



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Sobre la thinness en un grafo

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Gastón Abel Brito

Directora: Flavia Bonomo

Buenos Aires, 2020

SOBRE LA THINNESS EN UN GRAFO

La thinness de un grafo, definida inicialmente en 2007, es un invariante que generaliza a los grafos de intervalos. Todo grafo tiene un valor numérico de thinness y los grafos con thinness 1 coinciden con los grafos de intervalos.

Un grafo es k -thin si sus vértices pueden ordenarse de manera que exista una partición de los vértices en k clases cumpliendo que para cada tripla de vértices $r < s < t$ si r y s pertenecen a la misma clase y existe una arista entre r y t , entonces existe una arista entre s y t . La thinness es el menor valor de k tal que el grafo sea k -thin.

En este trabajo definimos un modelo de intersección para los grafos con thinness y thinness propia menor o igual a 2 restringiendo los grafos con boxicity 2. Se caracterizan los grafos k -thin y k -thin propios utilizando propiedades de la matriz de adyacencia. Se acota la thinness en relación al bandwidth de un grafo y se hace un resumen de las relaciones conocidas.

Se calcula la thinness utilizando SAT Module Theory (SMT), tanto para determinar si vale la propiedad k -thin como para obtener el valor de la thinness mediante optimización. Se evalúa la performance sobre familias de grafos con los resolvers Z3 Theorem Solver y OptiMathSat. Adicionalmente se utilizan las mismas técnicas para calcular una representación como una intersección de rectángulos en el plano.

Palabras claves: Grafo, Thinness, Intervalo, Boxicity, Intersección, Bandwidth, SMT, Z3

ON THE THINNESS OF A GRAPH

The thinness of a graph, initially defined in 2007, is an invariant that generalises the concept of an interval graph. Every graph has a numeric value of thinness and the graphs with thinness 1 are exactly the interval graphs.

A graph is k -thin if its vertices can be sorted in a way that exists a partition with k classes with the property that for each triplet of vertices $r < s < t$ if r and s belong to the same class and there is an edge between r y t , then an edge must exist between s and t . The thinness is the least value of k such that the graph is k -thin.

In this work we define an intersection model for graphs with thinness and proper thinness lower or equal than 2 by restricting graphs with boxicity 2. The properties of adjacency matrices are used to characterize k -thin and proper k -thin graphs. A bound of the thinness with the bandwidth is obtained and a summary of the know relations is made.

It is shown how to calculate the thinness with SAT Module Theory (SMT), both for checking the k -thin property and obtaining the thinness value with optimization. The performance of the Z3 Theorem Solver and OptiMathSat are analyzed. Additionally the same techniques are used to obtain a representation as an intersection of rectangles on the plane.

Keywords: Graph, Thinness, Interval, Boxicity, Intersection, Bandwidth, SMT, Z3

Índice general

1..	Introducción	1
1.1.	Definiciones básicas	3
1.2.	Notación	4
2..	Representación	5
2.1.	Representación de grafos de intervalos	5
2.2.	Representación de grafos 2-thin	7
3..	Modelo de intersección para 2-thin	9
3.1.	Boxicity	9
3.2.	Diagonales	13
3.3.	Ejemplos	17
4..	Modelo de intersección para 2-thin propio	20
5..	Matrices	24
5.1.	Permutaciones de matrices de adyacencia	25
5.2.	Propiedad C1D	26
5.3.	Grafos k-thin y C1D	27
5.4.	Propiedad C1L	28
5.5.	Grafos de k-thin propios	29
5.6.	NP-Complejidad	30
6..	Métricas	31
6.1.	Bandwidth	31
6.2.	Boxicity	32
6.3.	Grafos planares	32
6.4.	Resultados conocidos	33
7..	Reconocimiento	34
7.1.	Ejemplo	34
7.2.	Instancias para calcular thinness	36
7.3.	Instancias para boxicity 2	39
7.4.	Performance	41
8..	Conclusiones y trabajo futuro	44
	Apéndice	45
1.	Contraejemplo diagonal con thinness 3	46
2.	Instancia SMT para calcular thinness de C4	49

1. INTRODUCCIÓN

La thinness es una generalización de los grafos de intervalos, se comienza enunciando las definiciones relacionadas.

Definición 1.1 Un grafo $G = (V, E)$ es de intersección si a cada vértice $v_i \in V$ se le puede asignar un conjunto S_i de manera que $v_i v_j \in E \iff S_i \cap S_j \neq \emptyset$.

Definición 1.2 Un grafo de intervalos es un grafo de intersección de un conjunto de intervalos en \mathbb{R} . Es decir $S_i = [a_i, b_i]$ con a_i y b_i números reales y $a_i < b_i$.

Una caracterización posible de estos grafos es mediante el siguiente teorema.

Teorema 1.3 [2] Un grafo $G = (V, E)$ es de intervalos si existe un ordenamiento v_1, v_2, \dots, v_n de V tal que para cada tripla (r, s, t) con $r < s < t$ si $v_t v_r \in E$ entonces $v_t v_s \in E$.

La noción de grafos de intervalos se generalizó en [1] mediante la introducción de la noción de thinness.

Definición 1.4 [1] Un grafo $G = (V, E)$ es k -thin si existe un ordenamiento v_1, v_2, \dots, v_n de V y un particionamiento V^1, \dots, V^k de V en k clases tal que para cada tripla (r, s, t) con $r < s < t$ si v_r, v_s se encuentran en la misma clase y $v_t v_r \in E$ entonces $v_t v_s \in E$

Observar que un grafo es de intervalos si y sólo si es 1-thin. La thinness de un grafo G es el menor número k para el cual G es un grafo k -thin, y lo notamos $thin(G)$. Cuando un ordenamiento satisface la propiedad definida en 1.4 decimos que es un ordenamiento consistente con la partición.

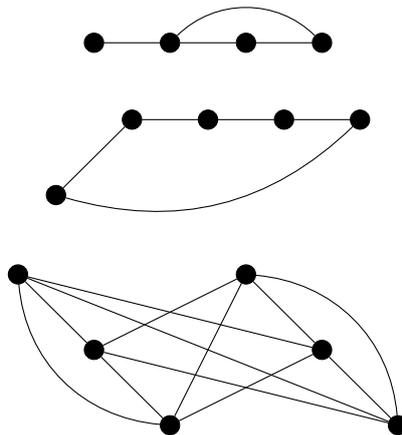


Fig. 1.1: Ejemplos de grafos con thinness 1,2 y 3.

Definición 1.5 Un grafo es de intervalo propio si existe un ordenamiento de los vértices v_1, v_2, \dots, v_n tal para cada tripla (r, s, t) con $r < s < t$ si $v_t v_r \in E$, entonces $v_t v_s \in E$ y $v_r v_s \in E$

Definición 1.6 [3] Un grafo es k -thin propio si existe un ordenamiento de los vértices v_1, v_2, \dots, v_n y una partición V^1, \dots, V^k de V en k clases tal para cada tripla (r, s, t) con $r < s < t$ se cumple que:

- Si v_r, v_s pertenecen a la misma clase y $v_t v_r \in E$, entonces $v_t v_s \in E$
- Si v_s, v_t pertenecen a la misma clase y $v_t v_r \in E$, entonces $v_r v_s \in E$

Las dos definiciones anteriores implican que el ordenamiento reverso v_n, v_{n-1}, \dots, v_1 también es consistente con la partición. La thinness propia de un grafo G es el menor número k para el que cumple la definición de k -thin propio. Se denota como $pthin(G)$. Cuando un ordenamiento verifica la definición de k -thin propio decimos que es un ordenamiento fuertemente consistente. Notemos que si un grafo es k -thin propio, entonces es k -thin. De esta observación se desprende la siguiente relación: $thin(G) \leq pthin(G)$.

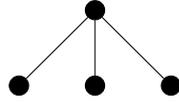


Fig. 1.2: Grafo G con $thin(G) = 1$ y $pthin(G) = 2$

Teorema 1.7 [3] Dado un grafo G y un ordenamiento de $V(G)$ se pueden construir en tiempo polinomial grafos $G_{<}$ y $\tilde{G}_{<}$ con las siguientes propiedades:

- $V(G) = V(G_{<}) = V(\tilde{G}_{<})$
- El número cromático de $G_{<}$ es igual al menor entero k tal que existe una partición de $V(G)$ en k conjuntos que son consistentes con el orden $<$ y las clases de color de un coloreo válido de $G_{<}$ forman una partición consistente con $<$
- El número cromático de $\tilde{G}_{<}$ es igual al menor entero k tal que existe una partición de $V(G)$ en k conjuntos que son fuertemente consistentes con el orden $<$ y las clases de color de un coloreo válido de $\tilde{G}_{<}$ forman una partición fuertemente consistente con $<$
- $G_{<}$ y $\tilde{G}_{<}$ son grafos de co-comparabilidad

Las aristas de cada grafo se generan de la siguiente manera:

$$v < w, \quad vw \in E(G_{<}) \iff \exists z : v < w < z \wedge zv \in E(G) \wedge zw \notin E(G)$$

$$v < w, \quad vw \in E(\tilde{G}_{<}) \iff (\exists z : v < w < z \wedge zv \in E(G) \wedge zw \notin E(G)) \vee (\exists x : x < v < w \wedge xw \in E(G) \wedge xv \notin E(G))$$

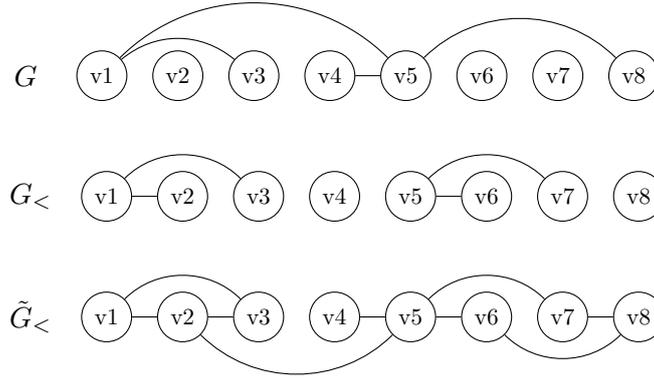


Fig. 1.3: Ejemplos de grafos \tilde{G} y $\tilde{G}_<$ utilizando el orden $v_1 < v_2 < v_3 < v_4 < v_5 < v_6 < v_7 < v_8$. Cuando se agrega o elimina una arista la thinness puede incrementarse o decrementarse.

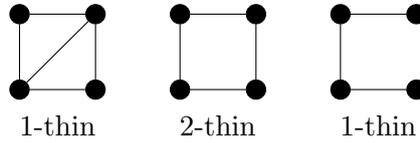


Fig. 1.4: Ejemplos de como cambia la thinness al agregar o remover una arista.

Al remover una arista, la thinness puede incrementarse, pero nunca por más de 1 unidad.

Teorema 1.8 Dado un grafo G y $vw \in E(G)$, $\text{thin}(G - vw) \leq 1 + \text{thin}(G)$

Demostración. Sea $k = \text{thin}(G)$ y v_1, v_2, \dots, v_n un ordenamiento de $V(G)$ consistente con la partición en las clases V^1, \dots, V^k . Se asume sin pérdida de generalidad que $v < w$.

Se ubica a v en la clase $V^{k+1} = \{v\}$. □

Determinar la thinness de un grafo y un orden y partición consistentes es un problema útil debido a que si en un grafo la thinness es acotada algunos problemas NP-Completos pueden resolverse en tiempo polinomial, dada la correspondiente representación. Un ejemplo es el conjunto independiente con pesos en los vértices [1].

1.1. Definiciones básicas

En este trabajo todos los grafos son simples sin bucles, siendo un bucle una arista con sus dos extremos en el mismo vértice. Se utiliza $V(G)$ para denotar el conjunto de vértices y $E(G)$ para el conjunto de aristas. Cualquier concepto no definido aquí puede consultarse en [5, 3].

El grado de un vértice es la cantidad de aristas incidentes que posee. El grado máximo de un grafo G es el mayor grado entre todos sus vértices y se nota como $\Delta(G)$.

Los vértices adyacentes a un vértice se conocen como los vecinos del vértice y se notan como $N(v)$. Se define $N[v] = N(v) \cup \{v\}$. Decimos que los vértices v, w son mellizos verdaderos si $N[u] = N[v]$.

El grafo $C_n = (\{v_1, v_2, \dots, v_n\}, \{v_1v_n\} \cup \{v_iv_{i+1} : 1 \leq i < n\})$ es el ciclo de n vértices, con $n \geq 3$.

El complemento de un grafo $G = (V, E)$ es el grafo $\overline{G} = (V, E')$, en donde $uv \in E' \iff uv \notin E$.

Un clique es un conjunto $X \subseteq V(G)$ tal que todo par de vértices de X tiene una arista entre ellos.

Un clique maximal es un clique X tal que $X \cup \{v\}$ no es un clique para todo $v \in V(G) - X$.

Un camino entre 2 vértices u, v es una secuencia de vértices v_1, v_2, \dots, v_k tal que $v_1 = u$, $v_k = v$ y existe una arista $v_i v_{i+1}$ si $1 \leq i < k$, estas son sus únicas aristas y no existen vértices repetidos en la secuencia. Un grafo es conexo si existe un camino entre todo par de vértices.

Un grafo orientado es un grafo $D = (V, E)$ en donde E es un subconjunto de pares ordenados de elementos de V .

Un ciclo en un grafo orientado $D = (V, E)$ es un ordenamiento v_1, v_2, \dots, v_n de un conjunto $W \subseteq V$ con $n = |W|$ de manera que $v_n v_1 \in E$ y $v_i v_{i+1} \in E$ para $1 \leq i < n$ y estos son sus únicos pares ordenados.

Una cuerda de un ciclo v_1, v_2, \dots, v_n es una arista entre 2 vértices del ciclo que no pertenece al conjunto $\{v_n v_1\} \cup \{v_i v_{i+1} : 1 \leq i < n\}$.

Un orden topológico de un grafo orientado es un ordenamiento de los vértices de manera que para todas las aristas uv se cumpla que u se encuentre antes que v en el ordenamiento.

Un grafo orientado es acíclico si no tiene ciclo dirigidos. Un grafo orientado tiene un orden topológico si y sólo si es acíclico [6].

El grafo K_n es el grafo completo de n vértices en donde todos los pares de vértices son adyacentes entre sí. \mathbb{R} es el conjunto de los números reales.

1.2. Notación

Cuando es necesario conocer el orden y/o la partición de los vértices, los grafos se dibujan de manera que se pueda saber el orden exacto mirando el gráfico de izquierda a derecha y de arriba hacia abajo. La partición se determina construyendo una clase por cada línea horizontal de vértices. Por ejemplo, en el siguiente grafo el orden es $a < b < c < d < e < f$ y la partición $V^1 = \{c, e\}$ $V^2 = \{a, b, d, f\}$

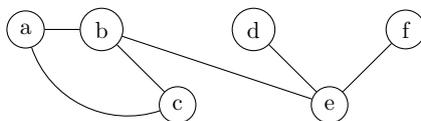


Fig. 1.5: Dibujo de grafo con orden y particionamiento implícito

2. REPRESENTACIÓN

En esta sección mostraremos una forma compacta de representar grafos de intervalos y grafos con thinness mayor a 1. Una motivación es que esto puede ayudar a enfocarse en lo esencial de cada familia de grafos.

Adicionalmente se van a utilizar estas representaciones para pasar grafos como parámetros a programas que implementan las reducciones del capítulo de reconocimiento.

2.1. Representación de grafos de intervalos

Los grafos de intervalos son las intersecciones de intervalos en \mathbb{R} y tienen varias caracterizaciones. Antes de intentar representar los grafos 2-thin se busca una manera compacta de almacenar los grafos de intervalos.

Un grafo de intervalos de n vértices puede representarse como una secuencia de n números enteros no negativos a_1, a_2, \dots, a_n con $0 \leq a_i \leq i - 1$. Cada elemento de la secuencia representa un vértice en el orden y la cantidad de vértices inmediatamente anteriores adyacentes a él.

La secuencia a_1, a_2, \dots, a_n con $0 \leq a_i \leq i - 1$ representa al grafo $G = (\{v_1, v_2, \dots, v_n\}, E)$ donde dado $i < j$, $v_i v_j \in E \iff a_j \geq j - i$

0	
0, 0	
0, 1	
0, 0, 0	
0, 0, 1	
0, 0, 2	
0, 1, 0	
0, 1, 1	
0, 1, 2	
0, 1, 0, 3	

Tab. 2.1: Ejemplos de grafos de intervalos generados desde una secuencia.

Lo bueno de esta representación es que se pueden generar todos los grafos de intervalos posibles y no hay manera de generar uno que no sea de intervalos.

Teorema 2.1 Un grafo $G = (V, E)$ es de intervalos si y sólo si existe una secuencia de números enteros no negativos $a_1, a_2, \dots, a_{|V|}$ con $0 \leq a_i \leq i - 1$ de manera que $H = (\{w_1, w_2, \dots, w_{|V|}\}, E')$ es isomorfo a G .

Demostración. \Leftarrow) Sea a_1, a_2, \dots, a_n una secuencia con $0 \leq a_i \leq i - 1$. Se construye $H = (V, E)$ con E tal que dado $i < j$, $(w_i, w_j) \in E \iff a_j \geq j - i$. Para probar que H es un grafo de intervalos se verifica que para cada tupla (r, s, t) con $1 \leq r < s < t \leq |V|$ si $(v_t, v_r) \in E$ se cumple que $(v_t, v_s) \in E$. Sea (r, s, t) tal que $1 \leq r < s < t \leq |V|$ y $(v_t, v_r) \in E$. Por definición se sabe que $a_t \geq t - r$. Como $r < s$ entonces $a_t \geq t - s$ y por lo tanto $(v_t, v_s) \in E$.

\Rightarrow) Sea $G = (V, E)$ un grafo de intervalos. Entonces existe un ordenamiento $v_1, v_2, \dots, v_{|V|}$ de V de manera que para cada tupla (r, s, t) con $1 \leq r < s < t \leq |V|$ si $(v_t, v_r) \in E$ se cumple que $(v_t, v_s) \in E$. Usar $a_i = i - \min(\{j : v_j \in N(v_i)\} \cup \{i\})$. \square

Teorema 2.2 La secuencia $a_1, a_2, \dots, a_{|V|}$ del teorema 2.1 corresponde al grafo de intersección de los intervalos $\{[2(i - a_i), 2i + 1] : 1 \leq i \leq |V|\}$.

