ejemplo, ciclo hamiltoniano, que es NP-completo en los grafos perfectos, es polinomial en los cografos. Como \mathcal{T}^* es una familia intermedia entre las dos anteriores, puede resultar interesante estudiar tales problemas en \mathcal{T}^* .

Grafos cordados

Como mencionamos en el **capítulo 1**, el problema de p -partición convexa se puede resolver trivialmente en los grafos cordados, pues existe una p -partición convexa, $\forall p \in [|V(G)|]$. La técnica para demostrar lo anterior, recupera la idea de los árboles de encontrar un orden en el que se pueden borrar vértices manteniendo una propiedad importante. En este caso, se pueden eliminar vértices de un grafo cordado de modo que lo que queda es convexo en el grafo original [1].

Por otra parte, como los grafos cordados son grafos perfectos, hay más problemas relevantes que se pueden resolver en esta familia, como encontrar su número cromático o el tamaño de

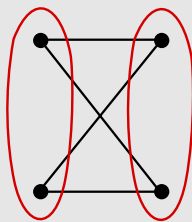
clique máximo. Más aún, problemas se pueden resolver en tiempo lineal en los grafos cordados [12].

Sin embargo, aunque estos grafos inspiran la idea de recuperar propiedades importantes de los árboles, estos resultan no tener mucha relación con los grafos de \mathcal{T}^* .

Proposición 2.52 *Los grafos de \mathcal{T}^* no necesariamente son cordados.*

Demostración:

Tenemos que C_4 está en \mathcal{T}^* , el cual no es cordado.



Proposición 2.53 *Los grafos cordados no necesariamente están en \mathcal{T}^* .*

Demostración:

Tenemos que BULL es un grafo cordado, pues su único ciclo es de tamaño 3.

Grafos intervalo

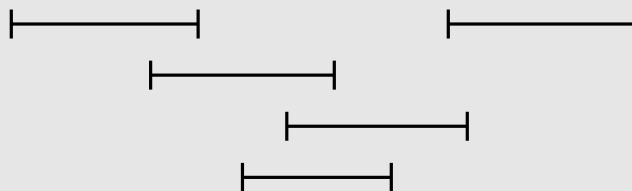
Definición 2.54 *Un grafo se dice grafo intervalo si sus vértices se pueden representar como un conjunto de intervalos en la recta real, de modo que dos vértices son adyacentes si y sólo si sus intervalos representantes se intersectan.*

Los grafos intervalo son una subfamilia de los grafos cordados.

Proposición 2.55 *Los grafos intervalos no están necesariamente en \mathcal{T}^* .*

Demostración:

Notemos que BULL lo podemos representar con una familia de intervalos como la siguiente



Grafos homogéneamente ordenables hereditarios

Definición 2.56 *Un vértice v de un grafo G se dice **h-extremal** si*

$$D_2(v) = \{u \in V(G) : d(u, v) \leq 2\}$$

contiene un módulo $M \subsetneq D_2(v)$ dominante en $G[D_2(v)]$. Es decir, todo vértice de $D_2(v) \setminus M$ es vecino de un vértice en M .

Definición 2.57 *Un orden v_1, \dots, v_n de los vértices de G se dice **orden de eliminación homogéneo** si v_i es h-extremal en $G[\{v_i, \dots, v_n\}]$, $\forall i \in [n-1]$. Si G tiene tal orden, se dice **homogéneamente ordenable**.*

Definición 2.58 *Un grafo se dice **homogéneamente ordenable hereditario** si todos sus subgrafos inducidos son homogéneamente ordenables.*

Los grafos homogéneamente ordenables hereditarios son una superfamilia de algunas de las familias mencionadas anteriormente, como los grafos module-composed o los distance-hereditary [5].

En esta familia, se pueden resolver otros problemas NP-completos que no habíamos mencionado anteriormente, como el problema del árbol de Steiner [5].

Definición 2.59 *Un grafo cordado G se dice k -SUN si $V(G) = U \cup W$, $U \cap W = \emptyset$ y*

1. $U = \{u_1, \dots, u_k\}$ es independiente, $W = \{w_1, \dots, w_k\}$ es un ciclo (no necesariamente inducido)
2. las únicas aristas incidentes a u_i son $u_i w_{i-1}, u_i w_{i+1}$, donde consideramos los índices módulo k .

Además, estos grafos resultan tener la siguiente caracterización por subgrafos prohibidos, mostrada en [5]:

Teorema 2.60 *Los subgrafos homogéneamente ordenables hereditario resultan ser los grafos HOUSE-HOLE-DOMINO-SUN-free.*

Como tenemos que $\text{BULL} \sqsubseteq \text{SUN}$, entonces por el **teorema 2.30** los grafos de \mathcal{T}^* son homogéneamente ordenables hereditarios.

Leaf power

Definición 2.61 *Sea $k \geq 2$. Un grafo G se dice **k -leaf power** si existe un árbol T tal que sus hojas corresponden a los vértices de G y $uv \in E(G)$ si y sólo si $d_T(u, v) \leq k$.*

Los grafos k -leaf power son una subfamilia de los grafos cordados [3], por lo que en general no están relacionados con \mathcal{T}^* . Sin embargo, veremos que para los primeros k , están contenidos en \mathcal{T}^* . Nos serán de interés las siguiente propiedades:

Proposición 2.62 [3] *Un grafo es 2-leaf power si y sólo si es unión disjunta de cliques*

Proposición 2.63 [3] *Un grafo es 3-leaf power si y sólo si se obtiene de un bosque al reemplazar sus vértices por cliques.*

Definición 2.64 *Dado un grafo G , definimos su potencia k , G^k , como el grafo con vértices en $V(G)$ tal que $uv \in E(G^k)$ si y sólo si $d(u, v) \leq k$.*

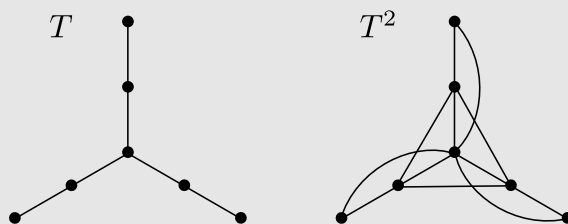
Proposición 2.65 [4] *Si T es un árbol, entonces T^2 es 4-leaf power.*

Proposición 2.66 *Los grafos 2 y 3-leaf power están en \mathcal{T}^* , pero los 4-leaf power no necesariamente.*

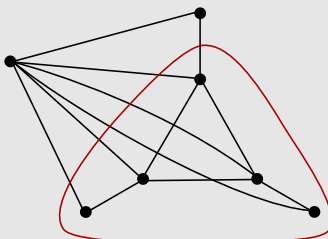
Demostración:

Por las **proposiciones 2.62, 2.63**, los grafos 2 y 3-leaf power están contenidos en \mathcal{T}^* , pues resultan de reemplazar los vértices de un bosque por cliques y los cliques están en \mathcal{T}^* .

Por otra parte, consideremos el siguiente árbol



Por la **proposición 2.65**, T^2 es 4-leaf power. Sin embargo, al reordenar el dibujo de T^2 es fácil ver que tiene a BULL como subgrafo inducido. Por lo tanto, $T^2 \notin \mathcal{T}^*$.



