

Grafos y Optimización Discreta

Jose Antonio Lorenzo Abril

Contents

1	Conceptos básicos de Teoría de Grafos	3
1.1	Grafos dirigidos y no dirigidos	3
1.2	Algunos tipos de grafos. Subgrafos	3
1.3	Grado de incidencia	4
1.3.1	Secuencia gráfica	5
1.4	Camino y ciclo	9
1.5	Representaciones matriciales de grafos	10
1.5.1	Matriz de adyacencia	10
2	Conectividad en grafos	12
2.1	Conexión. Componentes conexas	12
2.2	Obtención de las componentes conexas	13
2.2.1	Mediante las potencias de la matriz de adyacencia	13
2.2.2	Breadth-first search - Búsqueda primero en anchura	13
2.2.3	Depth-first search - Búsqueda primero en profundidad	14
2.3	Conjuntos separadores	15
2.3.1	Conjuntos separadores de vértices. Vértices de corte	15
2.3.2	Conjuntos separadores de ejes. Puentes	16
2.4	Conectividad. k -conexión	18
2.4.1	Teorema de Menger	20
2.4.2	Resultados para conectividad por ejes	21
3	Árboles	22
4	Camino más corto. Recorrido por aristas y vértices	29
5	Coloración de grafos. Grafos planos	37
5.1	Coloración	37
5.1.1	Grafos críticos	38
5.1.2	Polinomio cromático	39
5.2	Planaridad	40
5.2.1	Identidad de Euler	40
5.2.2	Grafo plano maximal	42
5.2.3	Teorema de Kuratowski	42
5.3	Coloración en grafos. El teorema de los 4 colores	43

6	Programación lineal entera	44
6.1	El modelo de programación lineal entera	44
6.2	Matrices totalmente unimodulares	45
6.2.1	Resultados y caracterizaciones de matrices totalmente unimodulares	47
6.3	Formulación de modelos de programación lineal entera	48
7	Resolución de problemas de programación lineal entera	53
7.1	Método de ramificación y acotación	53
7.1.1	¿Cómo seleccionamos el nodo activo a resolver?	55
7.1.2	¿Cómo seleccionamos la variable a ramificar?	55
7.2	Método de los hiperplanos de corte	56
7.2.1	Hiperplano de corte entero	56
7.2.2	Hiperplano de corte para un problema entero mixto	57
7.2.3	Desigualdades de Chvátal-Gomory	58

1 Conceptos básicos de Teoría de Grafos

1.1 Grafos dirigidos y no dirigidos

Un **grafo** representa las relaciones entre un conjunto de objetos con respecto a una cierta característica. El conjunto de objetos (**vértices o nodos**) se representa por V . Para indicar cómo se relacionan, se toma $X \subset V \times V : (i, j) \in X \implies i$ está relacionado con j .

Un **grafo no dirigido** es aquel en que la relación es simétrica:

$$(i, j) \in X \implies (j, i) \in X$$

y en este caso los elementos de X se denominan **ejes**.

Un **grafo dirigido** es aquel en que la relación no es simétrica, y los elementos de X se denominan **arcos**.

Definition 1.1. Dado un grafo $G = (V, E)$ se denomina **orden de G** al número de vértices, $|V|$, y **tamaño de G** al número de ejes, $|E|$.

Un eje de la forma (i, i) se denomina **bucle**.

En ocasiones, pueden haber varios ejes conectando los mismos pares de vértices, estos ejes se denominan **ejes paralelos**. En caso de tener ejes paralelos, se dice que el grafo es un **multigrafo**.

Definition 1.2. Un **grafo simple** es un grafo sin ejes paralelos ni bucles.

A partir de ahora, mientras no se diga lo contrario, consideraremos grafos simples no dirigidos.

1.2 Algunos tipos de grafos. Subgrafos

Definition 1.3. Sea $G = (V, E)$ un grafo no dirigido:

- Se llama **grafo complementario de G** al grafo $G^c = (V, E^c)$ donde

$$E^c = \{(i, j) \in V \times V \mid i \neq j, (i, j) \notin E\}$$

- Se dice que G es **completo** si $\forall i, j \in V, i \neq j \implies (i, j) \in E$. Se denota K_n al grafo completo de orden n , y tiene tamaño $\binom{n}{2}$
- G es **bipartito** si se puede particionar el conjunto de vértices V en dos subconjuntos disjuntos $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ tales que

$$\forall (i, j) \in E \implies \{i \in V_1, j \in V_2 \text{ o bien } i \in V_2, j \in V_1\}$$

se denota $K_{n,m}$ al grafo completo bipartito con $|V_1| = n$, $|V_2| = m$, tiene tamaño $n \cdot m$

Definition 1.4. Sea $G = (V, E)$ un grafo no dirigido.

- $G' = (V', E')$ es un **subgrafo** de G si $V' \subset V$ y $E' \subset E \cap (V' \times V')$
- Si $V' = V$ se dice que G' es un **subgrafo generador** de G
- Si $V' \subset V$, el **subgrafo de G inducido por V'** es el grafo

$$G_{V'} = (V', E_{V'}), \quad E_{V'} = E \cap (V' \times V')$$

Definition 1.5. Un **cliqué** es un subgrafo completo de G . Un cliqué se dice **maximal** si no está contenido en ningún otro cliqué de G

Definition 1.6. Dos grafos simples $G = (V, E)$, $G' = (V', E')$ son **isomorfos** si existe una biyección

$$\varphi : V \rightarrow V'$$

que verifica

$$\forall u, v \in V, (u, v) \in E \iff (\varphi(u), \varphi(v)) \in E'$$

En tal caso, φ es un **isomorfismo de grafos**

1.3 Grado de incidencia

Definition 1.7. Sea $G = (V, E)$ un grafo no dirigido y $e = (i, j) \in E$, entonces se dice que:

- i, j son **vértices extremos** del eje e
- el eje e es **incidente** a los vértices i, j
- los vértices i, j son **adyacentes** entre sí

Definition 1.8. Se define el **grado de un vértice v** , $o(v)$ como el número de ejes incidentes a v .

Un nodo de grado 0 se dice que es **aislado**.

Un nodo de grado 1 se dice que es **hoja**.