Demostración. Suponemos que existe una arista (w_i, w_j) , $i < j$ y $a_j \geq j - i$. Se comprueba que existe intersección entre los intervalos $[2(i - a_i), 2i + 1]$ y $[2(j - a_j), 2j + 1]$ dado que

$$i < j \implies 2i + 1 < 2j + 1$$

$$a_j \geq j - i \implies 2a_j \geq 2j - 2i \implies 2i \geq 2j - 2a_j \implies 2i + 1 > 2(j - a_j)$$

Recíprocamente probamos que dados 2 intervalos con intersección se genera la condición necesaria en la secuencia. Sea $i < j$ y $[2(i - a_i), 2i + 1] \cap [2(j - a_j), 2j + 1] \neq \emptyset$. Como se tiene que $2i + 1 < 2j + 1$ entonces tiene que ser el caso que $2(j - a_j) < 2i + 1$ y

$$2(j - a_j) < 2i + 1 \implies 2j - 2a_j < 2i + 1 \implies 2j - 2i < 2a_j + 1 \implies 2j - 2i \leq 2a_j \implies j - i \leq a_j$$

\square

En [4] se usa una construcción similar para generar los intervalos. Este teorema es importante porque genera en forma concreta los intervalos a partir de la secuencia.

Teorema 2.3 Sea $G = (V, E)$ un grafo de intervalos con la representación $a_1, a_2, \dots, a_{|V|}$. Entonces $\sum_i^{|V|} a_i = |E|$.

Demostración. Por inducción en la cantidad de elementos de la secuencia. Caso 1, hay un solo nodo en el grafo, que no tiene aristas y $a_1 = 0$ por definición. Caso $n + 1$, asumimos $G = (V, E)$ con $n = |V|$ y $\sum_1^n a_i = |E|$. Entonces agregando a la secuencia el elemento a_{n+1} se genera el grafo $G' = (V', E')$ con $|V'| = |V| + 1$ y $|E'| = |E| + a_{n+1}$. Por lo tanto $\sum_1^{n+1} a_i = a_{n+1} + \sum_1^n a_i = a_{n+1} + |E| = |E'|$. \square

2.2. Representación de grafos 2-thin

Para los grafos 2-thin de n nodos se puede usar una secuencia de n tuplas de números enteros no negativos $(z_1, a_1, b_1), (z_2, a_2, b_2), \dots, (z_n, a_n, b_n)$ con $z_i \in \{1, 2\}$, $0 \leq a_i \leq TOT(1, i-1)$, $0 \leq b_i \leq TOT(2, i-1)$ siendo $TOT(z, i) = |\{(z_k, a_k, b_k) : 1 \leq k \leq i \wedge z_k = z\}|$. El primer elemento de cada tupla es el número de clase al que pertenece el nodo. El segundo elemento es el número vértices inmediatamente anteriores adyacentes en la primer clase, mientras que el tercero es el número vértices inmediatamente anteriores adyacentes en la segunda clase.

$(1, 0, 0), (1, 0, 0)$	
$(1, 0, 0), (2, 0, 0)$	
$(1, 0, 0), (2, 1, 0)$	
$(1, 0, 0), (2, 1, 0), (1, 1, 0)$	
$(1, 0, 0), (2, 1, 0), (1, 1, 0), (2, 1, 1)$	
$(1, 0, 0), (1, 0, 0), (2, 2, 0), (2, 2, 0)$	
$(2, 0, 0), (2, 0, 0), (2, 0, 0), (1, 0, 3)$	

Tab. 2.2: Ejemplos de grafos 2-thin generados desde una secuencia.

Teorema 2.6 Un grafo $G = (V, E)$ es 2-thin si y sólo si existe una secuencia de tuplas de números enteros no negativos $(z_1, a_1, b_1), (z_2, a_2, b_2), \dots, (z_{|V|}, a_{|V|}, b_{|V|})$ con $z_i \in \{1, 2\}$, $0 \leq a_i \leq TOT(1, i-1)$, $0 \leq b_i \leq TOT(2, i-1)$ de manera que $H = (\{w_1, w_2, \dots, w_{|V|}\}, E')$ es isomorfo a G . Siendo E' dado $i < j$,

$$(w_i, w_j) \in E' \iff \left(z_i = 1 \wedge a_j \geq TOT(1, j-1) - TOT(1, i-1) \right) \vee \left(z_i = 2 \wedge b_j \geq TOT(2, j-1) - TOT(2, i-1) \right)$$

$$TOT(z, i) = |\{(z_k, a_k, b_k) : 1 \leq k \leq i \wedge z_k = z\}|$$

Demostración. \Leftarrow) Sea $(z_1, a_1, b_1), (z_2, a_2, b_2), \dots, (z_n, a_n, b_n)$ una secuencia con $z_i \in \{1, 2\}$, $0 \leq a_i \leq TOT(1, i-1)$, $0 \leq b_i \leq TOT(2, i-1)$. Se construye $H = (V, E)$ como se indica. Para probar que H es un grafo 2-thin se verifica que para cada tupla (r, s, t) con $1 \leq r < s < t \leq |V|$, v_r, v_s se encuentran en la misma clase y $(v_t, v_r) \in E$ se cumple que $(v_t, v_s) \in E$. Sea (r, s, t) tal que $1 \leq r < s < t \leq |V|$, v_r, v_s en la misma clase y $(v_t, v_r) \in E$. Se ve el caso $z_r = 1$. Por definición se sabe que $a_t \geq TOT(1, t-1) - TOT(1, r-1)$. Como $r < s$ entonces $TOT(1, r-1) \leq TOT(1, s-1)$ y por lo tanto $a_t \geq TOT(1, t-1) - TOT(1, s-1)$. De esta manera $(v_t, v_s) \in E$. El caso $z_r = 2$ es similar.

\Rightarrow) Sea $G = (V, E)$ un grafo 2-thin. Entonces existe un ordenamiento $v_1, v_2, \dots, v_{|V|}$ de V y una partición V^1, V^2 de V de manera que para cada tupla (r, s, t) con $1 \leq r < s < t \leq |V|$, v_r, v_s en la misma clase y $(v_t, v_r) \in E$ se cumple que $(v_t, v_s) \in E$. Utilizar

$$z_i = 1 \text{ si } v_i \in V^1 \text{ o } z_i = 2 \text{ si } v_i \in V^2$$

$$a_i = TOT(1, i-1) - \min\left(\{TOT(1, j-1) : v_j \in N(v_i) \cap V^1\} \cup \{TOT(1, i-1)\}\right)$$

$$b_i = TOT(2, i-1) - \min\left(\{TOT(2, j-1) : v_j \in N(v_i) \cap V^2\} \cup \{TOT(2, i-1)\}\right)$$

□

Teorema 2.7 Sea $G = (V, E)$ un grafo 2-thin con la representación

$(z_1, a_1, b_1), (z_2, a_2, b_2), \dots, (z_{|V|}, a_{|V|}, b_{|V|})$. Entonces $\sum_i^{|V|} a_i + b_i = |E|$.

Demostración. Por inducción en la cantidad de elementos de la secuencia. Caso 1, hay un solo nodo en el grafo, que no tiene aristas y $a_1 = b_1 = 0$ por definición. Caso $n+1$, asumimos $G = (V, E)$ con $n = |V|$ y $\sum_1^n a_i + b_i = |E|$. Entonces agregando a la secuencia el elemento $(z_{n+1}, a_{n+1}, b_{n+1})$ se genera el grafo $G' = (V', E')$ con $|V'| = |V| + 1$ y $|E'| = |E| + a_{n+1} + b_{n+1}$. Por lo tanto $\sum_1^{n+1} a_i + b_i = a_{n+1} + b_{n+1} + \sum_1^n a_i + b_i = a_{n+1} + b_{n+1} + |E| = |E'|$. □

Corolario 2.8 Todo prefijo de al menos un elemento de una secuencia representa un grafo 2-thin.

Notar que esto se puede generalizar a grafos k -thin, con tuplas de $k+1$ elementos.

3. MODELO DE INTERSECCIÓN PARA 2-THIN

Todo grafo puede representarse como un grafo de intersección, por ejemplo tomando

$$S_i = \{\{v_r, v_s\} : v_r v_s \in E(G) \wedge (v_r = v_i \vee v_s = v_i)\}$$

Esto significa que es posible encontrar un grafo de intersección para grafos con thinness a lo sumo 2. Idealmente los conjuntos deberían ser sencillos de interpretar como en los grafos de intervalos. Una idea que puede pensarse es en utilizar 2 intervalos para cada vértice [8].

$$S_i = [a_i, b_i] \cup [c_i, d_i]$$

El problema de esto es que permite definir grafos con thinness 3 como el grafo del siguiente ejemplo, que se encuentra definido en [4].

$$\begin{aligned} S_1 &= [1, 3] \cup [8, 11] \\ S_2 &= [2, 5] \cup [14, 16] \\ S_3 &= [4, 7] \cup [21, 23] \\ S_4 &= [6, 9] \cup [26, 28] \\ S_5 &= [10, 13] \cup [29, 31] \\ S_6 &= [12, 15] \cup [17, 19] \\ S_7 &= [18, 20] \cup [22, 25] \\ S_8 &= [24, 27] \cup [30, 32] \end{aligned}$$

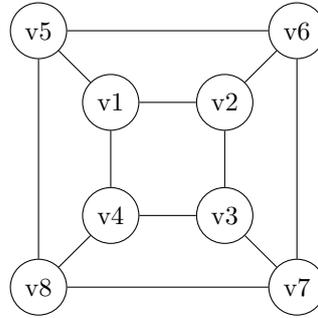
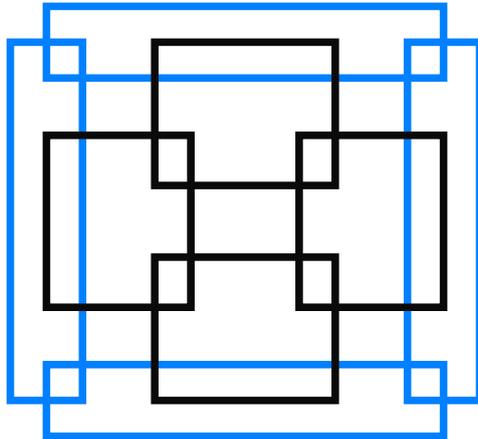


Fig. 3.1: Grafo con thinness 3.

3.1. Boxicity

Para construir el modelo de intersección utilizaremos el concepto de boxicity [15], que se define como la menor cantidad de dimensiones necesarias d para que el grafo sea la intersección de $|V(G)|$ cajas abiertas (“sin bordes”, se intersecan si comparten puntos interiores) con lados paralelos a los ejes Cartesianos en un espacio d -dimensional. El grafo anterior tiene boxicity 2:



Definimos las siguientes clases de grafos que son conjuntos de grafos que cumplen una determinada propiedad.

$$1\text{-THIN} = \{G : \text{thin}(G) = 1\}$$

$$2\text{-THIN} = \{G : \text{thin}(G) \leq 2\}$$

$$\text{BOXICITY-1} = \{G : \text{boxicity}(G) = 1\}$$

$$\text{BOXICITY-2} = \{G : \text{boxicity}(G) \leq 2\}$$

Teorema 3.1 [4] $\text{boxicity}(G) \leq \text{thin}(G)$

Dado el teorema anterior y teniendo en cuenta que $\text{BOXICITY-1} = 1\text{-THIN}$ se obtiene lo siguiente:

$$\text{BOXICITY-1} \subset 2\text{-THIN} \subset \text{BOXICITY-2}$$

La demostración utilizada en el teorema anterior para el caso $k = 2$ parte de un grafo 2-thin G con $V^1 = \{v_1^1, v_2^1, \dots, v_{p1}^1\}$, $V^2 = \{v_1^2, v_2^2, \dots, v_{p2}^2\}$ y construye 2 grafos de intervalos I^1, I^2 que cumplen las siguientes propiedades:

$$V(I^1) = V(I^2) = V(G)$$

$$G = I^1 \cap I^2$$

$$I^1[V^1] = G[V^1] \text{ e } I^2[V^2] = G[V^2]$$

$I^1[V^2]$ e $I^2[V^1]$ son grafos completos

Para todo $u \in V^2$ sea r el menor índice tal que $uv_r^1 \in E$. Entonces $uv_i^1 \in E^1$ para $i = r, \dots, p1$

Para todo $u \in V^1$ sea r el menor índice tal que $uv_r^2 \in E$. Entonces $uv_i^2 \in E^2$ para $i = r, \dots, p2$

En I^1 se utiliza el siguiente ordenamiento, que definiremos como una secuencia:

$$\Pi_1 = v_1^1, v_2^1, \dots, v_{p1}^1, v_1^2, v_2^2, \dots, v_{p2}^2$$

De la misma manera, definimos el ordenamiento en I^2 :

$$\Pi_2 = v_1^2, v_2^2, \dots, v_{p2}^2, v_1^1, v_2^1, \dots, v_{p1}^1$$

Utilizando la construcción anterior se definirá el grafo de intersección, pero previamente se definen los siguientes términos:

Definición 3.2 $\text{First}(G, v) =$ Menor elemento de $N[v]$ en G . Notar que $\text{First}(G, v) \leq v$

Definición 3.3 $\text{Pos}(\text{sec}, x) =$ La posición del elemento x en sec . Por ejemplo:

$$\text{Pos}([a, b, c], a) = 1$$

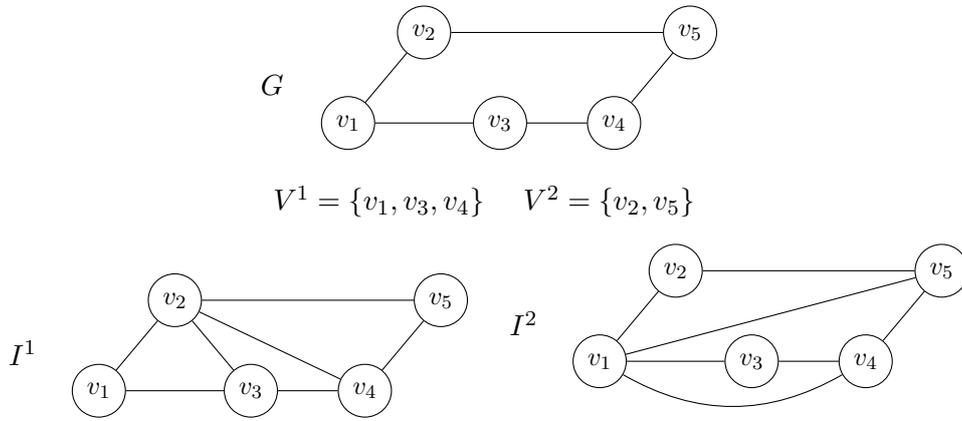
$$\text{Pos}([a, b, c], b) = 2$$

$$\text{Pos}([a, b, c], c) = 3$$

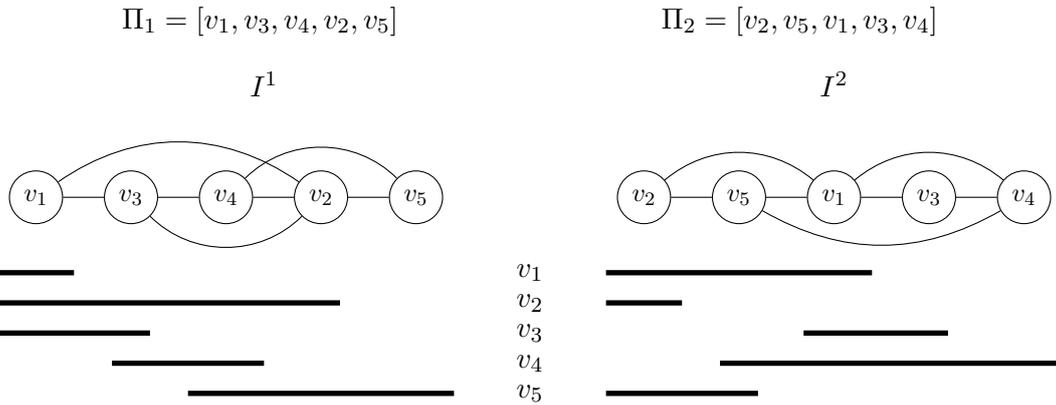
Dado un grafo 2-thin G y los grafos de intervalos I^1, I^2 generados en la demostración del Teorema 3.1, entonces puede generarse un modelo de intersección de cajas de la siguiente manera:

$$\{BOX(v) = (X_1 = Pos(\Pi_1, First(I^1, v)), \\ X_2 = Pos(\Pi_1, v) + 1, \\ Y_1 = Pos(\Pi_2, First(I^2, v)), \\ Y_2 = Pos(\Pi_2, v) + 1) : v \in V(G)\}$$

Se muestra un ejemplo.