Resumen

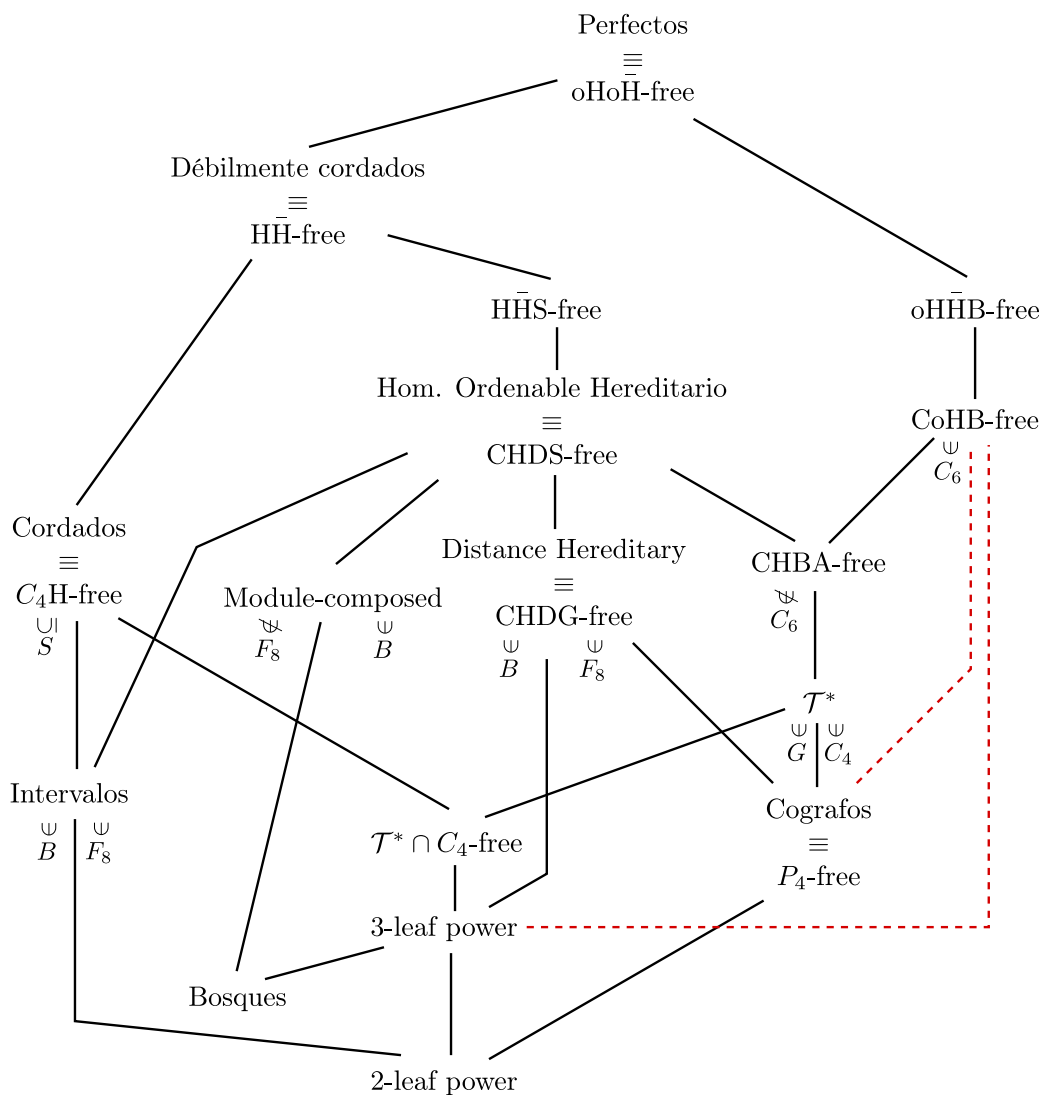


Figura 2.7: Resumen de las inclusiones entre las familias de grafos mencionadas. Al mencionar familias libres de algún subgrafo inducido, usaremos la siguiente notación: C = HOUSE, B = BULL, D = DOMINO, G = GEM, S = SUN, H = HOLES, \bar{H} = ANTIHOLES, oH = HOLES de tamaño impar, o \bar{H} = ANTIHOLES de tamaño impar. En la literatura, se abrevian HOLE y HOUSE con H, por lo que familias como CHDS-free se conocen como HHDS-free. Marcamos con una línea punteada las inclusiones que estaban documentadas como las mejores en la literatura, pero que como consecuencia de este trabajo ya no lo son.

Capítulo 3

Convexidad

Para lo que sigue nos interesa una suerte de análogo del **lema 1.20**, pero en vez de trabajar con las componentes conexas del grafo, trabajaremos con las componentes que quedan al retirar un clique. Esta idea extiende la noción de las componentes 2-conexas, que se forman al retirar un vértice de corte.

Consideremos G un grafo conexo, $U \subseteq V(G)$ un clique en G y V_1, \dots, V_r las componentes conexas de $G \setminus U$. Definimos $V'_i = V_i \cup U$, $G_i = G[V'_i]$.

Lema 3.1 *Si \mathcal{P} es partición convexa de G , entonces $\mathcal{P}_i = \{P \cap V'_i : P \in \mathcal{P}\} \setminus \{\emptyset\}$ es partición convexa de G_i , $\forall i \in [r]$.*

Demostración:

Claramente, \mathcal{P}_i es partición de V'_i . Ahora, veamos que sus elementos son convexos. Sean $P \in \mathcal{P}$ tal que $P \cap V'_i \in \mathcal{P}_i$ y $u, v \in P \cap V'_i \in \mathcal{P}_i$. Un (u, v) -camino mínimo en G no puede usar vértices de V_j , $j \neq i$, pues se podría acortar mediante U , ya que éste es un clique. Luego, los (u, v) -caminos mínimos en G_i son los mismos que los de G . Como \mathcal{P} es partición convexa, entonces los (u, v) -caminos mínimos se mantienen dentro de P . Por lo tanto, los (u, v) -caminos mínimos se mantienen dentro de $P \cap V'_i$.