El único eje incidente a una hoja se dice que es un **eje colgante**

Dado un grafo G , denotaremos

$$\delta_G = \min \{o(v) : v \in G\}$$

$$\Delta_G = \max \{o(v) : v \in G\}$$

Definition 1.9. Dado un grafo G , la secuencia formada por los grados de los vértices de G ordenados de forma decreciente se denomina **secuencia de grados de G**

Definition 1.10. Un grafo se dice **regular** si todos los vértices tienen el mismo grado

Theorem 1.11. *La suma de los grados de los vértices de un grafo es igual al doble del tamaño del grafo*

$$\sum_{v \in V} o(v) = 2 \cdot m$$

Proof. Si m es el tamaño del grafo, entonces hay m ejes, cada uno conectado a los dos nodos. Por tanto, al hacer la suma contamos dos veces cada eje \square

Corollary 1.12. *Todo grafo tiene un número par de nodos de grado impar*

Proof. Por el teorema anterior tenemos que

$$\sum_{v \in V} o(v) = 2m$$

y esta suma puede descomponerse como

$$\sum_{v \in V, o(v) \text{ impar}} o(v) + \sum_{v \in V, o(v) \text{ par}} o(v) = \sum_{v \in V, o(v) \text{ impar}} o(v) + 2k$$

o sea

$$\sum_{v \in V, o(v) \text{ impar}} o(v) = 2(m - k) = 2k'$$

Y tenemos una suma de números impares, cuyo resultado es par. Pero esto quiere decir que debe haber una cantidad par de sumandos \square

1.3.1 Secuencia gráfica

Una secuencia $S = (d_1, \dots, d_n)$ decreciente de enteros no negativos es una **secuencia gráfica** si existe un grafo simple G cuya secuencia de grados es S .

Theorem 1.13. Erdős y Gallai.

Sea $S = (d_1, \dots, d_n)$ una secuencia decreciente de enteros no negativos.

S es una secuencia gráfica si, y solo si, $\sum_{i=1}^n d_i$ es par y

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{i=k+1}^n \min\{k, d_i\}$$

para cada $1 \leq k \leq n-1$

Proof. [\implies] Tenemos en cuenta los k primeros nodos. Entre ellos hay, a lo sumo, $\binom{k}{2} = \frac{k(k-1)}{2}$ ejes, luego la suma de los grados de este subgrafo está acotada por el doble, $k(k-1)$.

Ahora bien, el nodo $k+i$ se une, como mucho, a k nodos entre los k primeros, o a d_{k+i} si $d_{k+i} < k$. Por tanto, se tiene la desigualdad:

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{i=k+1}^n \min\{k, d_i\}$$

[\impliedby] Necesitamos unas definiciones previas:

Subrealización: dado $S = (d_1, \dots, d_n)$ una subrealización es un grafo $G = (V, E^S)$, $V = \{1, \dots, n\}$ tal que $o(i) \leq d_i$, $\forall i$

Conjunto independiente: $X \subset V$ es independiente si $\forall v, w \in X$, $(v, w) \notin E$

Entorno: dado $v \in V$ un entorno es el conjunto $N(v) = \{u \in V : (v, u) \in E\}$

Índice crítico: dada una subrealización $G = (V, E^S)$, el índice crítico es el mayor índice h tal que $d_i = o(i)$, $\forall i < h$

Proof. Y se define el siguiente algoritmo: □

• **Paso inicial:**

$E^S = \emptyset$, $h = 1$, $G = (V, E^S)$ es subrealización de S

Si $S = (0, \dots, 0)$, FIN

• **Paso general:**

$G = (V, E^S)$ es la subrealización actual, h índice crítico.

$X = \{h + 1, \dots, n\}$ es independiente, porque en el paso inicial lo es (obvio) y en los siguientes no modificamos la independencia.

Se verifica

$$d_i - o(i) = 0 \quad \forall i < h$$

$$d_h - o(h) > 0$$

Aplicando cada uno de los siguientes pasos conseguiremos reducir $d_h - o(h)$ sin alterar $d_i - o(i)$, $i < h$. Cuando ninguno de los casos sea aplicable, veremos que $d_h = o(h)$.

Los casos son:

1. Supongamos que $\exists i \notin N(h) : o(i) < d_i$, entonces hacemos

$$E^S := E^S \cup \{(i, h)\}$$

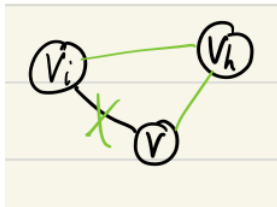
2. Supongamos que $\exists i < h : i \notin N(h)$. Entonces

$$o(i) = d_i \geq d_h > o(h) \implies \exists v \in N(i) : v \notin N(h)$$

y se pueden dar dos subcasos:

- (a) Si $d_h - o(h) \geq 2$, hacemos

$$E^S := (E^S \setminus \{(i, v)\}) \cup \{(i, h), (v, h)\}$$



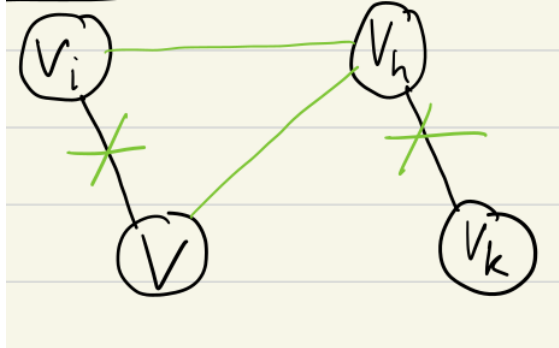
- (b) Si $d_h - o(h) = 1$, como $\sum_{i=1}^n d_i$ es par por hipótesis y $\sum_{i=1}^n o(i)$ es par por ser un subgrafo, entonces $\sum_{i=1}^n d_i - \sum_{i=1}^n o(i)$ es par, y se tiene que

$$\exists k > h : d_k > o(k)$$

y obtenemos dos nuevos subcasos:

- i. Si $k \notin N(h) \implies$ Caso 1
- ii. Si $k \in N(h)$, hacemos

$$E^S := E^S \setminus \{(i, v), (h, k)\} \cup \{(i, h), (v, h)\}$$



3. $1, \dots, h-1 \in N(h)$ y $\exists k > h$ tal que $o(k) \neq \min\{d_k, h\}$. Entonces $o(k) \leq d_k$, porque G^S es subrealización y $o(k) \leq h$ porque $X = \{h+1, \dots, n\}$ es independiente, como se ilustra:



Además, $o(k) < h$ pues $o(k) \neq \min\{d_k, h\}$ por hipótesis del caso 3.