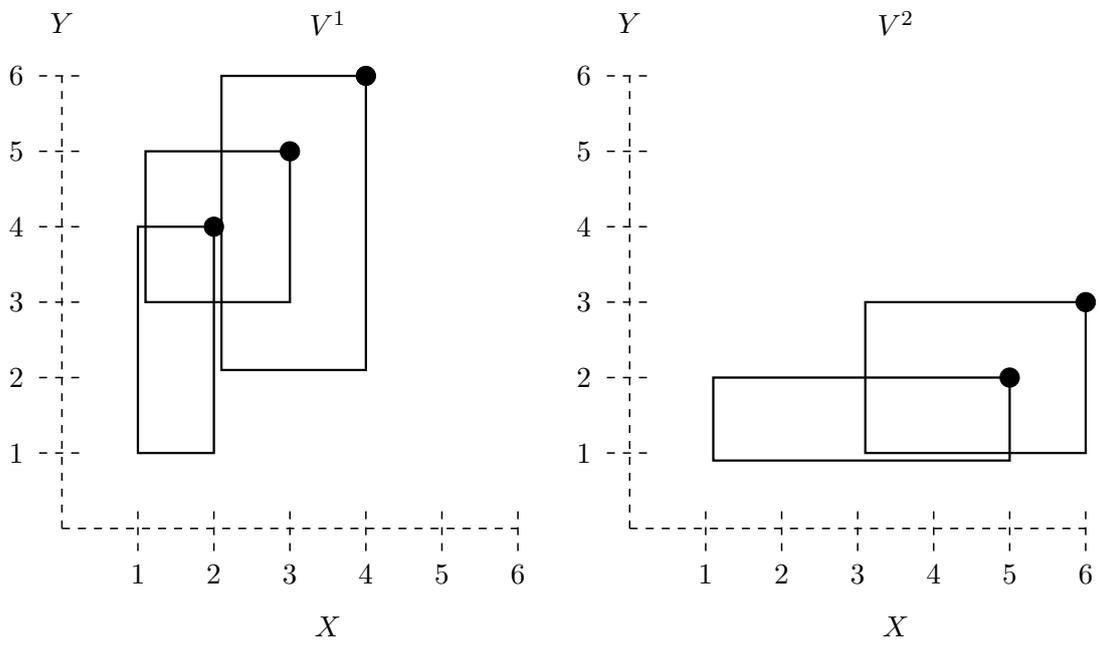


Se vuelven a dibujar los mismos grafos teniendo en cuenta el orden consistente de un grafo de intervalos.

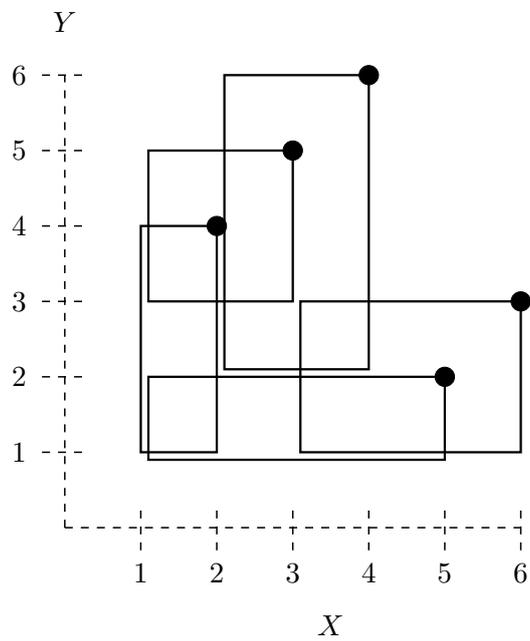


v	$Pos(\Pi_1, First(I^1, \Pi_1, v))$	$Pos(\Pi_1, v) + 1$	$Pos(\Pi_2, First(I^2, \Pi_2, v))$	$Pos(\Pi_2, v) + 1$
v_1	1	2	1	4
v_2	1	5	1	2
v_3	1	3	3	5
v_4	2	4	2	6
v_5	3	6	1	3

Con esta información las cajas quedan conformadas de la siguiente manera:



La representación completa es la siguiente:



3.2. Diagonales

Como se puede apreciar en los ejemplos, las esquinas superiores derechas de cada caja se encuentran ubicadas a lo largo de 2 líneas diagonales. Esto motiva la siguiente definición.

Definición 3.4 Un conjunto de cajas V es *diagonal* si existen dos números d_1 y d_2 tales que para cada caja definida por X_1, X_2, Y_1, Y_2 , siendo X_1, Y_1 la esquina inferior izquierda y X_2, Y_2 la esquina superior derecha, vale que o bien $Y_2 - X_2 = d_1$ o bien $Y_2 - X_2 = d_2$ y, si $V^i = \{v \in V : Y_2(v) - X_2(v) = d_i\}$, para $i = 1, 2$, se cumple que $\{(X_2(v), Y_2(v))\}_{v \in V^1} = \{(i + |V^2|, i)\}_{1 \leq i \leq |V^1|}$ y $\{(X_2(v), Y_2(v))\}_{v \in V^2} = \{(i, i + |V^1|)\}_{1 \leq i \leq |V^2|}$. Notar que $d_1 = -|V^2|$ y $d_2 = |V^1|$.

Lamentablemente la propiedad diagonal no alcanza para garantizar que un conjunto de cajas represente a un grafo 2-thin. Existen grafos con thinness 3 con un modelo de intersección de cajas diagonal, mostramos un ejemplo en el apéndice. Para remediar la situación se definen los conceptos necesarios para dar una construcción distinta de las cajas.

Definición 3.5 Dada una caja B se definen las prolongaciones de la caja B como las cajas $P_X(B)$ y $P_Y(B)$ que cumplen:

$$X_1(P_X(B)) = 0 \quad X_2(P_X(B)) = X_1(B) \quad Y_1(P_X(B)) = Y_1(B) \quad Y_2(P_X(B)) = Y_2(B)$$

$$X_1(P_Y(B)) = X_1(B) \quad X_2(P_Y(B)) = X_2(B) \quad Y_1(P_Y(B)) = 0 \quad Y_2(P_Y(B)) = Y_1(B)$$

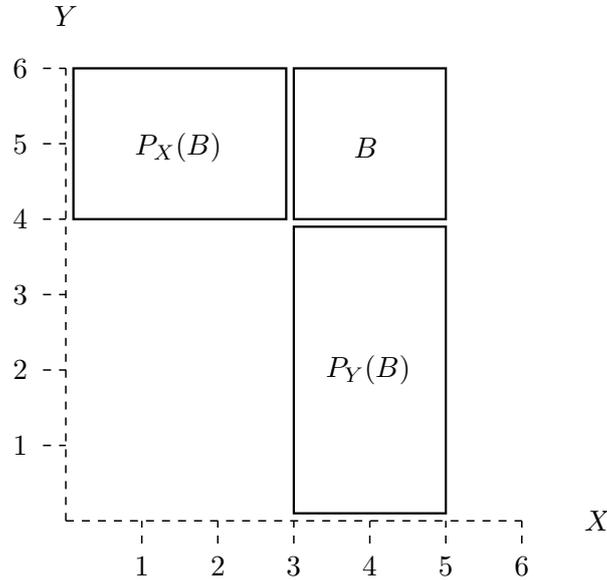


Fig. 3.2: Ejemplo de P_X y P_Y .

Definición 3.6 Un conjunto de cajas es *diagonal alineado* si es diagonal y para cada par de cajas $v \in V^1, w \in V^2$ se verifica que $v \cap w = \emptyset \implies (P_X(v) \cap w \neq \emptyset \vee v \cap P_Y(w) \neq \emptyset)$

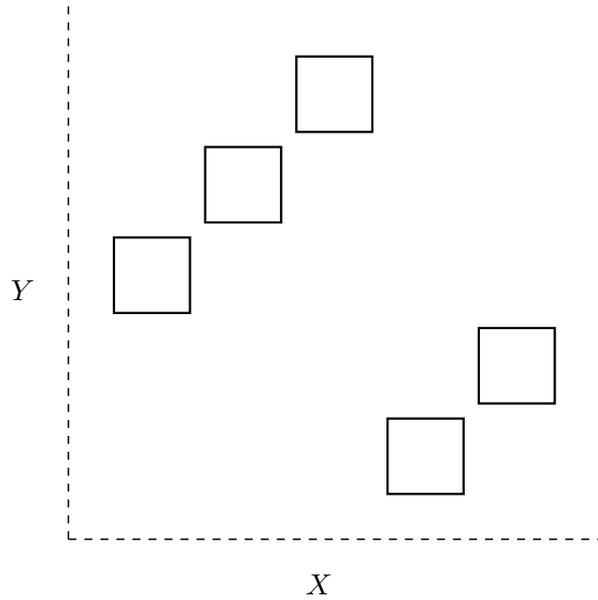


Fig. 3.3: Ejemplo de conjunto de cajas diagonal no alineado.

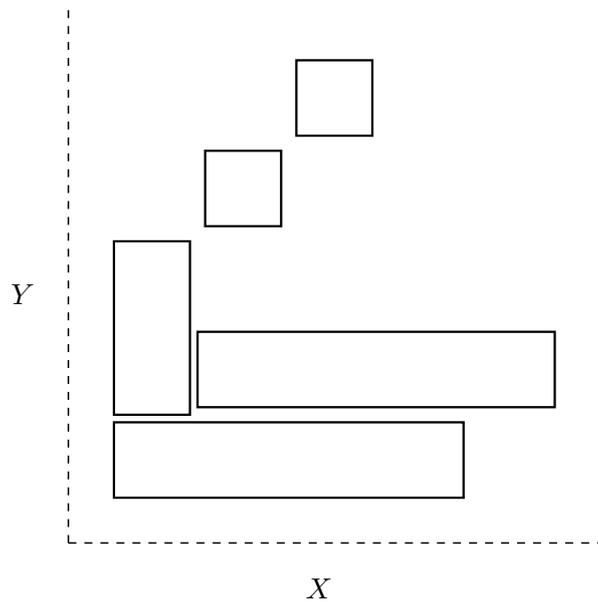


Fig. 3.4: Ejemplo de conjunto de cajas diagonal alineado.

Lema 3.7 Dado un conjunto de cajas C diagonal, C es diagonal alineado si y sólo si para todo $v \in V^1, w \in V^2$ vale que $X_1(v) < X_2(w)$ o $Y_1(w) < Y_2(v)$.

Notar que si $v \cap w \neq \emptyset$ entonces valen ambas.

Teorema 3.8 Un grafo es 2-thin si y sólo si tiene un modelo como intersección de cajas en el plano con lados paralelos a los ejes Cartesianos que es diagonal alineado.

Demostración. \Rightarrow) Sea $G = (V, E)$ un grafo 2-thin, con partición V^1, V^2 y orden consistente $<$. Sean $V^1 = v_1 < \dots < v_{n_1}$, $V^2 = w_1 < \dots < w_{n_2}$. Sean:

$$U(1, v_i) = \begin{cases} i & \text{Si } v_i \text{ no tiene vecinos menores que él en } V^1 \\ \text{mín}\{j : v_j < v_i, v_j v_i \in E(G)\} & \text{En caso contrario} \end{cases}$$

$$U(2, v_i) = \begin{cases} 0 & \text{Si } v_i \text{ es adyacente a todos} \\ & \text{los vértices de } V^2 \text{ menores que él} \\ \text{máx}\{j : w_j < v_i, w_j v_i \notin E(G)\} & \text{En caso contrario} \end{cases}$$

$$U(1, w_i) = \begin{cases} 0 & \text{Si } w_i \text{ es adyacente a todos} \\ & \text{los vértices de } V^1 \text{ menores que él} \\ \text{máx}\{j : v_j < w_i, v_j w_i \notin E(G)\} & \text{En caso contrario} \end{cases}$$

$$U(2, w_i) = \begin{cases} i & \text{Si } w_i \text{ no tiene vecinos menores que él en } V^2 \\ \text{mín}\{j : w_j < w_i, w_j w_i \in E(G)\} & \text{En caso contrario} \end{cases}$$

Definimos el siguiente modelo como intersección de cajas en el plano con los extremos:

$$\begin{aligned} v_i : \quad (X_2^i, Y_2^i) &= (i + n_2, i) \quad 1 \leq i \leq n_1 \\ (X_1^i, Y_1^i) &= (U(2, v_i) + 0.5, U(1, v_i) - 0.5) \quad 1 \leq i \leq n_1 \\ \\ w_i : \quad (X_2^i, Y_2^i) &= (i, i + n_1) \quad 1 \leq i \leq n_2 \\ (X_1^i, Y_1^i) &= (U(2, w_i) - 0.5, U(1, w_i) + 0.5) \quad 1 \leq i \leq n_2 \end{aligned}$$

Veamos que es un modelo para G y que es alineado (es claro que es diagonal y que respeta el orden relativo de cada clase).

Caso $v_i < v_j$ adyacentes: Entonces $U(1, v_j) \leq i$ y $U(2, v_j) \leq n_2$, por lo tanto las cajas de v_i y v_j se intersecan.

Caso $v_i < v_j$ no adyacentes: Por la consistencia entre partición y orden, $U(1, v_j) > i$, por lo tanto las cajas de v_i y v_j no se intersecan.

Caso $w_i < w_j$ adyacentes: Entonces $U(2, w_j) \leq i$ y $U(1, w_i) \leq n_1$, por lo tanto las cajas de w_i y w_j se intersecan.

Caso $w_i < w_j$ no adyacentes: Por la consistencia entre partición y orden, $U(2, w_j) > i$, por lo tanto las cajas de w_i y w_j no se intersecan.

Caso $v_i < w_j$: Entonces $U(2, v_i) < j$. Si son adyacentes, por la consistencia entre partición y orden, $U(1, w_j) < i$ y las cajas de v_i y w_j se intersecan. Si son no adyacentes, $U(1, w_j) \geq i$ y las cajas de v_i y w_j no se intersecan. En este caso, $P_Y(w_j) \cap v_i \neq \emptyset$.

Caso $w_i < v_j$: Entonces $U(1, w_i) < j$. Si son adyacentes, por la consistencia entre partición y orden, $U(2, v_j) < i$ y las cajas de v_i y w_j se intersecan. Si son no adyacentes, $U(2, v_j) \geq i$ y las cajas de v_i y w_j no se intersecan. En este caso, $P_X(v_j) \cap w_i \neq \emptyset$.

\Leftarrow) Consideremos un modelo de G como intersección de cajas con lados paralelos a los ejes Cartesianos que es diagonal alineado, y sean V^1 y V^2 como en la definición.

Vamos a definir un digrafo D con los mismos vértices que G y tal que cualquier orden topológico de D (si existe alguno) es consistente con la partición V^1, V^2 y extiende el orden de V^1 y V^2 definido por las componentes X_2 en cada uno de los conjuntos (aunque no necesariamente es el orden de las componentes X_2 en la unión de los conjuntos). Vamos a probar que ese digrafo es acíclico y por lo tanto admite un orden topológico. Para asegurar que extiende los órdenes de V^1 y V^2 , dados dos vértices v, v' en el mismo V^i , creamos el arco vv' en D si y sólo si $X_2(v) < X_2(v')$. Dados dos vértices $v_i \in V^i, v_{3-i} \in V^{3-i}$, creamos el arco $v_i v_{3-i}$ en D si y sólo si $v_i v_{3-i} \notin E(G)$ y existe $v'_{3-i} \in V^{3-i}$ con $X_2(v'_{3-i}) < X_2(v_{3-i})$ y $v_i v'_{3-i} \in E(G)$. En ese caso, notar que $P_X(v_{3-i}) \cap v_i \neq \emptyset$ o $P_Y(v_{3-i}) \cap v_i \neq \emptyset$.

Probemos primero que un orden topológico $<$ de D es consistente con V^1, V^2 . Sean $x < y < z$ con $xz \in E(G)$. Supongamos primero que $x, y, z \in V^1$ (las definiciones son simétricas con respecto a ambas clases). Entonces $X_2(x) < X_2(y) < X_2(z)$ y como $xz \in E(G)$, vale $X_1(z) < X_2(x) < X_2(y)$ e $Y_1(z) < Y_2(x) = X_2(x) + d_1 < X_2(y) + d_1 = Y_2(y)$. Por lo tanto $yz \in E(G)$. Ahora supongamos que $x, y \in V^1, z \in V^2$ (las definiciones son simétricas con respecto a ambas clases). Como $y < z$ en un orden topológico de D , $zy \notin A(D)$ y como $xz \in E(G)$ debe valer $yz \in E(G)$.

Probemos ahora que D es acíclico. Supongamos que no, y consideremos un ciclo orientado de D . En particular, existe uno sin cuerdas. Además, como el subgrafo inducido por cada clase es completo y el subdigrafo inducido por cada clase acíclico, el ciclo tiene a lo sumo dos vértices de cada clase, y necesariamente un arco de V^1 a V^2 y otro de V^2 a V^1 .

Caso 1: El ciclo consiste en dos vértices, $v_1 \in V^1$ y $v_2 \in V^2$.

No puede ocurrir, ya que no pueden ocurrir en simultáneo que $P_X(v_1)$ interseca v_2 y $P_Y(v_2)$ interseca v_1 sin que v_1 y v_2 se intersequen.

Caso 2: El ciclo es $v_1 w_1 v_2$ tal que $v_1, w_1 \in V^1$ y $v_2 \in V^2$.

Si $v_1 w_1 \in D$ entonces $X_2(v_1) < X_2(w_1)$ y por lo tanto también $Y_2(v_1) < Y_2(w_1)$. $P_X(v_1)$ interseca v_2 , por lo tanto $Y_1(v_2) < Y_2(v_1) < Y_2(w_1)$, y $P_Y(v_2)$ interseca w_1 , por lo tanto $X_1(w_1) < X_2(v_2)$, contradiciendo el hecho de que v_2 y w_1 no son adyacentes.

Caso 3: El ciclo es $v_2 w_2 v_1$ tal que $v_1 \in V^1$ y $v_2, w_2 \in V^2$.

Si $v_2 w_2 \in D$ entonces $X_2(v_2) < X_2(w_2)$. $P_Y(v_2)$ interseca v_1 , por lo tanto $X_1(v_1) < X_2(v_2) < X_2(w_2)$, y $P_X(v_1)$ interseca w_2 , por lo tanto $Y_1(w_2) < Y_2(v_1)$, contradiciendo el hecho de que v_1 y w_2 no son adyacentes.

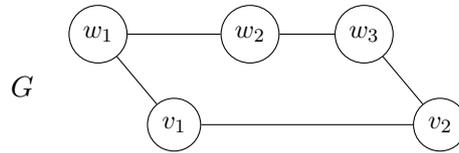
Caso 4: El ciclo es $v_1 w_1 v_2 w_2$ tal que $v_1, w_1 \in V^1$ y $v_2, w_2 \in V^2$.

Si $v_i w_i \in D$, para $i = 1, 2$, entonces $X_2(v_i) < X_2(w_i)$ y por lo tanto también $Y_2(v_i) < Y_2(w_i)$. $P_Y(v_2)$ interseca w_1 y v_2 no interseca w_1 , por lo tanto $Y_2(v_1) < Y_2(w_1) < Y_1(v_2)$. $P_X(v_1)$ interseca w_2 y v_1 no interseca w_2 , por lo tanto $X_2(v_2) < X_2(w_2) < X_1(v_1)$. Por Lema 3.7, eso contradice para v_1 y v_2 que el modelo es alineado.