Si ahora queremos realizar el procedimiento inverso, es decir, tenemos particiones convexas \mathcal{P}_i en cada G_i y queremos recuperar una partición \mathcal{P} de G , la primera condición es la siguiente:

Definición 3.2 *Si las particiones \mathcal{P}_i son particiones de G_i que inducen la misma partición en U para $i \in [k]$, diremos que $\{\mathcal{P}_i\}_{i=1}^r$ es U -compatible.*

Si $\{\mathcal{P}_i\}_{i=1}^r$ es U -compatible, a la partición inducida en U la llamaremos \mathcal{P}_U . La construcción natural es la siguiente: para cada $W \in \mathcal{P}_U$ y para cada $i \in [r]$, llamemos W_i al conjunto en \mathcal{P}_i tal que $W \subseteq W_i$ y consideremos la unión de todos ellos, $W' = \bigcup_{i=1}^r W_i$. La partición será

$$\mathcal{P} = \bigcup_{i=1}^r (\mathcal{P}_i \setminus \{W_i : W \in \mathcal{P}_U\}) \cup \{W' : W \in \mathcal{P}_U\}$$

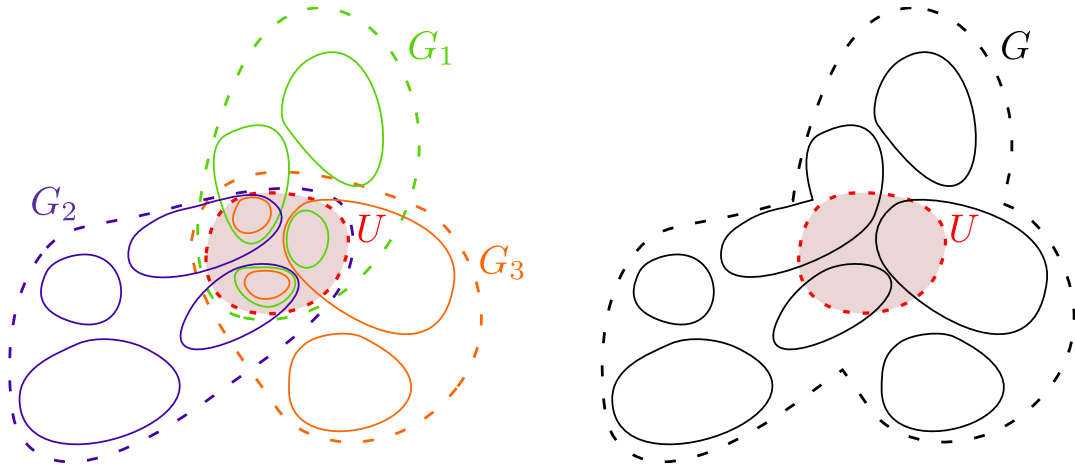


Figura 3.1: Diagrama de la construcción de \mathcal{P} . A la izquierda, tenemos las particiones en G_1, G_2, G_3 y a la derecha, la partición resultante.

En general, \mathcal{P} no es partición convexa, pero el siguiente lema nos da algunas condiciones para ello.

Lema 3.3 Si $\{\mathcal{P}\}_{i=1}^r$ es U -compatible y U además de clique es un módulo de G , entonces \mathcal{P} es partición convexa de G si y solo si se cumple alguna de las siguientes condiciones:

1. $\mathcal{P}_U = \{U\}$.
2. $\forall W \in \mathcal{P}_U, \exists i \in [r] : W' \in \mathcal{P}_i$.

Demostración:

(\Leftarrow) Primero, veremos que si $S \in \mathcal{P}_i$ para algún $i \in [r]$, entonces S es convexo en G . Sean $u, v \in S$. Como U es clique, entonces los (u, v) -caminos mínimos están contenidos en V'_i . Como además S es convexo en G'_i , entonces los (u, v) -caminos permanecen en S . De lo anterior, solo necesitamos verificar que los conjuntos de \mathcal{P} que no están en algún \mathcal{P}_i son convexos en G . Esto solo ocurre en el caso $\mathcal{P}_U = \{U\}$ y el único conjunto que no está en un \mathcal{P}_i es U' , por lo que solo resta verificar que ese conjunto es convexo. Sean $u, v \in U'$. Hay dos casos

1. $u, v \in V'_i$. Entonces, como U es clique, los (u, v) -caminos mínimos G se mantienen en V'_i y por lo tanto coinciden con los (u, v) -caminos mínimos de G_i . Además, como $U' \cap V'_i \in \mathcal{P}_i$, entonces es convexo en G_i , por lo que los (u, v) -caminos mínimos están contenidos en $U' \cap V'_i \subseteq U'$.
2. $u \in V_i, v \in V_j, i \neq j$. Entonces, todos los (u, v) -caminos pasan por U y, por lo tanto, los (u, v) -caminos mínimos en G son de la forma $uPxyP'v$, donde $x, y \in U$ (eventualmente son iguales) y $P \in \mathcal{G}(u, x), P' \in \mathcal{G}(y, v)$. Como $U' \cap V'_i = U_i, U' \cap V'_j = U_j$ son convexos, entonces los (u, v) -caminos mínimos están contenidos en U' .

(\Rightarrow) Supongamos que $\mathcal{P}_U \neq \{U\}$ y sea $W \in \mathcal{P}_U$. Supongamos además que W' es convexo y que existen $i, j \in [k], i \neq j$, tales que $W_i, W_j \neq W$. Sean $u \in W_i \setminus U$ y $v \in W_j \setminus U$. Como W' es convexo, existe un (u, v) -camino mínimo de la forma $uPxyP'v$, con $x, y \in W$.

Sin embargo, como U es módulo, podemos cambiar x por cualquier elemento de $U \setminus W$ y se sigue teniendo un (u, v) -camino mínimo. Por lo tanto, W' no es convexo $\rightarrow \leftarrow$.

Notemos que la condición de que U es un módulo solo es necesaria para demostrar que si \mathcal{P} es partición convexa, entonces se tienen las condiciones sobre \mathcal{P}_U .

Además, si $\{\mathcal{P}_i\}_{i=1}^r$ es U -compatible y cada \mathcal{P}_i es una p_i -partición convexa de G_i para $i \in [r]$, entonces G tiene una p -partición convexa, donde

$$p = \sum_{i=1}^r p_i - (r-1)\bar{p}$$

y \bar{p} es el tamaño de la partición \mathcal{P}_U inducida por $\{\mathcal{P}_i\}_{i=1}^r$ en U .

Además, tenemos que si $U = \{v\}$, donde v es un vértice de corte de G , entonces U es trivialmente un módulo de G , por ser un solo vértice, y siempre se tiene la primera condición del lema anterior. Así que siempre es posible reconstruir particiones convexas de G a partir de sus componentes 2-conexas.

3.1. Convexidad en \mathcal{T}^*

En esta sección, consideraremos $G \in \mathcal{T}_k \setminus \mathcal{T}_0$, $k \geq 1$, T un árbol subyacente minimal fijo y V_1, \dots, V_r , $r > 1$, su partición asociada. Además, llamaremos $G_i = G[V_i]$ a los grafos inducidos en cada conjunto de la partición asociada a T .

Proposición 3.4 *Si G es tal que ningún vértice interno de T es clique y T no es una arista, entonces si una partición convexa de G no es trivial de G , entonces es una partición en cliques.*

Demostración:

Sea S convexo en G , veremos que debe ser $V(G)$ o un clique, lo que implica el resultado. Supongamos que existen $u, v \in S$, $uv \notin E(G)$. Tenemos dos casos

1. $u, v \in V_i$. Sea V_j un vecino de V_i . Como $d(u, v) > 1$, entonces los elementos de $\{u\} \times V_j \times \{v\}$ son (u, v) -caminos mínimos y luego $V_j \subseteq S$. Por lo tanto, todos los vecinos de V_i están contenidos en S .