Surgen dos subcasos:

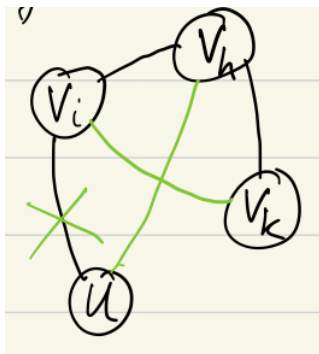
- (a) Si $k \notin N(h) \implies$ Paso 1
- (b) Si $k \in N(h)$, entonces $o(h) \geq h$ pues por hipótesis del caso 3, h está conectado con los $h-1$ nodos anteriores, y ahora con el k . O sea

$$o(h) \geq h > o(k)$$

Por tanto, $\exists i < h : i \notin N(k)$.

Además, $o(i) = d_i \geq d_h > o(h)$ por lo que $\exists u \in N(i) : u \notin N(h)$. Hacemos

$$E^S := E^S \setminus \{(i, u)\} \cup \{(i, k), (h, u)\}$$



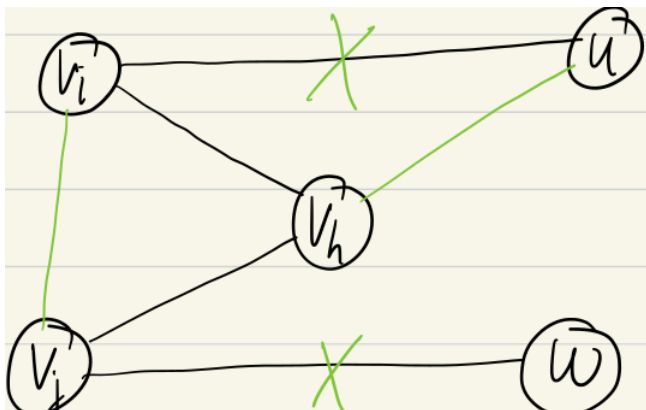
4. Supongamos que $1, \dots, h-1 \in N(h)$ y $\exists i, j < h : (i, j) \notin E^S$. Entonces

$$o(i) = d_i \geq d_h > o(h) \implies \exists u \in N(i) \setminus N(h)$$

$$o(j) = d_j \geq d_h > o(h) \implies \exists w \in N(j) \setminus N(h)$$

hacemos

$$E^S := E^S \setminus \{(i, u), (j, w)\} \cup \{(i, j), (u, h)\}$$



Supongamos ahora que no se da ninguno de los casos 1-4.

- Si no se da 2, entonces todos los nodos conectan al v_h
- Si no se da 4, entonces todos los nodos entre 1 y $h-1$ están conectados entre sí. O sea, el subgrafo inducido por $\{1, \dots, h\}$ es completo.
- Si no se da 3, $\forall k > h, o(k) = \min\{h, d_k\}$, entonces

$$\sum_{i=1}^h o(i) = h(h-1) + \sum_{i=h+1}^n o(i) \stackrel{*}{=} h(h-1) + \sum_{i=h+1}^n \min\{h, d_i\} \stackrel{**}{\geq} \sum_{i=1}^h d_i \implies o(h) = d_h$$

donde * se debe a que X es independiente, por lo que $\forall i > h, o(i)$ se corresponde con el número de conexiones con $\{1, \dots, h\}$, y ** es por hipótesis.

Así, obtenemos que d ya no es índice crítico, por lo que avanzaríamos el paso.

□

Theorem 1.14. Havel y Hakimi

Sea $S = (d_1, \dots, d_n)$ una secuencia decreciente de enteros no negativos con $n \geq 2$ y $d_i \geq 1$. Entonces, S es una secuencia gráfica si, y solo si, la secuencia

$$S' = (d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n)$$

es gráfica

Proof. [\Leftarrow] S' gráfica $\implies \exists G' = (V', E')$, $V' = \{2, \dots, n\}$ cuya secuencia de grados es S' . Añadimos un nuevo nodo llamado 1 y lo conectamos con $\{2, \dots, d_1 + 1\}$, obteniendo un grafo G con secuencia de grados S , luego S es gráfica.

[\implies] Supongamos que S es gráfica. Sea G tal que S es secuencia de grados de G y $\sum_{j \in N(1)} d_j$ es máxima.

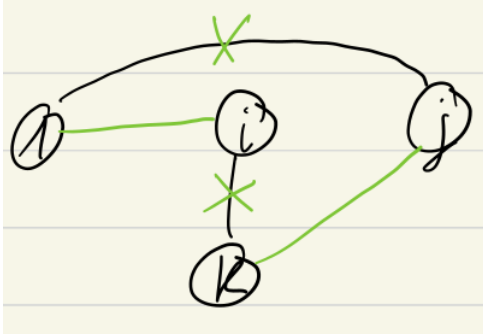
Afirmamos que 1 es adyacente a nodos de grados d_2, \dots, d_{d_1+1} .

Supongamos que no es así. Entonces, como $o(1) = d_1$, existirían $i < j$ tales que

$$j \in N(1), i \notin N(1), o(i) = d_i > d_j = o(j)$$

Como $o(i) > o(j) \implies \exists k \in N(i) \setminus N(j)$.

Consideremos $\bar{G} = (V, \bar{E})$ donde $\bar{E} = E \setminus \{(1, j), (i, k)\} \cup \{(1, i), (k, j)\}$:



\bar{G} tiene secuencia de grados S y $\sum_{j \in \bar{N}(1)} d_j > \sum_{j \in N(1)} d_j \#$ lo que contradice la maximalidad de G .

Por tanto, en G , 1 es adyacente a nodos de grados $\{d_2, \dots, d_{d_1+1}\}$.

Haciendo $G' = G \setminus \{1\}$ obtenemos un grafo de orden $n - 1$ y secuencia S' □

Algoritmo de Havel-Hakimi

Dada una secuencia decreciente de enteros no negativos $S = (d_1, \dots, d_n)$ que queremos saber si es secuencia gráfica:

1) Si $\sum d_i$ es impar \implies FIN, no es secuencia gráfica

Si $\max(d_i) = 0 \implies$ FIN, es secuencia gráfica

Si no, ir a 2)

2) Si $S = (d_1, \dots, d_n)$, hacer $S = (d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n)$ y volver a 1)

1.4 Caminos y ciclos

Definition 1.15. Dado un grafo $G = (V, E)$, un **paseo** que une v_0 y v_k de V es una secuencia de la forma

$$v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k$$

donde cada $e_i \in E$ es un eje incidente a v_{i-1} y v_i .