□

3.3. Ejemplos

El ejemplo anterior se vuelve a utilizar con la nueva reducción.

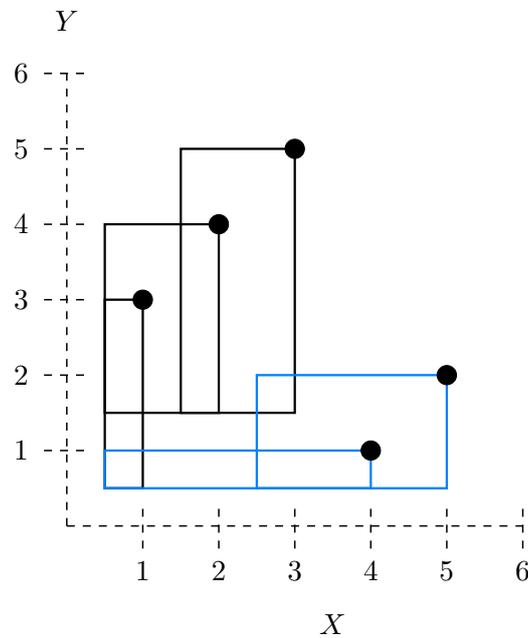


$$V^1 = \{v_1, v_2\} \quad V^2 = \{w_1, w_2, w_3\}$$

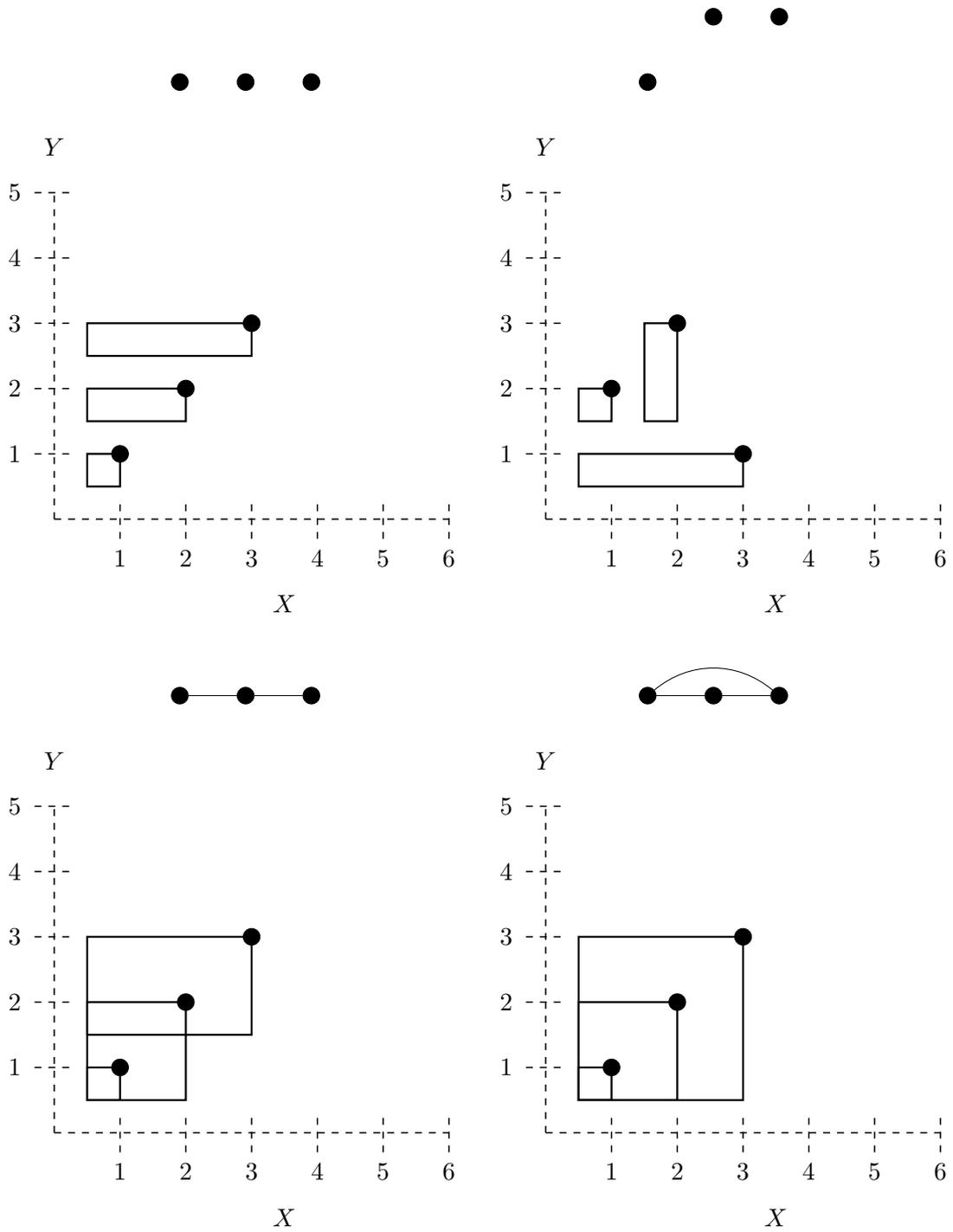
$$\begin{aligned} U(1, v_1) &= 1 & U(1, v_2) &= 1 \\ U(2, v_1) &= 0 & U(2, v_2) &= 2 \end{aligned}$$

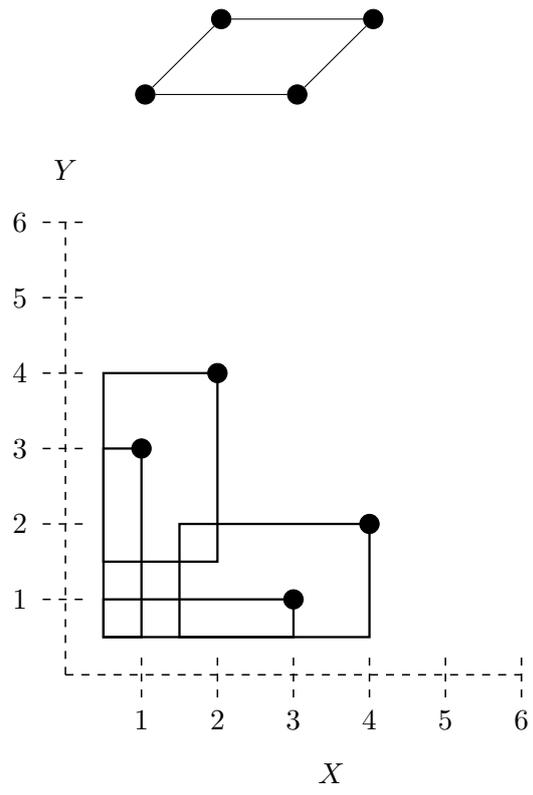
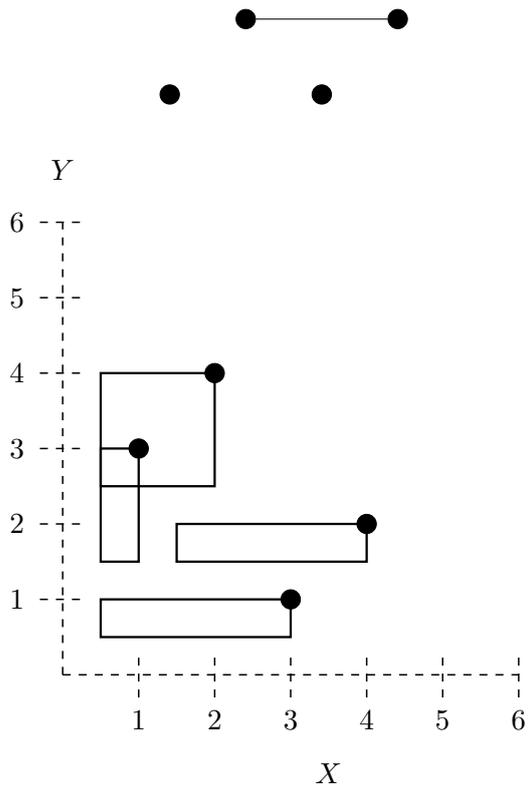
$$\begin{aligned} U(1, w_1) &= 0 & U(1, w_2) &= 1 & U(1, w_3) &= 1 \\ U(2, w_1) &= 1 & U(2, w_2) &= 1 & U(2, w_3) &= 2 \end{aligned}$$

v_1	$X_2 = 4$	$Y_2 = 1$	$X_1 = 0.5$	$Y_1 = 0.5$
v_2	$X_2 = 5$	$Y_2 = 2$	$X_1 = 2.5$	$Y_1 = 0.5$
w_1	$X_2 = 1$	$Y_2 = 3$	$X_1 = 0.5$	$Y_1 = 0.5$
w_2	$X_2 = 2$	$Y_2 = 4$	$X_1 = 0.5$	$Y_1 = 1.5$
w_3	$X_2 = 3$	$Y_2 = 5$	$X_1 = 1.5$	$Y_1 = 1.5$



Se muestran ejemplos adicionales de la reducción.





4. MODELO DE INTERSECCIÓN PARA 2-THIN PROPIO

Se extiende lo desarrollado en la sección anterior para definir un modelo de intersección para los grafos 2-thin propios.

Definición 4.1 Un conjunto de cajas con lados paralelos a los ejes Cartesianos cumple la propiedad *bi-semi-propia* si dadas 2 cajas A, B del conjunto se cumplen las siguientes condiciones:

$$X_2(A) < X_2(B) \Rightarrow X_1(A) \leq X_1(B)$$

$$Y_2(A) < Y_2(B) \Rightarrow Y_1(A) \leq Y_1(B)$$

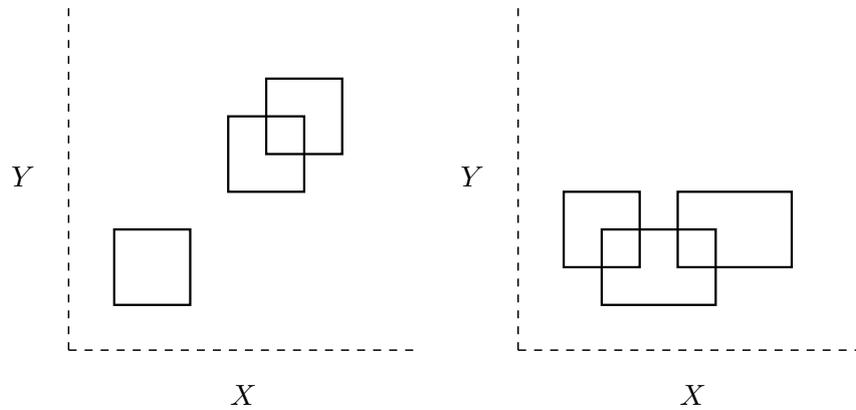


Fig. 4.1: Ejemplos de conjuntos de cajas que cumplen la propiedad bi-semi-propia.

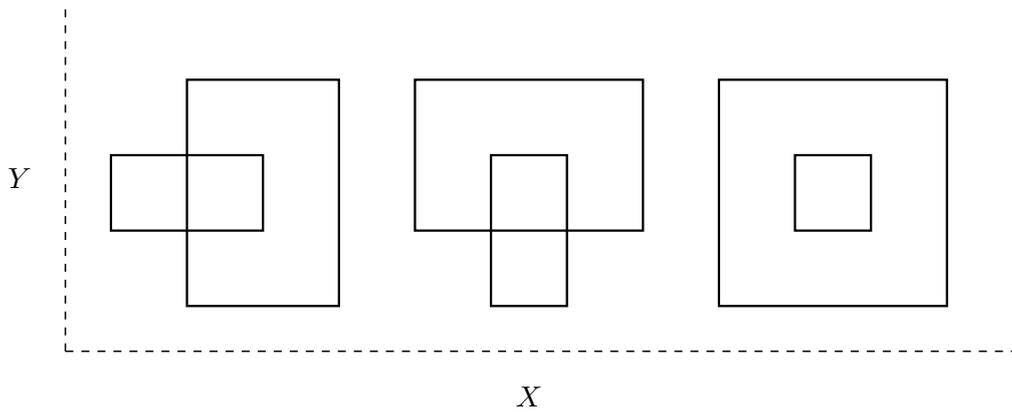


Fig. 4.2: Ejemplos de conjuntos de cajas que no cumplen la propiedad bi-semi-propia.

Teorema 4.2 G es un grafo 2-thin propio si y sólo si G tiene un modelo como intersección de cajas con lados paralelos a los ejes Cartesianos que cumple la propiedad diagonal, en el cual las cajas con extremo superior derecho en cada una de las diagonales cumplen la propiedad bi-semi-propia.

Demostración. \Rightarrow) Sea $G = (V, E)$ un grafo 2-thin propio, con partición V^1, V^2 y orden fuertemente consistente $<$. Sean $V^1 = v_1 < \dots < v_{n_1}$, $V^2 = w_1 < \dots < w_{n_2}$.

Se utilizan las mismas definiciones de $U(1, v_i)$, $U(2, v_i)$, $U(1, w_i)$, $U(2, w_i)$ que en la demostración del Teorema 3.8. Definimos el siguiente modelo como intersección de cajas en el plano, que también es idéntico al Teorema 3.8:

$$\begin{aligned} v_i : \quad (X_2, Y_2) &= (i + n_2, i) \quad 1 \leq i \leq n_1 \\ (X_1, Y_1) &= (U(2, v_i) + 0.5, U(1, v_i) - 0.5) \quad 1 \leq i \leq n_1 \end{aligned}$$

$$\begin{aligned} w_i : \quad (X_2, Y_2) &= (i, i + n_1) \quad 1 \leq i \leq n_2 \\ (X_1, Y_1) &= (U(2, w_i) - 0.5, U(1, w_i) + 0.5) \quad 1 \leq i \leq n_2 \end{aligned}$$

Como G es 2-thin propio, también sabemos que es 2-thin y por lo tanto el modelo representa al grafo y es diagonal. Solo necesitamos demostrar que se cumple la propiedad bi-semi-propia para cada diagonal.

Sean $v_a, v_b \in V^1$ tales que $a < b$ y $v_a < v_b$. Sabemos que $X_2(v_b) = b + n_2 > X_2(v_a) = a + n_2$, $X_1(v_a) = U(2, v_a) + 0.5$ y $X_1(v_b) = U(2, v_b) + 0.5$.

Se demuestra que $U(2, v_b) \geq U(2, v_a)$. Si $U(2, v_a) = 0$ se cumple. Si $U(2, v_a) = j > 0$ entonces existe $w_j \in V^2$ tal que $w_j < v_a$ y $w_j v_a \notin E(G)$.

Supongamos que existe k tal que $U(2, v_b) = w_k \in V^2$ y $w_k < w_j$. Por definición de $U(2, v_i)$ entonces se cumple que $w_k v_b \notin E(G)$ y $w_j v_b \in E(G)$. Por definición de 2-thin propio entonces $w_j v_a \in E(G)$, contradicción. Si $U(2, v_b) = 0$ vale que $w_j v_b \in E(G)$ y se obtiene la misma contradicción.

Para el eje Y sabemos que $Y_2(v_b) = b > Y_2(v_a) = a$, $Y_1(v_a) = U(1, v_a) - 0.5$ y $Y_1(v_b) = U(1, v_b) - 0.5$.

Se demuestra que $U(1, v_b) \geq U(1, v_a)$. Se ve primero el caso en que $U(1, v_a) = a$. Supongo que existe $U(1, v_b) = j_b < a$. Por definición de $U(1, v_i)$ vale que $v_{j_b} v_b \in E(G)$. Como $v_{j_b} < v_a < v_b$ y la definición de 2-thin propio, entonces es verdadero que $v_{j_b} v_a \in E(G)$. Esto implica que $U(1, v_a) < a$, contradicción.

Se ve el caso en donde $U(1, v_a) = j_a < a$. Supongo que existe $U(1, v_b) = j_b < j_a$. Por definición de $U(1, v_i)$ vale que $v_{j_b} v_b \in E(G)$. Como $v_{j_b} < v_{j_a} < v_a < v_b$ y la definición de 2-thin propio, entonces es verdadero que $v_{j_b} v_a \in E(G)$. Esto implica que $U(1, v_a) < j_a$, contradicción.

El razonamiento es similar para $v_a, v_b \in V^2$.

\Leftarrow) Consideremos un modelo de G como intersección de cajas con lados paralelos a los ejes Cartesianos que cumple la propiedad diagonal, en el cual las cajas con extremo superior derecho en cada una de las diagonales cumplen la propiedad bi-semi-propia. Sea V^i el conjunto de vértices v tales que $Y_2(v) - X_2(v) = d_i$, para $i = 1, 2$. Podemos

asumir sin pérdida de generalidad que $d_1 < d_2$ y que en cada V^i todas las coordenadas X_2 son distintas, ya que la propiedad a demostrar es cerrada por el agregado de vértices mellizos verdaderos. Vamos a definir un digrafo D con los mismos vértices que G y tal que cualquier orden topológico de D (si existe alguno) es fuertemente consistente con la partición V^1, V^2 y extiende el orden de V^1 y V^2 definido por las componentes X_2 en cada uno de los conjuntos (aunque no necesariamente es el orden de las componentes X_2 en la unión de los conjuntos). Vamos a probar que ese digrafo es acíclico y por lo tanto admite un orden topológico. Para asegurar que extiende los órdenes de V^1 y V^2 , dados dos vértices v, v' en el mismo V^i , creamos el arco vv' en D si y sólo si $X_2(v) < X_2(v')$. Dados dos vértices $v_i \in V^i, v_{3-i} \in V^{3-i}$, creamos el arco $v_i v_{3-i}$ en D si y sólo si $v_i v_{3-i} \notin E(G)$ y se cumple una de estas condiciones: existe $v'_{3-i} \in V^{3-i}$ con $X_2(v'_{3-i}) < X_2(v_{3-i})$ y $v_i v'_{3-i} \in E(G)$, o existe $v'_i \in V^i$ con $X_2(v'_i) > X_2(v_i)$ y $v'_i v_{3-i} \in E(G)$.