Si un vértice interno está contenido en S , podemos repetir el argumento anterior y llegar a que sus vecinos también están contenidos en S , pues no es un clique y, por lo tanto, contiene un par de vértices no adyacentes. Como V_i debe tener un vecino que es vértice interno, pues T es un árbol minimal que no es una arista, entonces llegamos a tener a todos los V_j contenidos en S . Por lo tanto, $S = V(G)$.

2. $u \in V_i$, $v \in V_j$, $i \neq j$ y V_i, V_j no son vecinos en T . Sea $V_i V_{i_1} \dots V_{i_p} V_j$ el (V_i, V_j) -camino mínimo en T . Entonces, por la **proposición 2.12**, el conjunto de (u, v) -caminos

mínimos es

$$\{u\} \times V_{i_1} \times \cdots \times V_{i_p} \times \{v\}$$

Luego, $V_{i_1} \subseteq S$, y como V_{i_1} es nodo interno en T , entonces no es un clique y nos reducimos al caso anterior, pues tenemos a dos vértices no adyacentes de V_{i_1} contenidos en S .

Luego, los únicos convexos que no son $V(G)$ deben ser cliques.

El caso en el que T es una arista, corresponde a la suma de los grafos G_1 y G_2 , por lo que las **proposiciones 1.30, 1.33** enumeran los casos para las particiones convexas de G .

El caso restante es cuando hay cliques en vértices internos de T . En tal caso, dividiremos G mediante los V_i internos que son cliques. Sean

$$C = \{i \in [r]: V_i \text{ es vértice interno de } T \text{ y es un clique}\}$$

y H el grafo obtenido a partir de G quitándole los V_i con $i \in C$. Llamemos $U_1, \dots, U_{q'}$ a las componentes conexas de H y para $i \in [q']$ definamos

$$U'_i = U_i \cup \{v \in V(G): v \text{ es vecino de algún } u \in U_i\}$$

$$H_i = G[U'_i]$$

Sean V_i, V_j dos vértices internos de T adyacentes. Si V_i y V_j son cliques, entonces la construcción anterior deja fuera a las aristas entre ellos. Es decir, nos falta considerar el subgrafo inducidos por $V_i \cup V_j$. Llamaremos $H_{q'+1}, \dots, H_q$ a una enumeración del siguiente conjunto

$$\{G[V_i \cup V_j]: i, j \in C, V_i V_j \in E(T)\}.$$

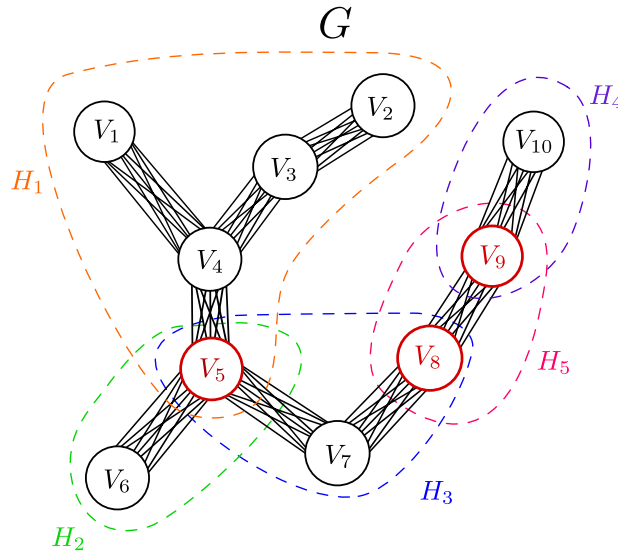


Figura 3.2: Diagrama que muestra los grafos H_i resultantes de cortar G por los V_j , $j \in C$. En este caso $C = \{5, 8, 9\}$.

Definición 3.5 Dados \mathcal{C} una partición de G y $S \subseteq V(G)$, diremos que \mathcal{C} corta a S si la partición inducida por \mathcal{C} en S no es la trivial, $\{S\}$.

Queremos usar el **lema 3.3** para reducir el problema de encontrar particiones convexas en G a encontrarlas en los H_i . Notemos que, dado $j \in C$ fijo, tenemos dos tipos de particiones convexas \mathcal{P}_i en los H_i , gracias a las **proposiciones 3.4, 1.30, 1.33**:

1. \mathcal{P}_i no corta a V_j .
2. \mathcal{P}_i es partición en cliques y corta a V_j .

En el segundo caso, veremos que podemos pedir algunas condiciones extras sobre la partición.

Definición 3.6 *Dados dos conjuntos A y B , diremos que se solapan si*

$$A \setminus B, B \setminus A, A \cap B \neq \emptyset.$$

Alternativamente, A y B no se solapan si

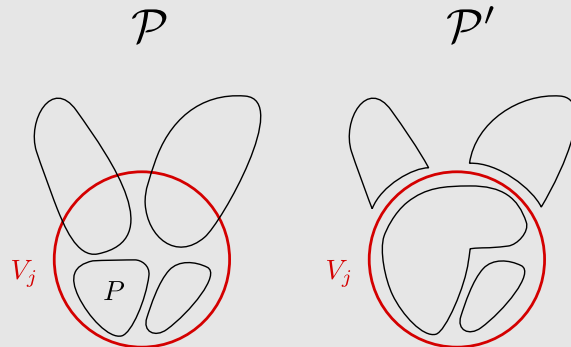
$$A \subseteq B \vee B \subseteq A \vee A \cap B = \emptyset.$$

Lema 3.7 *Si existe una p -partición convexa de G , entonces existe una p -partición convexa de G tal que ninguno de sus elementos se solapa con ningún V_j , $j \in C$.*

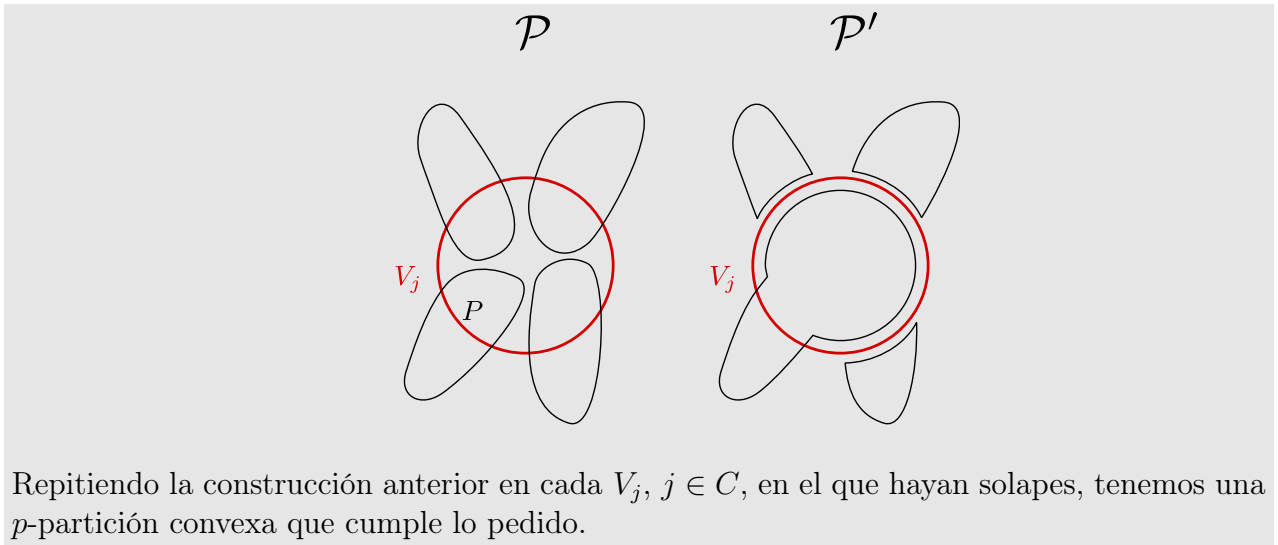
Demostración:

Sea \mathcal{P} p -partición convexa de G . Supongamos que existen elementos de \mathcal{P} que se solapan con algún V_j , $j \in C$. Construiremos \mathcal{P}' a partir de \mathcal{P} de modo que no se solape con V_j y mantenga su tamaño. Definamos $\mathcal{Q} = \{P \in \mathcal{P} : P \cap V_j \neq \emptyset\}$. Como \mathcal{P} corta a V_j , entonces los elementos de \mathcal{Q} deben ser cliques. Tenemos dos casos:

1. Existe $P \in \mathcal{Q}$ tal que $P \subseteq V_j$. Entonces, por cada $Q \in \mathcal{P}$ que se solape con V_j haremos la siguiente modificación a la partición: reemplazaremos Q por $Q \setminus V_j$ y P por $P \cup (Q \cap V_j)$. Como $Q \setminus V_j \neq \emptyset$, entonces se mantiene la cantidad de elementos de \mathcal{P} . Además, es claro que la partición resultante no se solapa con V_j . Como las partes distintas de la nueva partición son cliques, entonces son convexas y, por lo tanto, la nueva partición es convexa.



2. Todos los elementos de \mathcal{Q} se solapan con V_j . Sea $P \in \mathcal{Q}$ fijo. Cada elemento $Q \in \mathcal{Q}$ distinto de P lo reemplazaremos por $Q \setminus V_j$. Además, reemplazaremos P por $P \cup V_j$. Al igual que en el caso anterior, como $Q \setminus V_j \neq \emptyset$, entonces la partición mantiene su tamaño. Finalmente, por construcción, la nueva partición no solapa a V_j . Además, tenemos que la nueva parte, $P \cup V_j$, es un clique, pues P y V_j son cliques y V_j es un módulo. Luego, la nueva partición es una partición convexa.



Como el lema anterior nos permite trabajar con particiones que no se solapan con los V_j , $j \in C$, entonces se tienen las hipótesis del **lema 3.3** y basta solo con que la partición inducida en V_j sea la misma. Más aún, como V_j es un módulo y clique, entonces basta solo con que las particiones inducidas en V_j tengan el mismo tamaño.

Ahora, quedan por resolver dos problemas en \mathcal{T}^* : p -partición en cliques y p -partición convexa, los cuales revisaremos en lo que sigue.

3.1.1. Particiones en cliques

Como ya hemos mencionado, solo nos basta encontrar la partición en cliques de tamaño mínimo para un grafo, pues todos los tamaños mayores que ese son factibles. En lo que sigue, daremos la propiedad que nos permitirá calcular una partición mínima de manera recursiva para $G \in \mathcal{T}^*$.

Lema 3.8 *Existe un partición en cliques \mathcal{C} de G con tamaño mínimo, tal que la partición inducida por \mathcal{C} en cada G_i es mínima.*

Demostración:

Sea \mathcal{C} partición en cliques de G con tamaño mínimo y sea $\{C_1, \dots, C_Q\}$ la partición inducida por \mathcal{C} en G_{i_0} y supongamos que no es mínima. Sea $\{C'_1, \dots, C'_q\}$ una partición en cliques mínima de G_{i_0} . Entonces, $q < Q$.

Como un clique de G puede estar contenido en a lo más dos V_i , entonces existen cliques D_1, \dots, D_Q (algunos pueden ser \emptyset) en G tales que

$$C_i \cup D_i \in \mathcal{C}, \quad D_i \subseteq V_j \quad (j \neq i_0).$$

Consideremos ahora \mathcal{C}' dada por

$$\mathcal{C}' = \left(\left(\mathcal{C} \setminus \{C_i \cup D_i\}_{i=1}^q \right) \cup \{C'_i \cup D_i\}_{i=1}^q \cup \{D_i\}_{i=q+1}^q \right) \setminus \{\emptyset\}.$$

Tenemos que \mathcal{C}' es una partición de G en cliques tal que $|\mathcal{C}'| \leq |\mathcal{C}|$. Repitiendo el argumento para cada G_i con partición inducida no mínima, obtenemos una partición mínima de G que cumple lo pedido.

Notemos que al estar considerando cliques, sólo importa el tamaño de las particiones inducidas en cada G_i .

Proposición 3.9 *Sea \mathcal{C} partición en cliques de G y sea \mathcal{C}_i la partición inducida en G_i , $\forall i \in [r]$. Definamos*

$$x_{ij} = \left| \left\{ \{C, D\} : C \in \mathcal{C}_i, D \in \mathcal{C}_j, C \cup D \in \mathcal{C} \right\} \right|$$

Entonces,

$$|\mathcal{C}| = \sum_{i=1}^r |\mathcal{C}_i| - \sum_{V_i V_j \in E(T)} x_{ij}$$

Si tenemos particiones en cliques mínimas para cada G_i , resta combinarlas de manera óptima. En lo que sigue, sea \mathcal{C}_i partición en cliques mínima de G_i , $\forall i \in [r]$. Veremos que encontrar el tamaño de la partición en cliques mínima de G corresponde a encontrar el vector $\vec{x} = (x_{ij})_{V_i V_j \in E(T)}$ que optimiza el siguiente problema

$$\begin{aligned} X = \text{máx} \quad & \sum_{V_i V_j \in E(T)} x_{ij} \\ \text{s.a.} \quad & \sum_{j: V_i V_j \in E(T)} x_{ij} \leq c_i \quad \forall i \in [r] \end{aligned} \quad (3.1)$$

$$x_{ij} \in \mathbf{N} \quad \forall V_i V_j \in E(T) \quad (3.2)$$

En donde $c_i = |\mathcal{C}_i|$. Por la **proposición 3.9**, el tamaño de la partición óptima sería $\sum_{i=1}^r c_i - X$. Notemos que si en el problema anterior fijamos $c_{ij} = 1$, $\forall i \in [r]$, entonces el problema se transforma en encontrar el emparejamiento máximo en T . En el caso de los árboles, encontrar el emparejamiento máximo se puede hacer en tiempo lineal, usando un algoritmo glotón. Basados en la misma idea, resolveremos el problema de asignación anterior en tiempo lineal.