La **longitud del paseo** es el número de ejes, k

Si $v_0 = v_k$, el paseo se dice que es **cerrado**

Si todos los ejes son distintos el paseo se dice **simple**

Un paseo simple en el que todos los nodos intermedios son distintos se denomina **camino**

Un camino cerrado se denomina **ciclo**

Proposition 1.16. *Todo paseo entre dos vértices contiene un camino entre dichos vértices*

Proof. Sea el paseo $v_1 \dots v_k$.

Si es un camino, fin.

Si no es un camino, entonces $\exists w \in V$ tal que el camino puede escribirse como

$$v_1 \dots w \dots w \dots v_k$$

Llamamos $\gamma = w \dots w$ el paseo interior que une w consigo mismo y lo eliminamos del paseo original, quedando

$$v_1 \dots w \dots v_k$$

Reiterando esto hasta que no quede ningún w de esta forma, lo tenemos.

Y esto sucede en algún momento, pues el paseo es finito. \square

Proposition 1.17. *Cualquier paseo cerrado de longitud impar contiene un ciclo*

Proof. Sea un paseo cerrado $v = v_1 \dots w \dots w \dots v_k = v$, donde w está ahí así porque el paseo cerrado no es un camino cerrado, si lo fuera sería un ciclo.

Si no es un ciclo, podemos reordenar el paseo, expresándolo como

$$w \dots v_1 \dots w u \dots v_k \dots w$$

Hacemos ahora $a = w \dots v_1 \dots w$, $b = u \dots v_k \dots w$.

Como el paseo es de longitud impar, entonces a o b es de longitud impar. Tomamos el que sea de longitud impar. Si es un ciclo, fin. Si no es un ciclo, repetimos el proceso.

Al final se llega a un ciclo (el paseo es finito), como mínimo, de longitud 3 (pues ha de ser impar y un ciclo tiene longitud al menos 2) \square

1.5 Representaciones matriciales de grafos

1.5.1 Matriz de adyacencia

Sea $G = (V, E)$ un grafo no dirigido con $V = \{1, \dots, n\}$. La **matriz de adyacencia** $A = (a_{ij})$ es la matriz cuadrada de orden n definida como

$$a_{ij} = \begin{cases} 1 & \text{si } (i, j) \in E \\ 0 & \text{en otro caso} \end{cases}$$

Theorem 1.18. *Sea A la matriz de adyacencia de un grafo no dirigido G .*

Para cada $k \geq 1$ entero, la matriz A^k representa la cantidad de paseos de longitud k que hay entre cada par de vértices

Proof. Por inducción:

Para $k = 1$, se verifica por definición de matriz de adyacencia.

Supongamos que se verifica para $k - 1$ y la hipótesis de inducción.

Entonces

$$A^k = A \cdot A^{k-1}$$

es decir

$$(A^k)_{ij} = \sum_{h=1}^n a_{ih} a_{hj}^{[k-1]}$$

Donde a_{ih} indica la cantidad de paseos de i a h de longitud 1 (caso $k = 1$) y $a_{hj}^{[k-1]}$ la cantidad de paseos de h a j de tamaño $k - 1$ (por la hipótesis de inducción). Por tanto, su multiplicación representa la cantidad de paseos de i a j que pasan por h . Y la suma en h de estas cantidades nos da la cantidad total de paseos desde i hasta j , lo que prueba el enunciado. \square

2 Conectividad en grafos

2.1 Conexión. Componentes conexas

Definition 2.1. Dado un grafo $G = (V, E)$, dos vértices $u, v \in V$ se dice que están **conectados** si existe un camino uniendo los vértices u y v .

Un grafo es **conexo** si cada par de vértices están conectados entre sí, en caso contrario se dice **disconexo**. El grafo trivial se considera conexo.

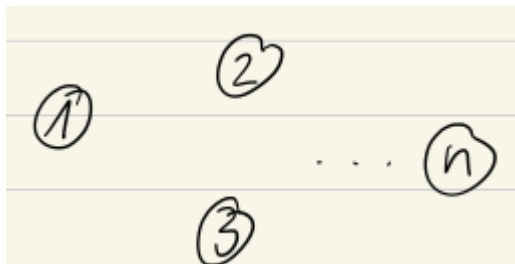
Definition 2.2. Dado un grafo $G = (V, E)$ y $V' \subset V$, el subgrafo inducido $G_{V'}$ es una **componente conexas** de G si:

- $G_{V'}$ es conexo
- V' es maximal verificando la condición anterior

Obsérvese que un grafo es conexo si solo tiene una componente conexas.

Proposition 2.3. Sea G un grafo de orden n y tamaño m . Si G es un grafo conexo, entonces $m \geq n - 1$

Proof. Un grafo conexo tiene una componente conexas. Comenzamos con el grafo con los nodos y sin ejes



Cada eje que añadimos, a lo sumo elimina una componente conexas, cuando al añadirlo une dos componentes conexas distintas.

Como partimos de n componentes conexas, necesitamos, como mínimo, añadir $n - 1$ ejes □

Definition 2.4. Sea $G = (V, E)$ un grafo conexo de orden n . Sean $u, v \in V$.

- Un camino entre u y v de longitud mínima se dice que es una **geodésica** entre u y v
- Se define la **distancia** entre u y v , $d(u, v)$ como la longitud de cualquier geodésica entre u y v
- Se define el **diámetro** de G , $diam(G)$, como el máximo de las distancias entre todos los pares de vértices

Theorem 2.5. Sea G un grafo de orden al menos 3. G es conexo si, y solo si, contiene dos vértices u y v tales que $G - u$ y $G - v$ son conexas

Nota: $G - u$ es el grafo inducido por $V \setminus \{u\}$.

Proof. [\Leftarrow] Demostraremos que para cada par de vértices $i, j \in V$, existe un paseo entre ellos.