Probemos primero que un orden topológico $<$ de D es fuertemente consistente con V^1, V^2 . Sean $x < y < z$ con $xz \in E(G)$. Supongamos primero que $x, y, z \in V^1$ (las definiciones son simétricas son respecto a ambas clases). Entonces $X_2(x) < X_2(y) < X_2(z)$ y como $xz \in E(G)$, vale $X_1(z) < X_2(x) < X_2(y)$ e $Y_1(z) < Y_2(x) = X_2(x) + d_1 < X_2(y) + d_1 = Y_2(y)$. Por lo tanto $yz \in E(G)$. Por la propiedad bi-semi-propia, $X_1(y) \leq X_1(z) < X_2(x)$ e $Y_1(y) \leq Y_1(z) < Y_2(x)$, y por lo tanto $xy \in E(G)$. Ahora supongamos que $x, y \in V^1, z \in V^2$ (las definiciones son simétricas son respecto a ambas clases). Como $y < z$ en un orden topológico de D , $zy \notin A(D)$ y como $xz \in E(G)$ debe valer $yz \in E(G)$. Por último, supongamos que $x \in V^1, y, z \in V^2$ (las definiciones son simétricas son respecto a ambas clases). Como $x < y$ en un orden topológico de D , $yx \notin A(D)$ y como $xz \in E(G)$ debe valer $yx \in E(G)$.

Probemos ahora que D es acíclico. Supongamos que no, y consideremos un ciclo orientado de D . En particular, existe uno sin cuerdas. Además, como el subgrafo inducido por cada clase es completo y el subdigrafo inducido por cada clase acíclico, el ciclo tiene a lo sumo dos vértices de cada clase, y necesariamente un arco de V^1 a V^2 y otro de V^2 a V^1 . Los arcos entre $v_1 \in V^1$ y $v_2 \in V^2$ verifican que $X_2(v_1) > X_2(v_2)$ e $Y_2(v_1) < Y_2(v_2)$. Usaremos el subíndice ij si la orientación del arco es de V^i a V^j .

Las siguientes propiedades son implicadas por el hecho de ser una representación bi-semi-propia.

$$\begin{aligned} \text{Tipo } a_{12}: & \quad Y_1(v_2) > Y_2(v_1) \text{ y } X_1(v_1) < X_2(v_2) \\ \text{Tipo } a_{21}: & \quad Y_1(v_2) < Y_2(v_1) \text{ y } X_1(v_1) > X_2(v_2) \end{aligned}$$

Veamos los distintos casos posibles de ciclos.

Caso 1: El ciclo consiste en dos vértices, $v_1 \in V^1$ y $v_2 \in V^2$.

No puede ocurrir, dado que a_{12} y a_{21} no pueden ocurrir en simultáneo.

Caso 2: El ciclo es $v_1 w_1 v_2$ tal que $v_1, w_1 \in V^1$ y $v_2 \in V^2$.

Si $v_1 w_1 \in D$ entonces $X_2(v_1) < X_2(w_1)$ y por lo tanto también $Y_2(v_1) < Y_2(w_1)$. Al ser bi-semi-propia la representación, $X_1(v_1) \leq X_1(w_1)$ e $Y_1(v_1) \leq Y_2(w_1)$.

El arco $w_1 v_2$, por lo visto antes, debe ser tipo a_{12} , con lo cual $Y_1(v_2) > Y_2(w_1)$ y $X_1(w_1) < X_2(v_2)$.

A su vez, el arco v_2v_1 , por lo visto antes, debe ser tipo a_{21} , con lo cual $Y_1(v_2) < Y_2(v_1)$ y $X_1(v_1) > X_2(v_2)$.

Los arcos son incompatibles ya que implicarían $Y_2(w_1) < Y_1(v_2) < Y_2(v_1)$ pero sabemos que $Y_2(v_1) < Y_2(w_1)$.

Caso 3: El ciclo es $v_2w_2v_1$ tal que $v_1 \in V^1$ y $v_2, w_2 \in V^2$.

Si $v_2w_2 \in D$ entonces $X_2(v_2) < X_2(w_2)$ y por lo tanto también $Y_2(v_2) < Y_2(w_2)$. Al ser bi-semi-propia la representación, $X_1(v_2) \leq X_1(w_2)$ e $Y_1(v_2) \leq Y_2(w_2)$.

El arco v_1v_2 , por lo visto antes, debe ser tipo a_{12} , con lo cual $Y_1(v_2) > Y_2(v_1)$ y $X_1(v_1) < X_2(v_2)$.

A su vez, el arco w_2v_1 , por lo visto antes, debe ser tipo a_{21} , con lo cual $Y_1(w_2) < Y_2(v_1)$ y $X_1(v_1) > X_2(w_2)$.

Los arcos son incompatibles ya que implicarían $Y_1(v_2) > Y_2(v_1) > Y_1(w_2)$, contradicción.

Caso 4: El ciclo es $v_1w_1v_2w_2$ tal que $v_1, w_1 \in V^1$ y $v_2, w_2 \in V^2$.

Si $v_iw_i \in D$, para $i = 1, 2$, entonces $X_2(v_i) < X_2(w_i)$ y por lo tanto también $Y_2(v_i) < Y_2(w_i)$. Al ser bi-semi-propia la representación, $X_1(v_i) \leq X_1(w_i)$ e $Y_1(v_i) \leq Y_2(w_i)$.

El arco w_1v_2 , por lo visto antes, debe ser tipo a_{12} , con lo cual $Y_1(v_2) > Y_2(w_1)$ y $X_1(w_1) < X_2(v_2)$.

A su vez, el arco w_2v_1 , por lo visto antes, debe ser tipo a_{21} , con lo cual $Y_1(w_2) < Y_2(v_1)$ y $X_1(v_1) > X_2(w_2)$.

Los arcos son incompatibles ya que implicarían $Y_2(v_1) > Y_1(w_2) \geq Y_1(v_2) > Y_2(w_1)$, contradicción.

□

5. MATRICES

Los grafos se pueden representar con matrices y una consecuencia de esto es que un grafo puede caracterizarse en base a propiedades sobre una matriz asociada. Para el caso de los grafos de intervalos una caracterización clásica es con los grafos cuya matriz de cliques maximales y vértices cumplen la propiedad C1P [9].

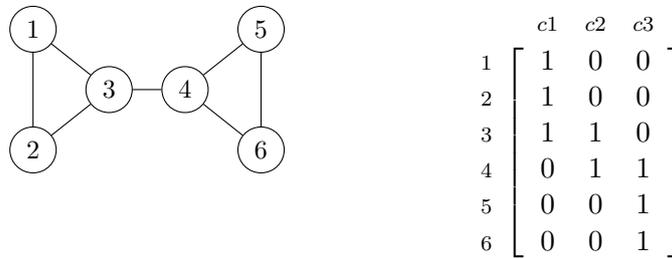
Definición 5.1 Una matriz cuyos elementos son 0 y 1 tiene la propiedad C1P (Consecutive Ones Property) si las filas tienen los 1 en forma consecutiva.

$$(j < k \wedge M_{i,j} = 1 \wedge M_{i,k} = 1) \implies (M_{i,x} = 1 \text{ para } j \leq x \leq k)$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Tab. 5.1: Ejemplos de matrices con la propiedad C1P.

Definición 5.2 $CV(G)$ es la matriz de vértices vs cliques maximales del grafo G . Existen $|V(G)|$ filas y tantas columnas como cliques maximales tenga G .



Tab. 5.2: Ejemplo de matriz de vértices vs cliques maximales.

Teorema 5.3 [10] Un grafo G es de intervalos si y sólo si pueden permutarse las columnas de $CV(G)$ de manera que la matriz resultante posea la propiedad C1P.

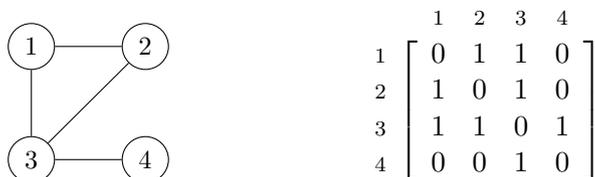
El teorema anterior es equivalente a decir que existe un ordenamiento de los cliques maximales C_1, C_2, \dots, C_k en donde cada vértice aparece en forma consecutiva en la secuencia. Expresado formalmente:

$$(i < j \wedge v \in C_i \cap C_j) \implies (v \in C_k \text{ para } i \leq k \leq j)$$

Más adelante se desarrolla una caracterización equivalente que tiene la particularidad de que no necesita tener en cuenta los cliques del grafo.

5.1. Permutaciones de matrices de adyacencia

Un grafo de n vértices puede representarse con una matriz binaria A de $n \times n$ en donde $A_{i,j} = 1$ si $v_i v_j \in E(G)$ y 0 en otro caso. Para grafos no dirigidos esta matriz es simétrica.



$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \\
 1 \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\
 2 \\
 3 \\
 4
 \end{array}$$

Ordenando los vértices de otra manera se obtiene una matriz con la misma información.

$$\begin{array}{c}
 4 \quad 1 \quad 3 \quad 2 \\
 4 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \\
 1 \\
 3 \\
 2
 \end{array}$$

Comenzando con una matriz de adyacencia cualquiera se puede obtener la permutación deseada multiplicando de ambos lados por una matriz de permutación. Una matriz de permutación es una matriz binaria en donde toda fila y toda columna tiene un solo elemento 1.

Dada P una matriz de permutación, se permutan filas de un matriz M realizando la multiplicación PM y se permutan columnas con MP

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}
 \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}
 =
 \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \\ b_1 & b_2 & b_3 & b_4 \\ d_1 & d_2 & d_3 & d_4 \\ a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}
 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}
 =
 \begin{bmatrix} a_4 & a_2 & a_1 & a_3 \\ b_4 & b_2 & b_1 & b_3 \\ c_4 & c_2 & c_1 & c_3 \\ d_4 & d_2 & d_1 & d_3 \end{bmatrix}$$

Observar que en el ejemplo al permutar las filas la matriz realiza la siguiente permutación:

Fila nueva 1 := Fila anterior 3
 Fila nueva 2 := Fila anterior 2
 Fila nueva 3 := Fila anterior 4
 Fila nueva 4 := Fila anterior 1

Mientras que al permutar columnas con la misma matriz se obtiene:

Columna nueva 1 := Columna anterior 4
 Columna nueva 2 := Columna anterior 2
 Columna nueva 3 := Columna anterior 1
 Columna nueva 4 := Columna anterior 3

Para lograr la misma permutación en las filas se puede transponer la matriz de permutación, de esa manera se logra la misma permutación que en las filas.

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a_3 & a_2 & a_4 & a_1 \\ b_3 & b_2 & b_4 & b_1 \\ c_3 & c_2 & c_4 & c_1 \\ d_3 & d_2 & d_4 & d_1 \end{bmatrix}$$

Columna nueva 1 := Columna anterior 3

Columna nueva 2 := Columna anterior 2

Columna nueva 3 := Columna anterior 4

Columna nueva 4 := Columna anterior 1

Lo desarrollado anteriormente permite obtener una permutación de la matriz de adyacencia permutando las filas y columnas. De la matriz A obtenemos una permutación realizando la operación $P \cdot A \cdot P^T$. De ahora en adelante cuando decimos que un grafo tiene una matriz de adyacencia, estamos asumiendo un orden específico.

5.2. Propiedad C1D

Definición 5.4 Una matriz binaria de $n \times n$ tiene la propiedad C1D si

$$\left(3 \leq j \leq n \wedge 1 \leq i \leq j - 2 \wedge M_{i,j} = 1 \right) \implies \left(M_{k,j} \neq 0 \text{ para } i \leq k \leq j - 1 \right)$$

Expresado en palabras, cada vez que hay un 1 en un elemento de una columna, esto implica que todas las posiciones en la misma columna debajo del elemento deben ser distintas a 0 hasta llegar a la diagonal.

Ejemplos de matrices con la propiedad C1D. Se utiliza el elemento * para las posiciones que no tienen efecto en la validez de la propiedad.

$$\begin{bmatrix} * & 1 & 0 & 0 \\ * & * & 1 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 0 & 0 & 1 \\ * & * & 0 & 1 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 1 & 0 \\ * & * & 1 & 1 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix}$$

Matrices que no cumplen la propiedad C1D

$$\begin{bmatrix} * & 1 & 1 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 0 & 0 & 1 \\ * & * & 0 & 1 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 1 & 1 \\ * & * & 1 & 0 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix}$$

Teorema 5.5 Un grafo G es de intervalos si y sólo si existe una matriz de adyacencia de G que tenga la propiedad C1D.

Demostración. \implies) Sea G un grafo de intervalos. Sea v_1, v_2, \dots, v_n un ordenamiento de $V(G)$ consistente. Se construye una matriz de adyacencia M utilizando el mismo orden. Sea $v_r, v_t \in V(G)$ tal que $r < t$ y $v_r v_t \in E(G)$, si no existe la propiedad vale trivialmente. La existencia de la arista implica que $M_{r,t} = 1$ y para los valores s tal que $r < s < t$ se obtiene que $v_s v_t \in E(G)$ y $M_{s,t} = 1$ debido a la definición de grafo de intervalos.

\impliedby) Sea M una matriz de adyacencia de G que cumple la propiedad C1D. Se utiliza el ordenamiento de columnas v_1, v_2, \dots, v_n de M . Sea $r < t$ tal que $M_{r,t} = 1$, en caso de no existir la propiedad vale trivialmente. Esto implica que $v_r v_t \in E(G)$ y como M cumple la propiedad C1D, entonces $M_{s,t} = 1$ para $r < s < t$ lo implica $v_s v_t \in E(G)$. \square

5.3. Grafos k-thin y C1D

La definición anterior puede extenderse para capturar a los grafos k-thin.

Definición 5.6 Dada la matriz A de adyacencia de G y $W \subseteq V(G)$ se define la matriz $FR(A, W)$ como

$$FR(A, W)_{i,j} = \begin{cases} A_{i,j} & \text{si } v_i \in W \\ 2 & \text{si } v_i \notin W \end{cases}$$

$$W = \{1, 3\} \quad A = \begin{array}{c} \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix} \end{array} \quad FR(A, W) = \begin{array}{c} \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 0 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix} \end{matrix} \end{array}$$

Tab. 5.3: Ejemplo de $FR(A, W)$.

Teorema 5.7 Un grafo G es k-thin si y sólo si existe una matriz A de adyacencia de G y una partición V^1, \dots, V^k de V tal que las matrices $FR(A, V^1), \dots, FR(A, V^k)$ tengan la propiedad C1D.

Demostración. \implies) Sea G un grafo k-thin. Sea v_1, v_2, \dots, v_n un ordenamiento de $V(G)$ consistente con la partición V^1, \dots, V^k . Se construye una matriz de adyacencia M utilizando el mismo orden. Sea $v_r, v_t \in V(G)$ tales que $r < t$ y $v_r v_t \in E(G)$. Si no existe ninguna arista entonces la propiedad vale trivialmente.

Sabemos que $M_{r,t} = 1$ y por la definición de k-thin vale que $v_s v_t \in E(G)$ para los $v_s \in V(G)$ que cumplen $r < s < t$ y se encuentran en la misma clase que v_r .

Sea V^Z tal que $v_r \in V^Z$. Debido a lo anterior también vale que $v_s \in V^Z$ y por lo tanto $FR(M, V^Z)$ tiene la propiedad C1D.

\impliedby) Sea M una matriz de adyacencia de G y una partición V^1, \dots, V^k de $V(G)$ tal que las matrices $FR(M, V^1), \dots, FR(M, V^k)$ tengan la propiedad C1D. Se utiliza el ordenamiento de columnas v_1, v_2, \dots, v_n de M para ordenar $V(G)$. Sea $M_{r,t} = 1$ con $r < t$, si no existe un elemento igual a 1 entonces la propiedad vale trivialmente. Sea V^Z tal que $v_r \in V^Z$. La matriz $FR(M, V^Z)$ cumple la propiedad C1D lo que implica que $M_{s,t} = 1$ para $r < s < t$ y $v_s \in V^Z$. Debido a esto $v_s v_t \in E(G)$ cuando $v_r \in V^Z, v_s \in V^Z$ y $r < s < t$. \square

A continuación se define una propiedad que nos servirá para caracterizar a los grafos de intervalos propios.

5.4. Propiedad C1L

Definición 5.8 Una matriz binaria de $n \times n$ tiene la propiedad C1L si

$$\left(1 \leq i \leq n-2 \wedge i+2 \leq j \leq n \wedge M_{i,j} = 1\right) \implies \left(M_{i,k} \neq 0 \text{ para } i+1 \leq k \leq j-1\right)$$

Cada vez que hay un 1 en un elemento, esto implica que todas las posiciones en la misma fila a la izquierda del elemento deben ser distintas a 0 hasta llegar a la diagonal.

Matrices que cumplen la propiedad C1L

$$\begin{bmatrix} * & 1 & 1 & 1 \\ * & * & 1 & 0 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 1 & 0 \\ * & * & 1 & 1 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 0 & 0 \\ * & * & 1 & 0 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix}$$

Matrices que no cumplen la propiedad C1L

$$\begin{bmatrix} * & 0 & 1 & 1 \\ * & * & 1 & 1 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 1 & 0 \\ * & * & 0 & 1 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 0 & 1 \\ * & * & 1 & 1 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix}$$

Teorema 5.9 Un grafo G es de intervalo propio si y sólo si existe una matriz de adyacencia de G que tenga la propiedad C1D y C1L.

Demostración. \implies) Sea G un grafo de intervalo propio. Sea v_1, v_2, \dots, v_n un ordenamiento de $V(G)$ consistente. Se construye una matriz de adyacencia M utilizando el mismo orden. Sea $v_r v_t \in V(G)$ tal que $r < t$ y $v_r v_t \in E(G)$. Si no existe la arista la propiedad vale trivialmente. Asumiendo que existe la arista entonces $M_{r,t} = 1$

Para los valores de s tal que $r < s < t$, sabemos que $v_s v_t \in E(G)$ y $v_r v_s \in E(G)$ por la definición de grafo de intervalo propio. De esta manera $M_{s,t} = 1$ y $M_{r,s} = 1$, por lo que las propiedades C1D y C1L se verifican.