Lema 3.10 *Los vectores factibles del problema anterior se corresponden con las formas de combinar los cliques de las particiones mínimas, \mathcal{C}_i . Además, X se puede calcular en tiempo $O(r)$.*

Demostración:

Claramente, una partición en cliques genera un vector $\vec{x} = \{x_{ij} : V_i V_j \in E(T)\}$ factible, es decir, que satisface 3.1 y 3.2. Ahora resta probar que dado \vec{x} factible, entonces existe una

partición en cliques de G de tamaño $\sum_{i=1}^r c_i - \sum_{V_i V_j \in E(T)} x_{ij}$. Consideremos la siguiente construcción: por cada $V_i V_j \in E(T)$, tomamos x_{ij} parejas de elementos $C \in \mathcal{C}_i, D \in \mathcal{C}_j$ no utilizados hasta el momento y los reemplazamos por $C \cup D$. La restricción (3.1) nos asegura que siempre habrán elementos disponibles para la operación anterior. Así, la partición resultante cumple lo pedido.

Veamos que podemos encontrar X en tiempo $O(r)$. Consideremos el siguiente algoritmo glotón:

1. Sea V_i una hoja de T con padre V_j . Asignaremos $x_{ij} = \min\{c_i, c_j\}$.
2. Se repite el procedimiento para $T \setminus \{V_i\}$ y reduciendo c_j en x_{ij} .
3. Se termina cuando T solo contiene un vértice.

Comprobemos que lo anterior, en efecto, entrega una asignación óptima. Sea \vec{x} una asignación óptima. Construiremos una asignación \vec{y} con el mismo valor y tal que $y_{ij} = \min\{c_i, c_j\}$. Como \vec{x} es óptimo, entonces

$$\sum_{i: V_i V_j \in E(T)} x_{ij} = c_j.$$

De otro modo, se podría aumentar el valor de x_i y mejorar la solución. De la igualdad anterior, tenemos que si $x_{ij} < \min\{c_i, c_j\}$, entonces se pueden reducir valores en las aristas que contienen a V_j para así aumentar el valor de x_{ij} y alcanzar $\min\{c_i, c_j\}$.

Teorema 3.11 *Se puede encontrar la partición en cliques mínima de $G \in \mathcal{T}^*$ en tiempo $O(|V(G)|^4)$.*

Demostración:

Modificaremos el algoritmo de reconocimiento diseñado en la **sección 2.2**, de modo que también entregue la partición en cliques mínima de G .

Si $G \in \mathcal{T}_0$, entonces es un bosque. Luego, una partición en cliques de G corresponde a un emparejamiento y una partición mínima corresponde a un emparejamiento de tamaño máximo. La demostración del **lema 3.10**, muestra que se puede resolver en tiempo $O(|V(G)|)$.

Para $G \in \mathcal{T}^* \setminus \mathcal{T}_0$, combinar las particiones mínimas de los G_i se puede hacer en tiempo $O(r) = O(|V(G)|)$.

En ambos casos, las operaciones para calcular la partición en cliques, no aumentan la complejidad de las operaciones que ya se realizan en REC para combinar las soluciones de los subproblemas en la recursión. Por lo tanto, calcular una partición en cliques mínima toma tiempo $O(|V(G)|^4)$.

Como mencionamos en la **sección 2.3**, los grafos de \mathcal{T}^* son débilmente cordados. Para tal familia, se conoce un algoritmo en tiempo $O(|V(G)|^3)$ para resolver el problema de la partición en cliques de tamaño mínimo [20]. Por lo tanto, el algoritmo diseñado en esta sección no es el más eficiente si ya se sabe que el grafo a tratar está en \mathcal{T}^* . Sin embargo, el mejor algoritmo

conocido para reconocer un grafo en \mathcal{T}^* toma tiempo $O(|V(G)|^4)$. Así, en tiempo $O(|V(G)|^4)$, podemos entregar una partición en cliques mínima si $G \in \mathcal{T}^*$ o responder que $G \notin \mathcal{T}^*$ en tal caso.

3.1.2. Particiones localmente convexas

Como vimos en la **sección 1.3.2**, la unión de convexas a distancia mayor a 2 es un conjunto localmente convexo. En el caso de \mathcal{T}^* , estos resultan ser los únicos conjuntos localmente convexas.

Proposición 3.12 *Sea $G \in \mathcal{T}^*$. Entonces, $S \subseteq V(G)$ es localmente convexo si y solo si $S = \bigcup_{i=1}^r S_i$, donde S_i es convexo en G y $d(S_i, S_j) > 2$, para todo par $i, j \in [r]$.*

Demostración:

(\Rightarrow) Sean S_1, \dots, S_t las componentes conexas de S en G . Si existieran S_i, S_j tales que $d(S_i, S_j) = 2$, entonces existe al menos un camino de largo dos entre algún vértice de S_i y alguno de S_j . Pero como S es localmente convexo, debe contener a los vértices en tales caminos, lo que es una contradicción. Luego, $d(S_i, S_j) > 2$, para todo par $i, j \in [t]$. Además, tenemos que cada S_i es localmente convexo. Por lo tanto, nos basta demostrar que si S es localmente convexo y conexo, entonces es convexo.

Sea T el árbol subyacente de G y V_1, \dots, V_r su partición asociada. Primero, probaremos lo pedido para el caso en que los vértices internos de T no son cliques. Si S es clique, entonces es convexo. Si no, demostraremos que $S = V(G)$. Sean $u, v \in S$ tales que $uv \notin E(G)$. Tenemos dos casos:

1. $u, v \in V_i$. Sea V_j un vértice interno de T adyacente a V_i . Entonces $V_j \subseteq S$. Como los vértices internos de T no son cliques, entonces se puede repetir el argumento y entonces $V(G) \subseteq S$.
2. $u \in V_i, v \in V_j, i \neq j$. Como S es conexo, podemos suponer que $d(V_i, V_j) = 2$. De otro modo, basta reemplazar u, v por elementos del (u, v) -camino contenido en S que estén a distancia 2. Entonces, tenemos que existe V_k vecino de V_i, V_j . Como S es localmente convexo, entonces $V_k \subseteq S$. Por otra parte, como V_k es vértice interno de T no es clique y, por lo tanto, nos podemos reducir al caso anterior.

Para el caso general, consideremos los grafos H_1, \dots, H_q obtenidos de dividir a G por los vértices internos de T que son cliques, según la construcción expuesta en la **sección 3.1**. Como las distancias en H_i se preservan, entonces $S \cap H_i$ también es localmente convexo y conexo en H_i . Por el caso anterior, $S \cap H$ es convexo. Ahora, dado V_l vértice interno de T que es clique, queremos probar que no existen $i \neq j$ tales que $V(H_i) \cap S$ y $V(H_j) \cap S$ solapen a V_l clique. Si existieran, sean $u \in H_i \cap S \setminus V_l$ y $v \in H_j \cap S \setminus V_l$ tales que son adyacentes a algún vértice de V_l . Como V_l es módulo, entonces u y v son adyacentes a todos los vértices de V_l . Luego están a distancia 2, por lo que $V_l \subseteq S$, pero entonces $V_l \subseteq V(H_i) \cap S, V(H_j) \cap S$. Con lo anterior, se cumplen las hipótesis del **lema 3.3** y tenemos que S es convexo.

(\Leftarrow) Es directo de la **proposición 1.35** y de que los convexas son localmente convexas.