- Si $\{i, j\} \neq \{u, v\}$, podemos suponer que ambos son distintos de u . Como $G - u$ es conexo, entonces i, j están conectados en $G - u \implies$ están conectados en G
- Si $\{i, j\} = \{u, v\}$, como $n \geq 3 \implies \exists w \neq u, v$. Como $G - u$ es conexo, existe un camino P_1 entre w y v en $G - u$. Análogamente, existe un camino P_2 entre w y u en $G - v$. Concatenando $-P_1$ con P_2 obtenemos un camino entre v y u

[\implies] Supongamos que $G = (V, E)$ es conexo. Sean $u, v \in V$ tales que $d(u, v) = \text{diam}(G)$, ¿ $G - u$, $G - v$ conexos?

Lo vemos para $G - u$:

Dados $i, j \neq u$, como G es conexo, tenemos que i, v están conectados en G y v, j también.

Consideremos P_1 la geodésica entre i, v , entonces $u \notin P_1$, ya que, si fuera $u \in P_1$, entonces sería $d(i, v) > d(u, v) = \text{diam}(G) \#$

De igual forma, consideramos P_2 la geodésica entre v, j y se tiene $u \notin P_2$ por la misma razón que antes.

Así, $P_1 P_2$ es un paseo de i a j en $G - u$, y este es conexo.

De igual forma para $G - v$ □

2.2 Obtención de las componentes conexas

2.2.1 Mediante las potencias de la matriz de adyacencia

Proposition 2.6. Sea A la matriz de adyacencia de un grafo G . Consideremos

$$\bar{A} = \sum_{k=1}^{n-1} A^k$$

Los nodos i, j están en la misma componente conexa si, y solo si, $\bar{a}_{ij} > 0$

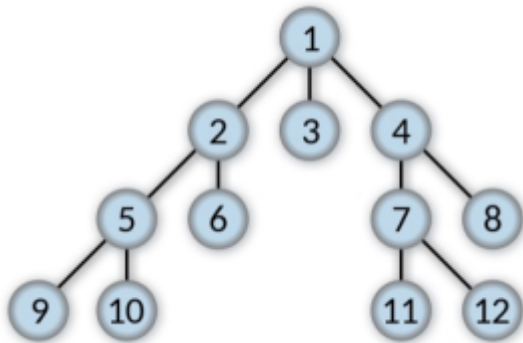
Proof. Nótese que $d(i, j) \leq n - 1$

[\implies] Si i, j están en la misma componente conexa, existe un paseo que los conecta. Si el paseo tiene longitud $k \leq n - 1$, se tendrá $a_{ij}^{[k]} \geq 1 \implies \bar{a}_{ij} > 0$

[\Leftarrow] Si $\bar{a}_{ij} > 0$, entonces para algún $k \leq n - 1$, se tiene $a_{ij}^{[k]} > 0$, luego i, j están conectados, y deben estar en la misma componente conexa □

2.2.2 Breadth-first search - Búsqueda primero en anchura

Se recorre el grafo comenzando en un nodo, y recorriendo todos sus vecinos. Luego se recorren los vecinos de sus vecinos, etc



Algoritmo BFS para computar el número de componentes conexas

Sea $G = (V, E)$ un grafo simple

Paso inicial

- $X = \emptyset$ es el conjunto de nodos visitados
- $h = 1$, el índice de las componentes conexas

Paso 1

- Tomamos $v \in V \setminus X$
- Hacemos $C(h) = \{v\}$, $r_h = v$
- Tomamos $i(v) = 0$, $i = 0$

Paso 2

- Tomamos $u \in C(h) \setminus X$ con $i(u) = i$. Si no existe tal nodo, ir al Paso 4

Paso 3

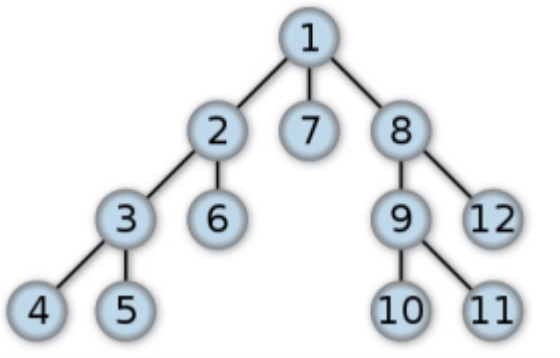
- Para todo $w \in N(u) \setminus C(h)$, hacemos $i(w) = i(u) + 1$, $C(h) = C(h) \cup \{w\}$, $X = X \cup \{u\}$. Volver al Paso 2

Paso 4

- Si $C(h) \subset X$, hemos completado la componente $C(h)$. Para cada $u \in C(h)$, $i(u) = d(r_h, u)$
 - Si $X = V$, FIN. Se han encontrado todas las componentes conexas.
 - Si no, hacer $h = h + 1$ e ir al Paso 1
- Si no, hacer $i = i + 1$ e ir al Paso 2

2.2.3 Depth-first search - Búsqueda primero en profundidad

Se recorre un grafo eligiendo un nodo, y se exploran sucesivamente nodos adyacentes al nodo anterior que no hayan sido previamente explorados. Cuando un nodo no tiene nodos adyacentes que no hayan sido explorados se vuelve a su predecesor y se continúa el procedimiento explorando otro nodo adyacente a este que no haya sido previamente explorado.



Algoritmo DFS para computar el número de componentes conexas

Sea $G = (V, E)$ un grafo simple

Paso Inicial

- $X = \emptyset$ es el conjunto de nodos visitados
- $h = 1$, el índice de las componentes conexas

Paso 1

- Tomamos $v \in V \setminus X$
- Hacemos $C(h) = \{v\}$, $u = v$, $p(u) = v$

Paso 2

- Si $N(u) \setminus \{p(u)\} \subset X$, hacemos $X = X \cup \{u\}$ (el nodo u ha sido completamente explorado). Ir al Paso 3
- Si no, sea $w \in N(u) \setminus X$, $w \neq p(u)$. Hacemos $C(h) = C(h) \cup \{w\}$. Hacemos $p(w) = u$. Volver al Paso 2

Paso 3

- Si $u = v$, se ha completado la componente conexas $C(h)$
 - Si $X = V$, FIN. Se han encontrado todas las componentes conexas
 - Si no, hacer $h = h + 1$ e ir al Paso 1
- Si $u \neq v$, hacer $u = p(u)$ y volver al paso 2

2.3 Conjuntos separadores

Dado un grafo $G = (V, E)$ y sean $V' \subset V$, $E' \subset E$. Denotamos por

- $G - V'$ al **subgrafo de G inducido por $V \setminus V'$** , es decir, el resultante de eliminar todos los nodos de V' y los ejes de E incidentes a algún nodo de V'
- $G - E'$ al **subgrafo generador de G resultante de eliminar los ejes de E'**

2.3.1 Conjuntos separadores de vértices. Vértices de corte

Definition 2.7. Sea $G = (V, E)$ un grafo conexo.