\Leftarrow) Sea M una matriz de adyacencia de G que cumple las propiedades C1D y C1L. Se utiliza en el grafo para ordenar los vértices el ordenamiento de columnas v_1, v_2, \dots, v_n de M . Sea $M_{r,t} = 1$ un elemento de la matriz, si no existe entonces no hay aristas y por lo tanto la propiedad vale trivialmente. Suponiendo que exista el elemento, entonces $v_r v_t \in E(G)$ con $r < t$.

Como M cumple la propiedad C1D, entonces $M_{s,t} = 1$ para $r < s < t$ y entonces $v_s v_t \in E(G)$ para $r < s < t$.

Dado que M cumple la propiedad C1L, entonces $M_{r,x} = 1$ para $r < x < t$ y entonces $v_r v_s \in E(G)$ para $r < s < t$. \square

5.5. Grafos de k-thin propios

Se extiende la definición anterior para el caso de los grafos k-thin propios.

Definición 5.10 Dada la matriz A de adyacencia de G y $W \subseteq V(G)$ se define la matriz $FC(A, W)$ como

$$FC(A, W)_{i,j} = \begin{cases} A_{i,j} & \text{si } v_j \in W \\ 2 & \text{si } v_j \notin W \end{cases}$$

$$W = \{2, 3, 5\} \quad A = \begin{array}{c} \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix} \quad FC(A, W) = \begin{array}{c} \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 2 & 0 & 1 & 2 & 1 \\ 2 & 0 & 1 & 2 & 0 \\ 2 & 1 & 0 & 2 & 1 \\ 2 & 1 & 0 & 2 & 0 \\ 2 & 1 & 0 & 2 & 0 \end{bmatrix} \end{matrix} \end{array}$$

Tab. 5.4: Ejemplo de $FC(A, W)$.

Teorema 5.11 Un grafo G es k-thin propio si y sólo si existe una matriz A de adyacencia de G y una partición V^1, \dots, V^k de V tal que las matrices $FR(A, V^1), \dots, FR(A, V^k)$ tengan la propiedad C1D y las matrices $FC(A, V^1), \dots, FC(A, V^k)$ tengan la propiedad C1L.

Demostración. \implies) Sea G un grafo k-thin propio. Sea v_1, v_2, \dots, v_n un ordenamiento de $V(G)$ consistente con la partición V^1, \dots, V^k . Se construye una matriz de adyacencia M utilizando el mismo orden. Sea $v_r v_t \in E(G)$ tal que $r < t$. Si no existe la arista entonces la propiedad vale trivialmente. Debido a la arista se cumple que $M_{r,t} = 1$. Sea V^Z tal que $v_r \in V^Z$. Por la definición de grafo k-thin se obtiene que $v_s v_t \in E(G)$ si se cumple que $r < s < t$ y $v_s \in V^Z$. Por lo tanto $M_{s,t} = 1$ y $FR(M, V^Z)$ tiene la propiedad C1D.

Sea V^Q tal que $v_t \in V^Q$. Por la definición de grafo k-thin se obtiene que $v_r v_s \in E(G)$ si se cumple que $r < s < t$ y $v_s \in V^Q$. Por lo tanto $M_{r,s} = 1$ y $FC(M, V^Q)$ tiene la propiedad C1L.

\impliedby) Sea M una matriz de adyacencia de G y una partición V^1, \dots, V^k de $V(G)$ tal que las matrices $FC(M, V^1), \dots, FC(M, V^k)$ tengan la propiedad . Se utiliza el ordenamiento de columnas v_1, v_2, \dots, v_n de M . Sea $M_{r,t} = 1$ con $r < t$. Si no existe el elemento igual a 1 entonces la propiedad vale trivialmente.

Sea V^Z tal que $FR(M, V^Z)_{r,t} = 1$ con $r < t$, esta matriz existe y es única debido a que V^1, \dots, V^k es una partición de $V(G)$.

Para los valores s tales que $r < s < t$ y $v_s \in V^Z$ se obtiene que $FR(M, V^Z)_{s,t} = 1$ porque $FR(M, V^Z)$ cumple la propiedad C1D. De esta manera $v_s v_t \in E(G)$ cuando $r < s < t$ y $v_r \in V^Z, v_s \in V^Z$.

Sea V^Q tal que $FC(M, V^Q)_{r,t} = 1$ con $r < t$, esta matriz existe y es única debido a que V^1, \dots, V^k es una partición de $V(G)$.

Para los valores s tales que $r < s < t$ y $v_s \in V^Q$ se obtiene que $FC(M, V^Q)_{r,s} = 1$ porque $FC(M, V^Q)$ cumple la propiedad C1L. De esta manera $v_r v_s \in E(G)$ cuando $r < s < t$ y $v_s \in V^Q, v_t \in V^Q$. \square

5.6. NP-Compleitud

En esta sección se ejemplifican 2 problemas NP-Completos que pueden especificarse en base a una matriz binaria.

Definición 5.12 Una matriz de $n \times n$ tiene la propiedad C si $A_{i,i+1} = 1$ para $1 \leq i < n$

$$\begin{bmatrix} * & 1 & 0 & 0 & 0 \\ * & * & 1 & 0 & 0 \\ * & * & * & 1 & 0 \\ * & * & * & * & 1 \\ * & * & * & * & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & 1 & 0 & 1 \\ * & * & 1 & 0 & 0 \\ * & * & * & 1 & 1 \\ * & * & * & * & 1 \\ * & * & * & * & * \end{bmatrix}$$

Tab. 5.5: Matrices que cumplen la propiedad C.

Problema 1

Instancia: Matriz simétrica binaria de $n \times n$

Pregunta: ¿ Pueden permutarse las filas y columnas simultáneamente de manera que la matriz resultante tenga la propiedad C ?

Problema 2

Instancia: Matriz binaria de $n \times n$, entero K

Pregunta: ¿ Existe un conjunto de K columnas de manera que la submatriz de $n \times K$ tenga la propiedad C1P ?

El problema 1 es el problema de determinar la existencia de un camino hamiltoniano y puede reducirse al problema 2 en tiempo polinomial. Por lo que ambos son problemas NP-Completos [11].

Los ejemplos anteriores, al igual que los muchos problemas existentes NP-Completo definidos sobre matrices [9] podrían ser útiles para demostrar que reconocer grafos k-thin y k-thin propios son problemas NP-Completo. No se ha encontrado hasta el momento una reducción.

6. MÉTRICAS

Se busca la relación de la thinness con otras propiedades existentes de grafos.

6.1. Bandwidth

Definición 6.1 Dado un grafo $G = (V, E)$ con $E \neq \emptyset$ y un ordenamiento v_1, \dots, v_n de V , se define la función

$$Bandwidth'(G, v_1, \dots, v_n) = \max\{|i - j| : v_i v_j \in E\}$$

De esta manera se puede definir el ancho de banda de un grafo.

$$Bandwidth(G) = \min\{Bandwidth'(G, v_1, \dots, v_n) : v_1, \dots, v_n \text{ es un ordenamiento de } V\}$$

Teorema 6.2 Si $E(G) \neq \emptyset$ entonces $thin(G) \leq Bandwidth(G)$

Demostración. Sea G un grafo con $E(G) \neq \emptyset$ y $Bandwidth(G) = k$. Debido a esto existe un ordenamiento v_1, \dots, v_n de $V(G)$ tal que $|i - j| \leq k$ si $v_i v_j \in E(G)$

Sea $G_{<}$ el grafo generado según el Teorema 1.7 para el ordenamiento anterior. Supongo que $G_{<}$ tiene un clique de $k + 1$ vértices. Sean a y b respectivamente el menor y mayor índice de los vértices del clique en el ordenamiento. Como el clique tiene $k + 1$ vértices entonces $b - a \geq k$

Por definición de $G_{<}$ existe un índice c tal que $c > b$ y $v_a v_c \in E(G)$ y $v_b v_c \notin E(G)$. Pero entonces $c - a > k$ lo que contradice que $Bandwidth(G) = k$ □

Observemos que la acotación puede no ser ajustada, dado que $thin(K_n) = 1$ y $Bandwidth(K_n) = n - 1$.

Definición 6.3 $diam(G) = \max\{d(u, w) : u \in V(G) \wedge w \in V(G) \wedge u \neq w\}$

Teorema 6.4 [13] Si G es un grafo conexo entonces

$$\left\lceil \frac{|V(G)|}{diam(G)} \right\rceil \leq Bandwidth(G) \leq |V(G)| - diam(G)$$

Corolario 6.5 Si G es un grafo conexo entonces

$$thin(G) \leq |V(G)| - diam(G)$$

Un aspecto positivo es que mientras calcular bandwidth es NP-Hard [11], el diámetro puede calcularse con complejidad temporal $O(|V(G)|^3)$ utilizando el algoritmo de Floyd-Warshall [14].

6.2. Boxicity

Teorema 6.8 [16] Dado un grafo G , si \overline{G} tiene a C_k o P_k como subgrafos inducidos con $k \geq 4$ entonces $\text{boxicity}(G) \geq \lceil \frac{k}{3} \rceil$

Debido a que la thinness es mayor o igual que la boxicity, se obtiene en forma inmediata lo siguiente.

Corolario 6.9 Si \overline{G} tiene a C_k o P_k como subgrafos inducidos con $k \geq 4$ entonces $\text{thin}(G) \geq \lceil \frac{k}{3} \rceil$

Llevandolo a valores minimales de k se obtiene lo siguiente:

Corolario 6.10 Si \overline{G} tiene a C_4 o P_4 como subgrafos inducidos entonces $\text{thin}(G) \geq 2$

Corolario 6.11 Si \overline{G} tiene a C_7 o P_7 como subgrafos inducidos entonces $\text{thin}(G) \geq 3$

Corolario 6.12 Si \overline{G} tiene a C_{10} o P_{10} como subgrafos inducidos entonces $\text{thin}(G) \geq 4$

Corolario 6.13 Si \overline{G} tiene a C_{3k-2} o P_{3k-2} como subgrafos inducidos con $k \geq 2$ entonces $\text{thin}(G) \geq k$

6.3. Grafos planares

A diferencia de la boxicity, en donde $\text{boxicity}(G) \leq 3$ si G es planar [17], la thinness de grafos planares no se encuentra acotada. Esto puede verse con el grafo de grilla.

Definición 6.14 Un grafo $G = (V, E)$ es un grilla de $n \times m$ si $V = \{(x, y) : 1 \leq x \leq n \wedge 1 \leq y \leq m\}$ y $E = \{(x_1, y_1)(x_2, y_2) : |x_2 - x_1| + |y_2 - y_1| = 1\}$

La thinness del grafo de grilla de $n \times n$ es $\Theta(n)$ [4]. Sin embargo el grafo de grilla siempre es planar, por lo que la thinness de un grafo planar no se encuentra acotada.

6.4. Resultados conocidos

Se muestran todas las relaciones conocidas, los términos no definidos se encuentran definidos en las referencias correspondientes.

	Resultado	Notas	Referencia
	$1 \leq \text{thin}(G) \leq V(G) $		[1]
	$\text{thin}(H) \leq \text{thin}(G)$	H subgrafo inducido de G	[4]
	$\text{boxicity}(G) \leq \text{thin}(G)$		[4]
	$\text{lmimw}(G) \leq \text{thin}(G)$		[3]
	$b_v(G)/\Delta(G) \leq \text{thin}(G)$		[4]
	$\text{thin}(G) \leq \text{pathwidth}(G) + 1$		[1]
	$\text{thin}(G) \leq \text{cutwidth}(G) + 1$		[3]
	$\text{thin}(T) \leq \text{altura}(T)$	T árbol	[7]
	$\text{thin}(G) \leq \text{pthin}(G)$		
	$\text{thin}(G) \leq \text{bandwidth}(G)$	$ E(G) \geq 1$	
	$\text{thin}(G) \leq V(G) - \text{diam}(G)$	G conexo	

7. RECONOCIMIENTO

No se ha avanzado lo suficiente con la teoría como para encontrar un algoritmo eficiente. Debido a esto se recurrirá a herramientas existentes para calcular la thinness de un grafo. Una primera aproximación que puede realizarse es utilizar un programa para resolver SAT, es decir para decidir si una fórmula de lógica proposicional es satisfacible o no. Una limitación de este enfoque es en el manejo de números, en particular de obtener el número de la thinness resolviendo una única instancia.

Se utilizará el enfoque de SMT (Sat Modulo Theory) que es una extensión de SAT para determinar si es satisfacible o no una fórmula de lógica de primer orden.

7.1. Ejemplo

Como introducción supongamos que se quiere encontrar los valores enteros de x, y, z, k, r que cumplen las siguientes condiciones:

$$\begin{aligned}x^2 + 2(y + 3) + z^3 &= 4163 \\ x = 2k + 1 &\Rightarrow y = 2r\end{aligned}$$

Es posible resolver el sistema anterior con un resolvidor SAT ? Sí, pero puede llevar bastante trabajo, por ejemplo una forma sería codificar en registros binarios de longitud fija a los números x, y, z, k, r y establecer restricciones codificando mediante expresiones booleanas algoritmos de suma y multiplicación.

Adicionalmente se pierden optimizaciones posibles, por ejemplo los resolvidores SAT en principio no saben que $x \leq x^2$, no aplican ese conocimiento al momento de resolver el problema. Un resolvidor SMT si puede tener y aplicar esa información.

Veamos un ejemplo de instancia en formato SMT-LIB 2.0 [18] para resolver el problema anterior.

```
(set-option :produce-models true)

(declare-const X Int)
(declare-const Y Int)
(declare-const Z Int)
(declare-const K Int)
(declare-const R Int)

(assert (= (+ (* X X) (* 2 (+ Y 3)) (* Z Z Z)) 4163))
(assert (=> (= X (+ (* 2 K) 1)) (= Y (* 2 R))))

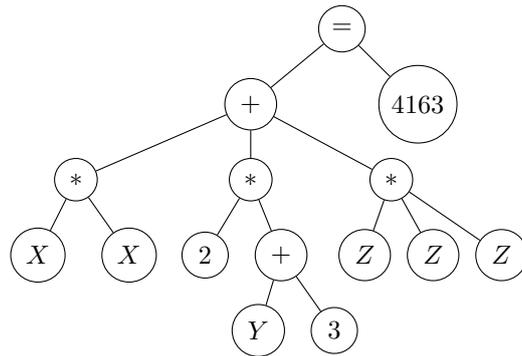
(check-sat)
(get-model)
```

Listado 7.1: Instancia SMT.

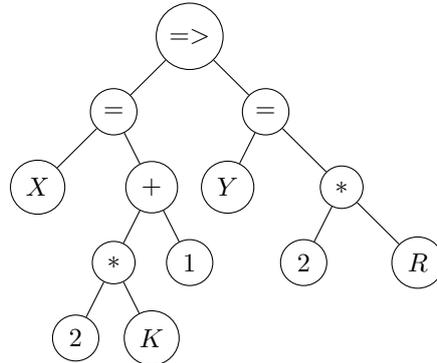
Un resolutor SMT aplica una teoría para resolver una instancia. Es posible especificarla manualmente, pero también se puede inferir de la instancia. En este caso es la teoría de los números enteros.

La sintaxis tiene un formato parecido al lenguaje de programación LISP y como se aprecia es bastante natural, con la excepción de las expresiones dentro de los assert. Se utiliza una notación parecida a la notación polaca, en donde el operador se encuentra al principio. Se muestran a continuación los árboles de las 2 expresiones assert.

```
(assert (= (+ (* X X) (* 2 (+ Y 3)) (* Z Z Z))) 4163))
```



```
(assert (=> (= X (+ (* 2 K) 1)) (= Y (* 2 R))))
```



Para resolver las instancias utilizaremos los programas Z3 [19] y OptiMathSat [21]. El ejemplo del problema anterior puede resolverse guardándolo en un archivo de nombre **prueba.smt** y ejecutando el siguiente comando.

```
z3 prueba.smt
```

La salida consiste en si es satisfacible o no la instancia, en caso de ser satisfacible se devuelve el modelo.

```

sat
(model
  (define-fun R () Int
    (- 4094))
  (define-fun K () Int
    (- 4096))
  (define-fun Z () Int
    21)
  (define-fun Y () Int
    (- 2677))
  (define-fun X () Int
    (- 16))
)

```

Listado 7.2: Resultado de la ejecución.

Los resolvers SMT soportan números reales, pero en esta tesis solo trabajaremos con números enteros.

Cuando en las expresiones se utilizan números enteros y términos no lineales, en el caso general no existe solución algorítmica, por lo que no siempre es posible obtener una solución.

Z3 y OptiMathSat soportan la optimización de términos lineales [20]. Esto evita la necesidad de realizar llamadas múltiples variando el rango de valores posibles de una variable.

7.2. Instancias para calcular thinness

Se presenta como generar instancias para encontrar el valor de thinness y thinness propia de un grafo.

Asumimos que los vértices del grafo se encuentran con una numeración arbitraria y fija v_1, v_2, \dots, v_n siendo $n = |V(G)|$.

La instancia SMT se genera partiendo de un grafo concreto, por lo que se define al grafo mediante constantes que indican la existencia o no de una arista. Existen tantas constantes como pares de vértices no ordenados y se escribe un solo assert en la instancia SMT por cada par.

```

(declare-const EiAj Bool)

(assert (= EiAj true))      ; Si  $v_i v_j$  en  $E(G)$ 
(assert (= EiAj false))    ; Si  $v_i v_j$  no en  $E(G)$ 

```

$$1 \leq i < j \leq n$$

Debe tenerse en cuenta que como los valores anteriores se encuentran fijos, los resolvers los optimizan reemplazando el valor de la constante en donde se utilice.

La clases de cada vértice se almacenan en variables Z_i .

```
(declare-const Zi Int)
(assert (and (>= Zi 1) (<= Zi K)))
```

 $1 \leq i \leq n$

La posición de cada vértice se guarda en una variable.

```
(declare-const Xi Int)
(assert (and (>= Xi 1) (<= Xi N)))
```

 $1 \leq i \leq n$

Como ejemplo se muestra cuales son los valores para los siguientes órdenes.