Por otra parte, sea S un conjunto localmente convexo de $G \in \mathcal{T}^*$. Consideremos además, \mathcal{C}_S la partición en cliques mínima de $G \setminus S$. Entonces los tamaños de particiones localmente convexas que contienen a S están en el intervalo $[\mathcal{C}_S + 1, |V(G)| - |S| + 1]$. Notemos que el extremo inferior de este intervalo no depende solo del tamaño de S , sino que puede depender de cuáles vértices lo componen. Por lo anterior, no parece haber una manera fácil de determinar cuáles son los conjuntos localmente convexas realmente importantes y es posible que sea necesario chequear todos los conjuntos localmente convexas, o al menos una gran parte de ellos, para describir el conjunto de tamaños de las particiones localmente convexas.

En lo que sigue, mediremos la complejidad de los algoritmos en términos de llamadas a un oráculo que decide si existe una p -partición localmente convexa en $G \in \mathcal{T}^*$, al que llamaremos $\text{LOC CONV}(G, p)$.

3.1.3. Programación Dinámica

Finalmente, para determinar si $G \in \mathcal{T}^*$ tiene una p -partición convexa, veremos como combinar las particiones en cada H_i , $i \in [q]$, donde H_i son los subgrafos que se obtienen de G al cortarlo por los vértices internos de su árbol subyacente que son cliques. En lo que sigue, supondremos que los H_i están ordenados de forma que $\bigcup_{i=1}^j H_i$ es conexo, $\forall j \in [q]$.

Definamos

$$L = \{i \in [q] : \text{el árbol subyacente de } H_i \text{ es una arista, con un clique y un no clique}\}.$$

Tenemos que H_i , $i \in L$, no puede estar en el interior del árbol subyacente de G , pues lo anterior requiere que ambos de sus extremos sean cliques. Luego, su extremo que no es clique debe corresponder a una hoja del árbol subyacente. Llamaremos $\text{hoja}(i)$ al índice correspondiente a tal hoja.

Para $j \in [q]$ consideremos los siguientes objetos

$$\begin{aligned} \hat{G}_j &= G \left[\bigcup_{i=1}^j V(H_i) \right] \\ p_j &= \text{el tamaño de la partición en cliques mínima de } H_j \\ C_j &= \{c \in C : V_c \subseteq V(H_j)\} \end{aligned}$$

Procederemos en dos etapas. Primero, determinaremos para cuáles p existe una p -partición convexa de G que no corta a ningún V_c , $c \in C$. Después, calcularemos los p restantes refinando las particiones que ya teníamos mediante los cliques V_c , $c \in C$. Sin embargo, no podemos refinar las particiones en cualquiera de estos cliques, pues en alguno de los H_i que contienen a V_c podemos haber usado una partición que no permite refinarlo. Entonces, diremos que V_c , $c \in C$, queda **fijo** si $c \in C_j$ y la partición convexa de H_j que se está considerando es la trivial o proviene de una partición localmente convexa.

Para resolver la primera etapa, usaremos la siguiente función booleana $a(S, j, p)$, donde $S \subseteq$

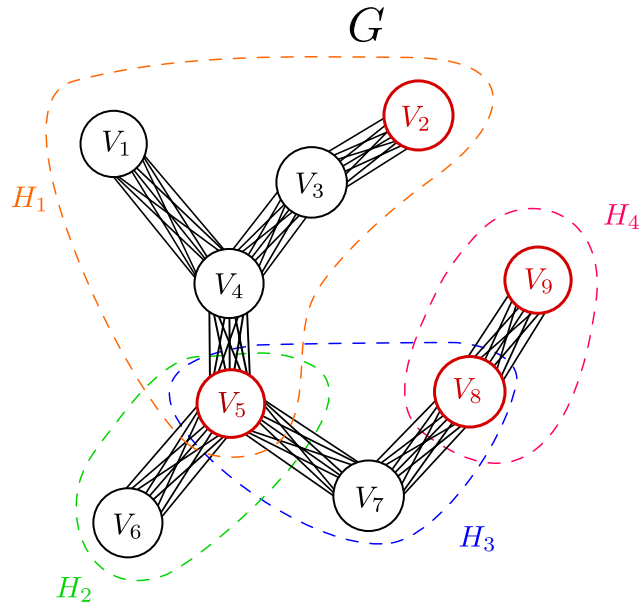


Figura 3.3: En la figura, tenemos que V_2, V_5, V_8, V_9 son cliques. En este caso, $L = \{5\}$ y $\text{hoja}(5) = 6$.

$C, j \in [q], p \in |V(G)|,$

$$a(S, j, p) = \begin{cases} 1 & \text{si existe una } p\text{-partición convexa de } \hat{G}_j \text{ que no corta a ningún } V_c, c \in C, \\ & \text{y tal que todos los } V_c, c \in S, \text{ quedan fijos,} \\ 0 & \text{si no.} \end{cases}$$

Para calcular esta función, veremos a la función a como una tabla que inicialmente tiene todas sus entradas con valor 0 y la iremos llenando con 1 donde corresponda. Primero, notemos que en el caso $j = 1$ se tiene que

- $a(C_1, 1, 1) = 1$, lo que corresponde a elegir la partición trivial en H_1 .
- $a(C_1, 1, p) = 1$, si $1 \in L$ y $\text{LOCCONV}(G_{\text{hoja}(1)}, p) = 1$, lo que corresponde a tomar las particiones convexas construidas a partir de las particiones localmente convexas de $G_{\text{hoja}(1)}$.
- $a(\emptyset, 1, p) = 1$ si $p \in [p_1, |V(H_1)| - \sum_{c \in C_1} |V_c|]$, lo que corresponde a elegir particiones en cliques en H_1 que no cortan a $V_c, c \in C_1$.

Ahora, si tenemos que $a(S, j, p) = 1$, queremos construir las particiones convexas de \hat{G}_{j+1} que surgen a partir de la de \hat{G}_j . Tenemos que

- $a(S', j+1, p) = 1$, donde $S' = S \cup C_{j+1}$. Esto corresponde a considerar la partición trivial en H_{j+1} .
- $a(S', j+1, p+p'-1) = 1$, si $j+1 \in L$, $\text{LOCCONV}(G_{\text{hoja}(j+1)}, p') = 1$ y donde $S' = S \cup C_{j+1}$. Esto corresponde a elegir las particiones obtenidas de las particiones localmente convexas de $G_{\text{hoja}(j+1)}$.
- $a(S, j+1, p+x-1) = 1$, donde $x \in [p_{j+1}, |V(H_{j+1})| - \sum_{c \in C_{j+1}} |V_c|]$. Esto corresponde a elegir las particiones en cliques de H_{j+1} que no cortan a $V_c, c \in C_{j+1}$.

De lo antes dicho, podemos calcular $a(S', j+1, p)$, dados los valores de $a(S, j, p')$ para $S \subseteq S'$ y $p' \leq p$.