- Un conjunto de vértices $V' \subset V$ se dice que es un **conjunto separador de vértices** de G si $G - V'$ es desconexo.
Si $|V'| = k$ se dice que es un conjunto k -separador
- Un vértice $v \in V$ es un **vértice de corte** de G si $\{v\}$ es un conjunto separador de vértices de G

Theorem 2.8. Sea G un grafo conexo de orden mayor o igual que 3.

Un vértice u es un vértice de corte de G si, y solo si, existen dos vértices v, w distintos de u tales que cualquier camino de v a w pasa por u

Proof. [\Leftarrow] Si eliminamos u rompemos todos los caminos de v a w en $G - u \Rightarrow G - u$ desconexo $\Rightarrow u$ vértice de corte

[\Rightarrow] $G - u$ desconexo $\Rightarrow \exists i, j$ no conectados en $G - u$.

Como i, j están conectados en G (conexo), si existiera un camino P entre i y j que no pasa por u , al quitar u ese camino no se vería alterado, por lo que i, j seguirían conectados en $G - u$.

Esto quiere decir que para todo camino P entre i y j , P debe pasar por u □

Theorem 2.9. Un grafo conexo con al menos dos vértices contiene al menos dos vértices que no son vértices de corte

Proof. Es un corolario del teorema 2.5

G conexo \Rightarrow hay dos nodos u, v tales que $G - u$ y $G - v$ son conexos $\Rightarrow u, v$ no son vértices de corte □

2.3.2 Conjuntos separadores de ejes. Puentes

Definition 2.10. Sea $G = (V, E)$ un grafo no dirigido, y sea $\emptyset \neq V' \subsetneq V$.

Denotamos por $[V', V \setminus V']$ al conjunto de ejes de E que son incidentes a un vértice de V' y otro de $V \setminus V'$. Estos conjuntos de ejes se denominan **corte**.

Un corte de cardinal k se denomina **k -corte**

Un eje e se dice que es un **eje de corte** si $\{e\}$ es un corte

Definition 2.11. Sea $G = (V, E)$ un grafo conexo.

Un conjunto de ejes $E' \subset E$ se dice que es un **conjunto separador de ejes** de G si $G - E'$ es desconexo

Un eje e se dice que es un **puente** si $\{e\}$ es un conjunto separador de ejes

Proposition 2.12. Todo corte es un conjunto separador de ejes

Proof. Tomamos $E' = [S, V \setminus S]$. En $G - E'$, S y $V \setminus S$ no están conectados. □

Nótese que el recíproco no es cierto, ya que un conjunto separador de ejes puede tener más ejes de los necesarios.

Proposition 2.13. Todo conjunto separador de ejes contiene un corte

Proof. Si E' es un conjunto separador de ejes, entonces $G - E'$ es desconexo.

Sea V_1 una componente conexa de $G - E'$, entonces $[V_1, V \setminus V_1]$ es un corte de G contenido en E' □

Como consecuencia de estos dos resultados, los conceptos de puente y eje de corte son equivalentes. Además, obsérvese que si e es un puente, entonces $G - e$ tiene exactamente dos componentes conexas. Sin embargo, un corte puede dar lugar a más componentes conexas.

Definition 2.14. Un corte X de un grafo conexo G es **minimal** si no existe $Y \subset X$ tal que Y es un corte de G .

Un **corte mínimo** de un grafo G es un corte cuyo cardinal es mínimo entre todos los cortes de G

Proposition 2.15. Si X es un corte minimal de un grafo conexo G , entonces $G - X$ tiene exactamente dos componentes conexas

Proof. Si tiene más de dos componentes conexas, entonces existe un eje $e = (i, j)$ con $i \in S$, $j \in V \setminus S$ de forma que al añadir ese eje, eliminamos una componente conexas, pero el grafo seguiría siendo desconexo. Por tanto, el corte $[S \cup \{j\}, V \setminus (S \cup \{j\})]$ está contenido estrictamente en el corte inicial, y este no es minimal. \square

Vamos ahora a ver una serie de resultados que caracterizan los puentes de un grafo conexo.

Theorem 2.16. Sea G un grafo conexo. Un eje e es un puente de G si, y solo si, existen vértices u y v tales que e está contenido en cualquier camino de u a v en G

Proof. $[\implies]$ Supongamos que $e = (u, v)$ es un puente. Supongamos que existe un camino P conectando u, v tal que $e \notin P$. En tal caso, al quitar e , u y v siguen conectados por P , por lo que $G - e$ sigue siendo conexo#

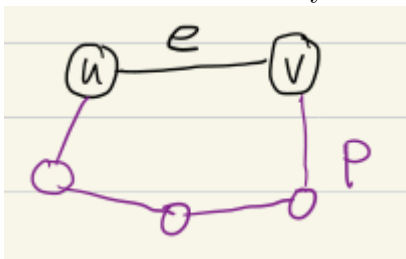
$[\impliedby]$ De ser así, al eliminar e se rompen todos los caminos entre u y v . En $G - e$, u, v no están conectados, por lo que $G - e$ es desconexo y e es un puente \square

Theorem 2.17. Un eje e es un puente de un grafo conexo G si, y solo si, no pertenece a ningún ciclo de G

Proof. $[\implies]$ Si e es un puente, entonces es un eje de corte

Si e estuviera en un ciclo, habría, al menos, otro eje en el corte#

$[\impliedby]$ Supongamos que $e = (u, v)$ no es un puente. Entonces $G - e$ sería conexo. Por tanto, existe un camino P conectando u y v con $e \notin P$, pero entonces $P \cup \{e\}$ es un ciclo.



\square

Corollary 2.18. Si G es un grafo simple conexo y sin ciclos, entonces cada eje es un puente de G

2.4 Conectividad. k -conexión

Definition 2.19. Sea G un grafo conexo no completo. La **conectividad de G** , $\kappa(G)$ es el menor índice k para el que existe un conjunto k -separador de vértices

Por convenio, si G es desconexo o trivial, $\kappa(G) = 0$.

Si G es un grafo completo, $\kappa(G) = n - 1$.