$[v_1, v_3, v_2, v_4]$	$X_1 = 1$	$X_2 = 3$	$X_3 = 2$	$X_4 = 4$
$[v_4, v_1, v_2, v_3]$	$X_1 = 2$	$X_2 = 3$	$X_3 = 4$	$X_4 = 1$

Se debe asegurar que las posiciones no se repitan.

```
(assert (not (= Xi Xj)))
```

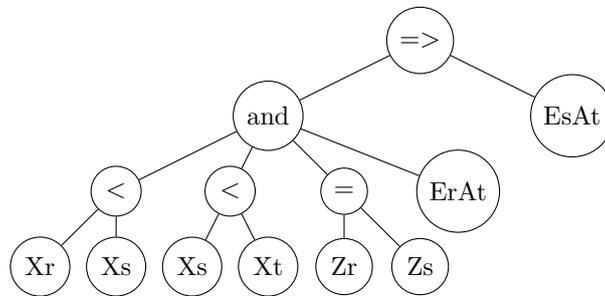
 $1 \leq i < j \leq n$

Finalmente se especifica la definición de thinness. Dado 3 vértices r, s, t es necesario asegurarse que la definición se cumpla para los 6 posibles ordenamientos.

```
(assert (=> (and (< Xr Xs) (< Xs Xt) (= Zr Zs) ErAt) EsAt))
(assert (=> (and (< Xr Xt) (< Xt Xs) (= Zr Zt) ErAs) EsAt))
(assert (=> (and (< Xs Xr) (< Xr Xt) (= Zs Zr) EsAt) ErAt))
(assert (=> (and (< Xs Xt) (< Xt Xr) (= Zs Zt) ErAs) ErAt))
(assert (=> (and (< Xt Xr) (< Xr Xs) (= Zt Zr) EsAt) ErAs))
(assert (=> (and (< Xt Xs) (< Xs Xr) (= Zt Zs) ErAt) ErAs))
```

 $1 \leq r < s < t \leq n$

El árbol para el primer assert es el siguiente:



En el caso de que se este calculando la thinness propia es necesario comprobar adicionalmente en la dirección opuesta.

```
(assert ( $\Rightarrow$ ) (and (< Xr Xs) (< Xs Xt) (= Zs Zt) ErAt) ErAs))
(assert ( $\Rightarrow$ ) (and (< Xr Xt) (< Xt Xs) (= Zt Zs) ErAs) ErAt))
(assert ( $\Rightarrow$ ) (and (< Xs Xr) (< Xr Xt) (= Zr Zt) EsAt) ErAs))
(assert ( $\Rightarrow$ ) (and (< Xs Xt) (< Xt Xr) (= Zt Zr) ErAs) EsAt))
(assert ( $\Rightarrow$ ) (and (< Xt Xr) (< Xr Xs) (= Zr Zs) EsAt) ErAt))
(assert ( $\Rightarrow$ ) (and (< Xt Xs) (< Xs Xr) (= Zs Zr) ErAt) EsAt))
```

$$1 \leq r < s < t \leq n$$

El rango de valores posibles de *thinness* puede acotarse para reducir el tiempo de búsqueda.

```
(declare-const K Int)
(assert (and ( $\geq$  K 1) ( $\leq$  K n)))
```

Los últimos comandos instruyen al resolutor que es lo que debe minimizarse.

```
(minimize K)
(check-sat)
```

Mostramos un ejemplo completo de una instancia en el apéndice.

7.3. Instancias para boxicity 2

Estas instancias determinan si el grafo tiene boxicity menor o igual a 2, con las cajas ubicadas en las diagonales. Las cajas se generan en forma alineada. No se ha desarrollado una instancia para calcular la boxicity en general.

En forma similar a las instancias de thinness se utiliza una variable para representar una arista posible en el grafo.

```
(declare-const EiAj Bool)

(assert (= EiAj true))      ; Si vi vj en E(G)
(assert (= EiAj false))    ; Si vi vj no en E(G)
```

$$1 \leq i < j \leq n$$

A cada vértice del grafo le corresponde una caja, que se representa con 4 valores enteros X_1, Y_1, X_2, Y_2 que cumplen las propiedades $1 \leq X_1, 1 \leq Y_1, X_1 < X_2$ y $Y_1 < Y_2$.

```
(declare-const BiX1 Int)
(declare-const BiX2 Int)
(declare-const BiY1 Int)
(declare-const BiY2 Int)

(assert (and (<= 1 BiX1) (<= 1 BiY1)
             (< BiX1 BiX2) (< BiY1 BiY2) ))
```

$$1 \leq i \leq n$$

La arista existe si y sólo si las cajas tienen intersección. En este caso se puede igualar directamente el resultado de la conjunción con el valor del booleano de la arista.

```
(assert (= EiAj (and (< BjY1 BiY2) (> BjY2 BiY1)
                    (< BjX1 BiX2) (> BjX2 BiX1) )))
```

$$1 \leq i < j \leq n$$

Se agrega la condición de que las esquinas $X_2 Y_2$ se encuentren a lo largo de 2 líneas diagonales. Se identificará a las 2 líneas diagonales con los valores $D1Y D2Y$, que representan los puntos de intersección con el eje Y. Esto significa que los puntos $X=0, Y=D1Y$ y $X=0, Y=D2Y$ pertenecen cada uno a la línea diagonal.

```
(declare-const D1Y Int)
(declare-const D2Y Int)

(assert (not (= D1Y D2Y)))
```

2 puntos $X_1, Y_1 X_2, Y_2$ se encuentran en la misma diagonal si $Y_1 - X_1 = Y_2 - X_2$ por lo que la condición de que las esquinas superiores derechas de cada caja se encuentren en una de las 2 diagonales es la siguiente:

```
(assert (or (= D1Y (- BiX2 BiY2)) (= D2Y (- BiX2 BiY2))))
```

$$1 \leq i \leq n$$

En este caso nos aseguramos que las esquinas X2,Y2 no se repitan,.

```
(assert (not (and (= BiX2 BjX2) (= BiY2 BjY2))))
```

$$1 \leq i < j \leq n$$

Usamos una variable para almacenar en que diagonal se encuentra cada caja.

```
(declare-const BiV1 Bool)
(assert (= BiV1 (= DiY (- BiY2 BiX2))))
```

$$1 \leq i \leq n$$

Se codifica que las cajas deben encontrarse en el cuadrante correcto.

```
(declare-const X0 Int)
(declare-const Y0 Int)
(assert (> X0 0))
(assert (> Y0 0))
```

```
(assert (=> BiV1 (and (> BiX2 X0) (< BiY2 Y0))))
(assert (=> (not BiV1) (and (< BiX2 X0) (> BiY2 Y0))))
```

$$1 \leq i \leq n$$

Como comprobación final, se debe verificar que las cajas sean alineadas.

```
(assert (=> (and (not BiV1) BjV1)
            (or (< BjX1 BiX2) (< BiY1 BjY2) )))
```

$$1 \leq i \leq n \quad 1 \leq j \leq n \quad i \neq j$$

No es necesario realizar una optimización.

```
(check-sat)
```

7.4. Performance

Se evaluará la performance de los resolvedores utilizando las instancias anteriores. Todas las ejecuciones fueron realizadas con un procesador Core i5 de 4 núcleos a 3.20 GHz y 16 GB de RAM. Se utilizan las versiones Z3 4.8.7 y OptiMathSat 1.6.3.

Caso 1 - Grafo de grilla

El primer caso es utilizando el grafo de grilla.

Grilla	$ V $	$ E $	Thinness	Z3	OMS
2×2	4	4	2	0.01 s	0.01 s
3×3	9	12	2	0.06 s	0.09 s
4×4	16	24	3	1.30 s	0.88 s
4×5	20	31	3	4.22 s	2.02 s
3×7	21	32	2	1.66 s	2.21 s
4×6	24	38	3	19.49 s	3.32 s
5×5	25	40	3	38.34 s	5.67 s
4×7	28	45	3	82.53 s	7.19 s
5×6	30	49	3	219.73 s	10.09 s
4×8	32	52	3	230.28 s	12.80 s
5×7	35	58	3	1207.05 s	29.65 s
6×6	36	60	4	3647.89 s	244.20 s

Tab. 7.1: Tiempos de cálculo de thinness para grafos de grilla.

Grilla	$ V $	$ E $	Thinness propia	Z3	OMS
2×2	4	4	2	0.01 s	0.01 s
3×3	9	12	3	0.06 s	0.10 s
4×4	16	24	3	1.10 s	1.04 s
4×5	20	31	3	7.12 s	2.41 s
3×7	21	32	3	5.44 s	2.94 s
4×6	24	38	3	27.98 s	5.42 s
5×5	25	40	4	67.13 s	9.61 s
4×7	28	45	3	98.19 s	8.67 s
5×6	30	49	4	227.57 s	17.23 s
4×8	32	52	3	333.96 s	16.06 s
5×7	35	58	4	523.71 s	37.15 s
6×6	36	60	4	*	55.20 s

Tab. 7.2: Tiempos de cálculo de thinness propia para grafos de grilla.

* No se ha podido terminar la ejecución por tiempo excesivo.

Caso 2 - Grafo T_{2n}

Como segundo caso de pruebas se usa el grafo T_{2n} definido de la siguiente manera.

$$V(T_{2n}) = \{x_1, x_2, \dots, x_n\} \cup \{y_1, y_2, \dots, y_n\}$$

$$E(T_{2n}) = \{x_i y_j : i \neq j\} \cup \{x_i x_j : i \neq j\} \cup \{y_i y_j : i \neq j\}$$

El grafo T_{2n} tiene thinness n [3].

Grafo	$ V $	Z3	OMS
T_4	4	0.01 s	0.01 s
T_6	6	0.02 s	0.01 s
T_8	8	0.12 s	0.10 s
T_{10}	10	4.12 s	1.73 s
T_{12}	12	951.93 s	184.25 s

Tab. 7.3: Tiempos de cálculo de thinness para T_{2n} .

Caso 3 - Grafo de Mycielski

Se define la secuencia de grafos de Mycielski en forma recursiva.

$$M_2 = K_2$$

$$M_{x+1} = \mu(M_x)$$

Dado $G = (V, E)$ con $V = \{v_1, v_2, \dots, v_n\}$ se define $\mu(G) = (V', E')$ como:

$$V' = V \cup \{w\} \cup \{u_1, u_2, \dots, u_n\}$$

$$E' = E \cup \{u_i w : 1 \leq i \leq n\} \cup \{u_i v_j : v_i v_j \in E\} \cup \{v_i u_j : v_i v_j \in E\}$$

Grafo	$ V $	Thinness	Z3	OMS
M_3	5	2	0.01 s	0.01 s
M_4	11	4	1.45 s	0.01 s
M_5	23	5	2251.18 s	620.13 s

Tab. 7.4: Tiempos de cálculo de thinness el grafo de Mycielski.

Como se puede apreciar, es posible obtener la thinness con SMT, pero para grafos de una cantidad acotada de vértices. En general OptiMathSat logra mejores resultados que Z3.

Rango de variable K

Una cuestión que debe tenerse en cuenta en el caso de las instancias para calcular *thinness* es el rango del parámetro K en la búsqueda.

```
(assert (and (>= K 1) (<= K T)))
```

Esto puede tener un impacto grande en el tiempo de ejecución. En todos los casos anteriores se utilizó $T = |V(G)|$. Como ejemplo, en el grafo de grilla de 5 x 5, que tiene *thinness* 3, se obtienen los siguientes tiempos de ejecución en función del valor de T. Se utiliza Z3.

T	Tiempo de ejecución
25	37.74 s
12	31.22 s
3	15.26 s
2	0.66 s

Tab. 7.5: Tiempos de cálculo de *thinness* según el valor de T.

En el caso $T = 2$ el resolvidor determina rápidamente que no existe solución 2-thin.

Comparación entre *thinness* y *boxicity*

Se compara la eficiencia entre las instancias para *thinness* y las instancias para *boxicity*. Como las instancias para *boxicity* no pueden calcular grafos con *thinness* mayor a 2, se va a tener en cuenta de comparar en la forma más parecida posible limitando el rango de K. El grafo que se va a utilizar es el del apéndice, que tiene 72 vértices y *thinness* 3. Se utiliza OptiMathSat.

Instancia	Tiempo de ejecución	Resultado
Thinness $T = 2$	11 s	No satisfacible
Thinness $T = 3$	31 s	Satisfacible
Boxicity	93 s	No satisfacible

Tab. 7.6: Tiempos de cálculo de *thinness* con distintas instancias.

Se observa que es más eficiente hacer el cálculo con las instancias de *thinness*, puede deberse a la mayor cantidad de variables al usar las cajas.

8. CONCLUSIONES Y TRABAJO FUTURO

Se ha logrado un progreso en el entendimiento de la thinness de un grafo. La relación entre la thinness y la boxicity se puede apreciar con mayor facilidad, al menos para los valores iguales a 2. Se ha avanzado en caracterizaciones basadas en matrices y fue posible establecer la comparación de la thinness con el bandwidth. Hemos presentado una manera de calcular la thinness y la thinness propia para grafos con una cantidad reducida de vértices.

El bandwidth de un grafo es una invariante muy conocida con varios resultados que pueden utilizarse para obtener más propiedades para thinness.

El problema de decidir si la boxicity de un grafo es menor o igual 2 es NP-Completo, sería interesante ver si esto se mantiene cuando las cajas se encuentran en 2 diagonales. La reducción conocida es no trivial [22, 23].

Sería interesante analizar qué resultados se pueden obtener partiendo desde el modelo de intersección de cajas. Adicionalmente hay muchas representaciones geométricas de grafos que pueden compararse con el modelo de intersección de cajas. Deben analizarse también grafos k -thin con valores de k mayores a 2.

Resta encontrar un algoritmo de tiempo polinomial o una demostración de que el problema de calcular la thinness expresado como problema de decisión es NP-Completo. Se encuentra demostrado que cuando la partición es fija y solo se busca un orden consistente el problema es NP-Completo [3].

Resta mucho trabajo en mejorar la eficiencia del cálculo de la thinness.

Apéndice

.1. Contraejemplo diagonal con thinness 3

Sea G el grafo definido de la siguiente manera, $V(G) = A \cup B$, en donde:

$$A = \{a_1, a_2, \dots, a_{35}, a_{36}\} = A_0 \cup A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5$$

$$B = \{b_1, b_2, \dots, b_{35}, b_{36}\} = B_0 \cup B_1 \cup B_2 \cup B_3 \cup B_4 \cup B_5$$

$$A_0 = \{a_i : 1 \leq i \leq 32, i \text{ impar}\} \quad A_5 = \{a_{33}, a_{34}, a_{35}, a_{36}\}$$

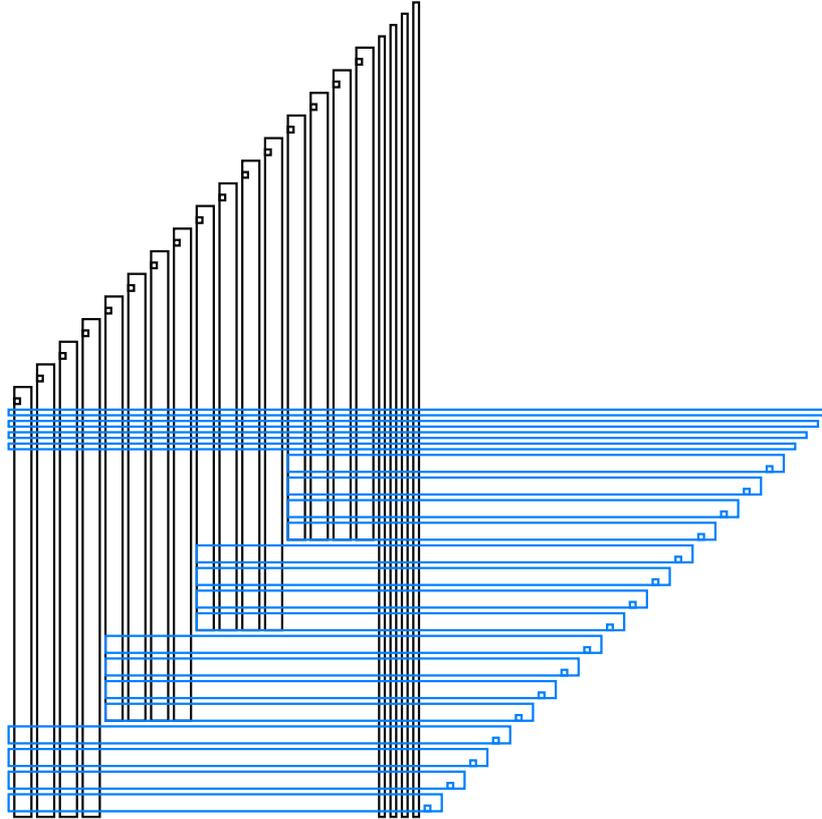
$$A_k = \{a_i : 8(k-1) < i \leq 8k, i \text{ par}\} \quad 1 \leq k \leq 4$$

$$B_j = \{b_i : a_i \in A_j\} \quad 0 \leq j \leq 5$$

Las aristas entre A y B son tales que: para $1 \leq j \leq 4$, A_j es completo con B_j y B_5 , y anticompleto con B_0 y B_i , $1 \leq i \leq 4$, $i \neq j$; A_5 es anticompleto con B_0 y completo con $B \setminus B_0$. Adicionalmente, $a_{2k-1}a_{2k} \in E(G)$ y $b_{2k-1}b_{2k} \in E(G)$, para $1 \leq k \leq 16$, y estas son las únicas aristas internas de A y B .

Proposición A El grafo G tiene una representación como una intersección de cajas con la propiedad diagonal.

Demostración. La representación es la siguiente: la esquina superior derecha de a_i es $(i, i+36)$ para $1 \leq i \leq 36$; las esquinas inferiores izquierda son, para a_i en A_0 , $(i-0.5, i+35.5)$, para a_i en A_1 , $(i-1.5, 0)$, para a_i en A_2 , $(i-1.5, 8.5)$, para a_i en A_3 , $(i-1.5, 16.5)$, para a_i en A_4 , $(i-1.5, 24.5)$, para a_i en A_5 , $(i-0.5, 0)$. Si la esquinas inferiores izquierdas y superiores derechas de a_i son (x, y) y (w, z) respectivamente, entonces las esquinas inferiores izquierdas y superiores derechas de b_i son (y, x) y (z, w) respectivamente. No es difícil verificar que esta es una representación de G . \square



Lema B Sea H un grafo bipartito completo con bipartición (A, B) . En toda representación 2-thin de H , excepto tal vez para el mayor vértice de A y el mayor vértice de B (de acuerdo al orden asociado con la representación), todo vértice de A se encuentra en una clase y todo vértice de B se encuentra en la otra clase.