Una vez calculada a , tenemos que los valores de p para los cuales $a(S, q, p) = 1$ son los valores para los cuales existe una p -partición convexa de $\hat{G}_q = G$ tal que no corta a ningún V_c , $c \in C$, y tal que los cliques V_c , $c \in S$, quedaron fijos. Si solo nos interesaran las particiones convexas de G que no cortan a los cliques, entonces no sería necesario almacenar la información de cuales cliques quedaron fijos y solo bastaría calcular una sola función $a(j, p)$, que no depende de S . Sin embargo, esto se hace necesario al querer encontrar particiones que si cortan a los cliques V_c , $c \in C$, como veremos a continuación.

Para encontrar los tamaños de las particiones convexas que faltan, notemos que una partición convexa de G que puede cortar a los cliques V_c , $c \in C \setminus S$ se puede construir a partir de las particiones antes calculadas. Así, si $a(S, q, p) = 1$, entonces G tiene una p' -partición convexa, para $p' \in [p, p + \sum_{c \in C \setminus S} |V_c|]$.

Ahora veamos la complejidad del procedimiento anterior. Primero, tenemos que la función a es una tabla con $O(2^q q |V(G)|)$ entradas. Por cada valor 1 en la tabla, tenemos que actualizar $O(|V(G)|)$ valores. Además, por cada H_j se realiza un cálculo de su partición convexa mínima y $O(|V(G)|)$ llamadas al oráculo LOCCONV, si es el caso correspondiente. Por lo tanto, el procedimiento anterior toma tiempo

$$O\left(2^q q |V(G)|^2 + \sum_{i=1}^q |V(H_i)|^4 + \sum_{p'=1}^p \sum_{i=1}^q \text{LOCCONV}(H_i, p')\right).$$

Es decir, tenemos tiempo exponencial en q y pq llamadas al oráculo LOCCONV.

Conclusión

El principal foco de este trabajo fue el estudiar el problema de decidir si un grafo tiene una p -partición convexa. En general, este problema es NP-completo, pero al restringirse a familias particulares de grafos, se obtienen algoritmos polinomiales. En este caso, se trabajó sobre una familia de grafos definida recursivamente, reemplazando los vértices de un bosque por grafos de la misma familia, que llamamos \mathcal{T}^* .

Lo que inspiró a trabajar con esta familia es la idea de recuperar la programación dinámica que es posible realizar cuando se trabaja con los árboles. Así, basta calcular las particiones convexas en subárboles que no contienen cliques en sus vértices internos y después combinarlas. En este trabajo, vemos que las particiones convexas de \mathcal{T}^* están bastante cerca de ser particiones en cliques, salvo por el caso en el que consideramos una suma de un clique y un no clique. En tal caso, aparece un tipo de conjunto nuevo: los conjuntos localmente convexas. La dificultad al atacar el problema que surge con estos conjuntos transforma a los cliques en vértices internos del árbol subyacente, que parecían la oportunidad de poder simplificar el problema, en uno de los mayores problemas al estudiar los convexas en \mathcal{T}^* . Aún ignorando el problema que presentan las particiones localmente convexas, la estrategia de combinar las particiones convexas no resulta muy efectiva en esta familia, pues es necesario almacenar cuáles de los cliques en vértices internos del árbol subyacente se pueden cortar en un refinamiento de la partición, lo que obliga a manejar una cantidad exponencial de información para computar el problema en el grafo original.

Para ubicar a \mathcal{T}^* en un contexto más general, se estudiaron algunas propiedades estructurales de esta familia. Así, esta resulta ser cerrada para subgrafos inducidos. Por lo tanto, esta se puede caracterizar por una familia de subgrafos prohibidos. El mejor resultado al que se llegó en este trabajo, fue determinar que éstos son HOUSE-HOLE-BULL-A-free, pero se desconoce si estos subgrafos alcanzan para caracterizar a \mathcal{T}^* . Probablemente, un análisis más fino de los casos problemáticos que surgen en el algoritmo de reconocimiento baste para determinar los grafos faltantes, si es que existen. Además, se encontraron algunas inclusiones interesantes, como que los grafos de \mathcal{T}^* son perfectos o que los cografos están en \mathcal{T}^* . Esto resuelve algunas preguntas en \mathcal{T}^* , gracias a lo que se conoce sobre los grafos perfectos, y también da lugar a otras, como qué técnicas de los cografos se pueden recuperar en \mathcal{T}^* . Por ejemplo, problemas como el de ciclo hamiltoniano se resuelven en tiempo polinomial en los cografos, subfamilia de \mathcal{T}^* , pero son NP-completo en los grafos perfectos, superfamilia de \mathcal{T}^* , aparecen como problemas interesantes de estudiar en esta familia.

Bibliografía

- [1] D. Artigas, S. Dantas, M.C. Dourado, and J.L. Szwarcfiter. Partitioning a graph into convex sets. *Discrete Mathematics*, 311(17):1968 – 1977, 2011.
- [2] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182 – 208, 1986.
- [3] Andreas Brandstädt and Van Bang Le. Structure and linear time recognition of 3-leaf powers. *Inf. Process. Lett.*, 98(4):133–138, May 2006.
- [4] Andreas Brandstädt, Van Bang Le, and R. Sriharan. Structure and linear-time recognition of 4-leaf powers. *ACM Trans. Algorithms*, 5(1):11:1–11:22, December 2008.
- [5] Andreas Brandstädt, Feodor F. Dragan, and Falk Nicolai. Homogeneously orderable graphs. *Theoretical Computer Science*, 172(1):209 – 232, 1997.
- [6] R. Sergio Canoy, Jr. and I.J.L. Garces. Convex sets under some graph operations. *Graphs and Combinatorics*, 18(4):787–793, 2002.
- [7] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.
- [8] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 3rd edition, 2009.
- [9] D.G. Corneil, H. Lerchs, and L.Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163 – 174, 1981.
- [10] D.G. Corneil, H. Lerchs, and L.Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163 – 174, 1981.
- [11] Martin Farber and Robert E Jamison. On local convexity in graphs. *Discrete Mathematics*, 66(3):231 – 247, 1987.
- [12] Martin Charles Golumbic. Algorithmic graph theory and perfect graphs. Computer Science and Applied Mathematics; New York: Academic Press., 1980.
- [13] Luciano N. Grippo, Martín Matamala, Martín D. Safe, and Maya J. Stein. Convex p-partitions of bipartite graphs. *Theor. Comput. Sci.*, 609(P2):511–514, January 2016.

- [14] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 6 1981.
- [15] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325 – 356. North-Holland, 1984.
- [16] Frank Gurski and Egon Wanke. *On Module-Composed Graphs*, pages 166–177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [17] Ryan B. Hayward, Jeremy P. Spinrad, and R. Sritharan. Improved algorithms for weakly chordal graphs. *ACM Trans. Algorithms*, 3(2), May 2007.
- [18] Edward Howorka. A characterization of distance-hereditary graphs. *The Quarterly Journal of Mathematics*, 28(4):417–420, 1977.
- [19] Mamadou Moustapha Kanté and Lhouari Nourine. *Polynomial Time Algorithms for Computing a Minimum Hull Set in Distance-Hereditary and Chordal Graphs*, pages 268–279. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [20] Jeremy Spinrad and R. Sritharan. Algorithms for weakly triangulated graphs. *Discrete Applied Mathematics*, 59(2):181 – 191, 1995.