Definition 2.20. Un grafo se dice **k -conexo** si $\kappa(G) \geq k$

Definition 2.21. Sea G un grafo conexo. La **conectividad por ejes de G** , $\lambda(G)$ es el menor índice k para el que existe un conjunto k -separador de ejes

Por convenio, si G es desconexo o trivial, $\lambda(G) = 0$.

Si G es conexo y tiene un puente, $\lambda(G) = 1$.

Definition 2.22. Un grafo se dice **k -conexo por ejes** si $\lambda(G) \geq k$

Proposition 2.23. Dado $n \geq 1$, se verifica $\lambda(K_n) = n - 1$

Proof. Para verlo:

- $\lambda(K_n) \leq n - 1$: sea E' el conjunto de ejes incidentes a un nodo $v \in V$. Como $o(v) = n - 1$, $\forall v \in V$, entonces $|E'| = n - 1$ y $G - E'$ es desconexo.
- $\lambda(K_n) \geq n - 1$: sea E' un corte mínimo, $E' = [V', V \setminus V']$, supongamos que $|V'| = q$, $1 \leq q \leq n - 1$

Entonces, como estamos en K_n , todos los nodos de V' conectan con todos los de $V \setminus V'$, luego $|E'| = q \cdot (n - q)$

$$- q \geq 1 \implies (q - 1) \geq 0$$

$$- q \leq n - 1 \implies (n - q - 1) \geq 0$$

Por tanto

$$(q - 1)(n - q - 1) \geq 0$$

por lo que es

$$q(n - q - 1) - (n - q - 1) \geq 0 \implies q(n - q) - q - (n - 1) + q \geq 0 \implies$$

$$\implies q(n - q) - (n - 1) \geq 0 \implies q(n - q) \geq n - 1 \implies |E'| \geq n - 1$$

□

Theorem 2.24. Sea G un grafo conexo. Entonces

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

Proof. Vamos a ver primero que $\lambda(G) \leq \delta(G)$:

Sea v tal que $o(v) = \delta(G)$ y sea E' el conjunto de los vértices incidentes a $v \implies |E'| = \delta(G)$, y E' es un conjunto separador de ejes, luego

$$\lambda(G) \leq |E'| = \delta(G)$$

Y ahora vemos que $\kappa(G) \leq \lambda(G)$

- Si G es desconexo o trivial, $\kappa(G) = \lambda(G) = 0$
- Si G es completo $\kappa(G) = \lambda(G) = n - 1$
- Supongamos que G es conexo no completo, con $n \geq 3$. Sea X un corte mínimo de G :

$$|X| = \lambda(G) \leq \delta(G) \stackrel{*}{\leq} n - 2$$

Por la proposición 2.15, como X es mínimo, $G - X$ tiene dos componentes conexas V_1, V_2 que podemos suponer que verifican

$$|V_1| = q, |V_2| = n - q, 1 \leq q \leq n - 1$$

Y vemos dos posibles casos:

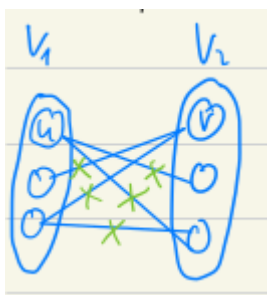
- Si en G todo vértice de V_1 es adyacente a todo vértice de V_2 , entonces

$$|X| = q(n - q) \geq n - 1 \#$$

lo que es una contradicción, por $*$.

- Existen $u \in V_1, v \in V_2$ no adyacentes en G . Construimos V' del siguiente modo: para cada $e \in X$, si u es incidente a e , añadimos a V' el otro extremo de e . Si v es incidente a e , añadimos a V' el otro extremo de e . Si e no es incidente ni a u ni a v , añadimos a V' uno de los extremos de e . De esta forma

$$|V'| \leq |X|$$



- * Y en $G - V'$, u y v no están conectados, porque
 - No son adyacentes
 - Los ejes que llevan u a V_2 están rotos
 - Los ejes que llevan v a V_1 están rotos
 Por tanto, V' es un conjunto separador, por lo que

$$\kappa(G) \leq |V'| \leq |X| = \lambda(G)$$

□

2.4.1 Teorema de Menger

Definition 2.25. Sea $G = (V, E)$ un grafo conexo no dirigido. Un conjunto de vértices S **separa dos vértices** u, v si $G - S$ es desconexo y u, v pertenecen a distintas componentes conexas.

Definition 2.26. Una colección de caminos $\{P_1, \dots, P_h\}$ conectando u y v se dice que son **internamente disjuntos** si cada par de estos caminos no tienen otros vértices en común más que u y v

Theorem 2.27. De Menger

Sean u, v dos vértices no adyacentes en un grafo G . El cardinal de un conjunto de menor tamaño que separa u y v coincide con el máximo número de caminos entre u y v internamente disjuntos en G

Proposition 2.28. (Ejercicio 2.11)

Sea $G = (V, E)$ un grafo k -conexo y $e \in E$ un eje de G . Entonces $G - e$ es $(k - 1)$ -conexo

Proof. Si $\kappa(G - e) < k - 1 \implies \exists S \subset V, |S| < k - 1$ tal que $(G - e) - S$ es desconexo. Pero entonces, suponiendo $e = (u, v)$

$$S \cup \{u\} \text{ o } S \cup \{v\}$$

es conjunto separador de G , con $|S \cup \{u\}| < k$, contradicción, pues G es k -conexo. □

Theorem 2.29. Un grafo $G = (V, E)$ es k -conexo si, y solo si, para cada par de vértices $u \neq v \in V$ hay en G al menos k caminos internamente disjuntos entre u y v

Proof. [\implies] □

- Si $G = K_n$, se tiene $\kappa(G) = n - 1$, y si $i \sim j$, entonces tenemos los caminos (i, j) y $\forall k \neq i, j$ el (i, k, j) , en total hay $n - 1$ caminos internamente disjuntos
- Sea G k -conexo no completo. Sean $u \neq v \in V$. Entonces

- Si no son adyacentes, sea U el conjunto de menor tamaño que separa u y v . Como $G - U$ es desconexo, se tiene

$$|U| \geq \kappa(G) \geq k$$

y por el teorema de Menger hay al menos k caminos internamente disjuntos entre u y v en G

- Si $(u, v) \in E$, como G es k -conexo, entonces $G - e$ es $(k - 1)$ -conexo, y entonces hay al menos $k - 1$ internamente disjuntos entre u y v en $G - e$ (y en G), aplicando el caso anterior. Añadiendo el camino e en G tenemos, al menos, k caminos internamente disjuntos entre u y v en G

[\Leftarrow] Supongamos que $\forall u, v \in V$, (*) $\exists k$ caminos internamente disjuntos entre u y v en G

Sea S un conjunto separador de vértices de menor tamaño, $|S| = \kappa(G)$.