Demostración. Sea $<$ el orden asociado con la representación 2-thin de H . Sea a_M (resp. b_M) el mayor vértice de A (resp. B) de acuerdo con $<$. Por simetría del grafo (dado que la cantidad de elementos de A y B no se encuentran especificados en el lema), podemos asumir sin pérdida de generalidad que $a_M > b_M$.

Se demuestra que $A \setminus \{a_M\}$ es completo con $B \setminus \{b_M\}$ en $G_{<}$.

Sea $a \in A \setminus \{a_M\}$, $b \in B$ tales que $b < a$. Entonces, $b < a < a_M$, $a_M b \in E(G)$ y $a_M a \notin E(G)$, por lo tanto $ab \in E(G_{<})$. Sea $b \in B \setminus \{b_M\}$, $a \in A$ tal que $a < b$. Entonces, $a < b < b_M$, $b_M a \in E(G)$ y $b_M b \notin E(G)$, por lo que se obtiene $ab \in E(G_{<})$. Debido a lo anterior, $A \setminus \{a_M\}$ es completo con $B \setminus \{b_M\}$ en $G_{<}$.

Por lo anterior, $A \setminus \{a_M\}$ y $B \setminus \{b_M\}$ se encuentran en conjuntos distintos de la partición, y dado que hay solamente dos conjuntos en la partición, el lema queda demostrado. \square

Proposición C El grafo G tiene thinness 3.

Demostración. Sea $V_1 = A \setminus A_0$, $V_2 = B \setminus B_0$, $V_3 = A_0 \cup B_0$ y el orden dado por $a_1, a_2, b_1, b_2, a_3, a_4, b_3, b_4, \dots, a_{35}, a_{36}, b_{35}, b_{36}$. No es difícil de ver que el orden y la partición son consistentes.

Supongamos que G admite una representación 2-thin $(<, V_1, V_2)$. Notar que $(A_j \cup A_5, B_j \cup B_5)$ induce un grafo completo bipartito, para $j = 1, \dots, 4$. Entonces, por Lema B y transitividad, excepto tal vez para algunos pocos vértices, los vértices de $A \setminus A_0$ se encuentran en uno de los conjuntos de la partición, supongamos V_1 , y los vértices de $B \setminus B_0$ se encuentran en el otro conjunto, V_2 . Sean A' los vértices de $A \setminus A_0$ que se encuentran en V_2 y B' los vértices de $B \setminus B_0$ que se encuentran en V_1 . Para $j = 1, \dots, 5$, sea $A'_j = A_j \setminus A'$ y $B'_j = B_j \setminus B'$.

Para $1 \leq j \leq 5$, sean a_M^j, a_S^j y b_S^j, b_M^j los vértices mayores y menores de A'_j y B'_j , respectivamente.

Sean $\{i, j, k, \ell\} = \{1, 2, 3, 4\}$. Entonces, para cada vértice $a \in A'_j \cup A'_k \cup A'_\ell$, vale uno de los casos $a > b_M^i$ o $a < a'$ para cada $a' \in A'_i$, y en forma análoga, para cada vértice $b \in B'_j \cup B'_k \cup B'_\ell$, vale uno de los casos $b > a_M^i$ o $b < b'$ para cada $b' \in B'_i$.

Por simetría del grafo G , se puede asumir que $a_M^1 < a_M^2 < a_M^3 < a_M^4$ y $a_M^4 > \max\{b_M^1, b_M^2, b_M^3, b_M^4\}$. Por la observación de arriba, para cada $b \in B'_3 \cup B'_2 \cup B'_1$ y $b' \in B'_4$, se cumple que $b < b'$. En particular, $b_M^4 > b_S^4 > \max\{b_M^3, b_M^2, b_M^1\}$. Supongamos que $a_M^3 > b_S^4$. Entonces $a_M^3 > b_S^4 > b_M^3$, $a_M^3 b_M^3 \in E(G)$ y $a_M^3 b_S^4 \notin E(G)$, una contradicción. Entonces $a_M^3 < b_S^4 < b_M^4$, y por lo tanto para cada $a \in A'_3 \cup A'_2 \cup A'_1$ y $a' \in A'_4$, se cumple que $a < a'$, es decir, $a_S^4 > a_M^3$.

Supongamos que $b_M^3 > a_S^4$. Entonces $b_M^3 > a_S^4 > a_M^3$, $a_M^3 b_M^3 \in E(G)$ y $b_M^3 a_S^4 \notin E(G)$, una contradicción. Entonces $b_M^3 < a_S^4$. Supongamos ahora que $b_M^2 > a_M^3$. Entonces $b_M^2 > a_M^3 > a_M^2$, $b_M^2 a_M^2 \in E(G)$ y $b_M^2 a_M^3 \notin E(G)$, una contradicción. Entonces $b_M^2 < a_M^3$, y por lo tanto para cada $b \in B'_2 \cup B'_1$ y $b' \in B'_3$, se cumple que $b < b'$, es decir, $b_S^3 > \max\{b_M^2, b_M^1\}$.

Supongamos que $a_M^2 > b_S^3$. Entonces $a_M^2 > b_S^3 > b_M^2$, $a_M^2 b_M^2 \in E(G)$ y $a_M^2 b_S^3 \notin E(G)$, una contradicción. Entonces $a_M^2 < b_S^3 < b_M^3$, y por lo tanto para cada $a \in A_2' \cup A_1'$ y $a' \in A_3'$, se cumple que $a < a'$, es decir, $a_M^2 < a_S^3$. Supongamos que $b_M^2 > a_S^3$. Entonces $b_M^2 > a_S^3 > a_M^2$, $a_M^2 b_M^2 \in E(G)$ y $b_M^2 a_S^3 \notin E(G)$, una contradicción. Entonces $b_M^2 < a_S^3$.

Supongamos que $b_M^1 > a_M^2$. Entonces $b_M^1 > a_M^2 > a_M^1$, $b_M^1 a_M^1 \in E(G)$ y $b_M^1 a_M^2 \notin E(G)$, una contradicción. Entonces $b_M^1 < a_M^2$, y por lo tanto para cada $b \in B_1'$ y $b' \in B_2'$, se cumple que $b < b'$, es decir, $b_S^2 > b_M^1$. Supongamos que $b_M^1 > a_S^2$. Entonces $b_M^1 > a_S^2 > a_M^1$, $a_M^1 b_M^1 \in E(G)$ y $b_M^1 a_S^2 \notin E(G)$, una contradicción. Entonces $b_M^1 < a_S^2$.

Finalmente, supongamos que $a_M^1 > b_S^2$. Entonces $a_M^1 > b_S^2 > b_M^1$, $a_M^1 b_M^1 \in E(G)$ and $a_M^1 b_S^2 \notin E(G)$, una contradicción. Entonces $a_M^1 < b_S^2 < b_M^2$, y por lo tanto para cada $a \in A_1'$ y $a' \in A_2'$, se cumple que $a < a'$, es decir, $a_M^1 < a_S^2$. Supongamos que $b_M^1 > a_S^2$. Entonces $b_M^1 > a_S^2 > a_M^1$, $a_M^1 b_M^1 \in E(G)$ y $b_M^1 a_S^2 \notin E(G)$, una contradicción. Entonces $b_M^1 < a_S^2$.

Entonces, $A_1' < A_2' < A_3' < A_4'$, $B_1' < B_2' < B_3' < B_4'$, y $\max\{a_M^1, b_M^1\} < \min\{a_S^2, b_S^2\} < \max\{a_M^2, b_M^2\} < \min\{a_S^3, b_S^3\} < \max\{a_M^3, b_M^3\} < \min\{a_S^4, b_S^4\} < b_M^4 < a_M^4$.

Los vértices en B_5' tienen que ser mayores que b_M^3 , que es un no vecino de a_M^4 y menor que este. De manera similar, los vértices en A_5' tienen que ser mayores que a_M^3 .

Sean a_2^2, a_3^2 el segundo y tercer vértice mayor de A_2' respectivamente. Sea a_0 el vecino de a_2^2 en A_0 . Supongamos primero que $a_0 \in V_1$. Si $a_0 > a_2^2$, entonces $a_0 < a_M^2$, porque $a_0 a_M^2 \notin E(G)$. Si $a_0 < a_2^2$, entonces $a_0 > a_3^2$, porque $a_2^2 a_3^2 \notin E(G)$. Sea $b^5 \in B_5'$. Entonces $b^5 > b_M^3 > a_M^2 > a_0 > a_S^2$, pero $b^5 a_S^2 \in E(G)$ y $b^5 a_0 \notin E(G)$, una contradicción.

Supongamos que $a^0 \in V_2$. Si $a^0 > a_2^2$, entonces $a^0 < a_M^2$, porque $a^0 a_M^2 \notin E(G)$. Si $a^0 < a_2^2$, entonces $a^0 > b_M^1$, porque $a_2^2 > b_M^1$ y $a_2^2 b_M^1 \notin E(G)$. Sea $a^5 \in A_5'$. Entonces $a^5 > a_M^3 > a_2^2 > a^0 > b_M^1$, pero $a^5 b_M^1 \in E(G)$ y $a^5 a^0 \notin E(G)$, una contradicción. \square

.2. Instancia SMT para calcular thinness de C4

```
(set-option :produce-models true)
```

```
(declare-const K Int)
(assert (and (>= K 1) (<= K 4)))
```

```
-----
; Aristas
-----
```

```
(declare-const E1A2 Bool)
(declare-const E1A3 Bool)
(declare-const E1A4 Bool)
(declare-const E2A3 Bool)
(declare-const E2A4 Bool)
(declare-const E3A4 Bool)
```

```
(assert (= E1A2 true))
(assert (= E1A3 false))
(assert (= E1A4 true))
(assert (= E2A3 true))
(assert (= E2A4 false))
(assert (= E3A4 true))
```

```
-----
; Particion
-----
```

```
(declare-const Z1 Int)
(declare-const Z2 Int)
(declare-const Z3 Int)
(declare-const Z4 Int)
```

```
(assert (and (>= Z1 1) (<= Z1 K)))
(assert (and (>= Z2 1) (<= Z2 K)))
(assert (and (>= Z3 1) (<= Z3 K)))
(assert (and (>= Z4 1) (<= Z4 K)))
```

```
-----
; Orden
-----
```

```
(declare-const X1 Int)
(declare-const X2 Int)
(declare-const X3 Int)
(declare-const X4 Int)
```

```

(assert (and (>= X1 1) (<= X1 4)))
(assert (and (>= X2 1) (<= X2 4)))
(assert (and (>= X3 1) (<= X3 4)))
(assert (and (>= X4 1) (<= X4 4)))
(assert (not (= X1 X2)))
(assert (not (= X1 X3)))
(assert (not (= X1 X4)))
(assert (not (= X2 X3)))
(assert (not (= X2 X4)))
(assert (not (= X3 X4)))

;-----
; Thinness
;-----

(assert (=> (and (< X1 X2) (< X2 X3) (= Z1 Z2) E1A3) E2A3))
(assert (=> (and (< X1 X3) (< X3 X2) (= Z1 Z3) E1A2) E2A3))
(assert (=> (and (< X2 X1) (< X1 X3) (= Z2 Z1) E2A3) E1A3))
(assert (=> (and (< X2 X3) (< X3 X1) (= Z2 Z3) E1A2) E1A3))
(assert (=> (and (< X3 X1) (< X1 X2) (= Z3 Z1) E2A3) E1A2))
(assert (=> (and (< X3 X2) (< X2 X1) (= Z3 Z2) E1A3) E1A2))

(assert (=> (and (< X1 X2) (< X2 X4) (= Z1 Z2) E1A4) E2A4))
(assert (=> (and (< X1 X4) (< X4 X2) (= Z1 Z4) E1A2) E2A4))
(assert (=> (and (< X2 X1) (< X1 X4) (= Z2 Z1) E2A4) E1A4))
(assert (=> (and (< X2 X4) (< X4 X1) (= Z2 Z4) E1A2) E1A4))
(assert (=> (and (< X4 X1) (< X1 X2) (= Z4 Z1) E2A4) E1A2))
(assert (=> (and (< X4 X2) (< X2 X1) (= Z4 Z2) E1A4) E1A2))

(assert (=> (and (< X1 X3) (< X3 X4) (= Z1 Z3) E1A4) E3A4))
(assert (=> (and (< X1 X4) (< X4 X3) (= Z1 Z4) E1A3) E3A4))
(assert (=> (and (< X3 X1) (< X1 X4) (= Z3 Z1) E3A4) E1A4))
(assert (=> (and (< X3 X4) (< X4 X1) (= Z3 Z4) E1A3) E1A4))
(assert (=> (and (< X4 X1) (< X1 X3) (= Z4 Z1) E3A4) E1A3))
(assert (=> (and (< X4 X3) (< X3 X1) (= Z4 Z3) E1A4) E1A3))

(assert (=> (and (< X2 X3) (< X3 X4) (= Z2 Z3) E2A4) E3A4))
(assert (=> (and (< X2 X4) (< X4 X3) (= Z2 Z4) E2A3) E3A4))
(assert (=> (and (< X3 X2) (< X2 X4) (= Z3 Z2) E3A4) E2A4))
(assert (=> (and (< X3 X4) (< X4 X2) (= Z3 Z4) E2A3) E2A4))
(assert (=> (and (< X4 X2) (< X2 X3) (= Z4 Z2) E3A4) E2A3))
(assert (=> (and (< X4 X3) (< X3 X2) (= Z4 Z3) E2A4) E2A3))

(minimize K)
(check-sat)
(get-value (K))
(get-value (Z1 Z2 Z3 Z4 ))
(get-value (X1 X2 X3 X4 ))

```

Bibliografía

- [1] Carlo Mannino, Gianpaolo Oriolo, Federico Ricci, Sunil Chandran, The Stable Set Problem and the Thinness of a Graph, *Operations Research Letters* 35, 2007, 1–9.
- [2] Stephan Olariu, On the homogeneous representation of interval graphs, *Journal of Graph Theory*, Volume 15, 1991, 65–80.
- [3] Flavia Bonomo, Diego de Estrada, On the thinness and proper thinness of a graph, *Discrete Applied Mathematics* 261, 2019, 78–92.
- [4] Carlo Mannino, Gianpaolo Oriolo, Solving Stability Problems on a Superclass of Interval Graphs, *manuscrito*.
- [5] Douglas West, *Introduction to Graph Theory*, Prentice Hall, 2nd edition, 2000.
- [6] Donald Knuth, *The Art of Computer Programming*, volume 1. Addison–Wesley, Reading, MA, 1968.
- [7] Lucía Rabinowicz, Sobre la thinness de árboles, Tesis de licenciatura en Ciencia de la computación, Universidad de Buenos Aires, 2019.
- [8] William T. Trotter Jr., Frank Harary, On Double and Multiple Interval Graphs, *Journal of Graph Theory* 3, 1979, 205–211.
- [9] Michael Dom, Algorithmic Aspects of the Consecutive-Ones Property, *Bulletin of the European Association for Theoretical Computer Science* 98, 2009, 27–59.
- [10] Delbert R. Fulkerson, Oliver A. Gross, Incidence matrices and interval graphs, *Pacific Journal Of Mathematics*, Vol. 15, No. 3, 1965, 835–855.
- [11] Michael Garey, David S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [12] Withold Lipski Jr, Generalizations of the consecutive ones property and related NP-complete problems, *Coordinated Science Laboratory Report no. T-67*, University of Illinois at Urbana-Champaign, 1978, 53–69.
- [13] Phyllis Z. Chinn, Jarmila Chvátalová, Alexander K. Dewdney, Norman E. Gibbs, The bandwidth problem for graphs and matrices—a survey, *Journal of Graph Theory* 6, 1982, 223–254.
- [14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*, Second Edition, The MIT Press.
- [15] Fred S. Roberts, On the boxicity and cubicity of a graph. In W.T. Tutte, editor, *Recent Progress in Combinatorics*, 301–310. Academic Press, 1969.
- [16] Margaret B. Cozzens, Fred S. Roberts, Computing the boxicity of a graph by covering its complement by cointerval graphs, *Discrete Applied Mathematics* 6, 1983, 217–228.

-
- [17] Carsten Thomassen, Interval Representations of Planar Graphs, *Journal of Combinatorial Theory, Series B* 40, 1986, 9–20.
- [18] Clark Barrett, Aaron Stump, Cesare Tinelli, The SMT-LIB Standard - Version 2.0, *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories* (Edinburgh, England), eds: A. Gupta and D. Kroening, 2010.
- [19] Leonardo Mendonça De Moura, Nikolaj Skallerud Bjørner, Z3: an efficient SMT solver, *TACAS'08/ETAPS'08: Proceedings of the Theory and practice of software, 14th international conference on Tools and algorithms for the construction and analysis of systems*, *Lecture Notes in Computer Science*, vol 4963. Springer, 2008, 337–340.
- [20] Nikolaj Bjørner, Anh-Dung Phan, Lars Fleckenstein, vZ - An Optimizing SMT Solver. In: Baier C., Tinelli C. (eds) *Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2015*. *Lecture Notes in Computer Science*, vol 9035. Springer, Berlin, Heidelberg, 2015, 194–199.
- [21] Roberto Sebastiani, Patrick Trentin, OptiMathSAT: A tool for Optimization Modulo Theories, *Proc. Int. Conf. on Computer Aided verification, CAV'15*, 2015.
- [22] Jan Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, *Discrete Applied Mathematics* 52, 1994, 233–252.
- [23] Stephen J. Bellantoni, Irith Ben-Arroyo Hartman, Teresa Przytycka, Sue Whitesides, Grid intersection graphs and boxicity, *Discrete Mathematics* 114, 1993, 41–49.