Como $G - S$ es desconexo, sean u, v dos vértices en distintas componentes conexas de $G - S$. Entonces S separa u y v .

Por (*) y el teorema de Menger, debe ser

$$\kappa(G) = |S| \geq k$$

luego G es k -conexo.

2.4.2 Resultados para conectividad por ejes

Un conjunto $X \subset E$ **separa dos vértices** u, v si $G - X$ es desconexo y u, v están en distintas componentes conexas de $G - X$.

Theorem 2.30. Sean u, v dos vértices de un grafo conexo G . El mínimo número de ejes que separan u, v es igual al máximo número de caminos en G , sin ejes comunes, uniendo u, v

Theorem 2.31. Un grafo $G = (V, E)$ es k -conexo por ejes si, y solo si, para cada par de vértices $u \neq v \in V$, hay en G al menos k caminos sin ejes comunes entre u y v , disjuntos entre u y v

3 Árboles

Definition 3.1. Un grafo $G = (V, E)$ es un **árbol** si es conexo y no contiene ciclos
 Un **árbol generador** de un grafo es un subgrafo parcial conexo y sin ciclos
 Un **bosque** es un grafo sin ciclos
 En un árbol, los nodos con grado de incidencia 1 se denominan **hojas**

Theorem 3.2. Caracterización de árboles

Sea $G = (V, E)$. Son equivalentes:

1. G es conexo y sin ciclos
2. Entre cada par de vértices distintos de V existe un único camino
3. G es conexo y $m = n - 1$
4. G no contiene ciclos y $m = n - 1$
5. G está minimalmente conectado
6. G no contiene ciclos y si añadimos una arista entre dos vértices no adyacentes cualesquiera de V , el grafo resultante contiene un único ciclo

Proof. Veamos todas las implicaciones:

[1 \implies 2] Como G es conexo, entre cada par de vértices distintos de V , existe un camino que los une. ¿Es único?

Supongamos que no lo es, entonces tenemos los caminos

$$v_1 \dots v_i v_{i+1} \dots v_{j-1} v_j \dots v_k$$

$$v_1 \dots v_i v_{i+1}^* \dots v_{j-1}^* v_j \dots v_k$$

luego hay un ciclo entre v_i y v_j . El camino debe ser único.

[2 \implies 3] Es conexo porque todos los vértices están conectados entre sí.

Para ver que $m = n - 1$, hacemos inducción sobre n :

- $n = 1$: obvio ✓
- $n = 2$: el grafo solo puede ser K_2 , luego $n = 2, m = 1$ ✓

Supongamos que se verifica para los grafos de orden $< k$ y la hipótesis de inducción.

Entonces, si tenemos un grafo de orden k verificando (2), quitamos un eje, y quedan dos componentes conexas de órdenes n_1, n_2 con $n_1 + n_2 = n$. Por inducción, se tiene que $m_1 = n_1 - 1$ y $m_2 = n_2 - 1$ luego el grafo inicial tiene tamaño

$$m = m_1 + m_2 + 1 = n_1 - 1 + n_2 - 1 + 1 = n - 1 \checkmark$$

[3 \implies 4] Solo hay que ver que G no tiene ciclos.

Supongamos que tiene algún ciclo. Como es conexo y tiene un ciclo, al quitar un eje del ciclo el grafo sigue siendo conexo y el grafo resultante tiene $n - 2$ ejes. Pero vimos que un grafo conexo tiene al menos $n - 1$ ejes.

[4 \implies 5] Si vemos que es conexo, ya está, pues $m = n - 1$.

Supongamos que no es conexo, y sean G_1, \dots, G_q las componentes conexas de G . Ahora bien, cada G_i es conexo y sin ciclos, luego usamos $1 \implies 3$ y obtenemos que $m_i = n_i - 1$, y entonces

$$n - 1 = m = \sum_i (n_i - 1) = \sum_i n_i - q = n - q$$

y debe ser $q = 1$ y solo hay una componente conexa. **Contradicción**, pues supusimos que G era desconexo.

Por tanto, G ha de ser conexo.

[5 \implies 6] Si tuviera algún ciclo, podríamos quitar un eje del ciclo y seguiría siendo conexo, pero esto contradice la hipótesis de que G es minimalmente conexo. Por tanto, no contiene ciclos.

Si al añadir un eje se forma más de un ciclo, es que antes había algún ciclo, y esto no es así. Por tanto, se forma un único ciclo.

[6 \implies 1] No contiene ciclos por hipótesis. Y es conexo porque de no serlo, podríamos añadir un eje sin formar ningún ciclo, sería un eje que conectase dos componentes conexas distintas. \square

Definition 3.3. En un árbol podemos distinguir un nodo cualquiera que pasará a llamarse **nodo raíz** r

El par compuesto por árbol y raíz se denomina **árbol con raíz**

El resto de nodos con grado de incidencia mayor que 1 se denominan **nodos**

Los nodos con grado de incidencia 1 (a veces se excluye la raíz) se llamarán **hojas**

Cada nodo $v \neq r$ del árbol enraizado estará conectado a r mediante un único camino $rv_1 \dots v_k v$.

Diremos que los nodos r, v_1, \dots, v_k son los **predecesores** de v en el árbol y que v es el **sucesor** de estos nodos.

La relación de precedencia

$$u \leq v \iff u \text{ precede a } v \text{ o bien } u = v$$

induce un orden parcial en los nodos del árbol enraizado.

Si $u \leq v$ o $v \leq u$ diremos que u y v son **comparables**.

Definition 3.4. Diremos que la raíz del árbol enraizado está a **nivel cero**.

El **nivel** del resto de nodos será la longitud del camino que los une a la raíz.

Llamamos **altura** del árbol enraizado al máximo de los niveles de sus nodos

Proposition 3.5. *Todo árbol con $n \geq 2$ tiene al menos dos hojas*

Proof. Como es un árbol, se tiene $m = n - 1$.

Se tiene entonces

$$\sum_{v \in V} o(v) = 2m = 2n - 2$$

y esto puede expresarse como

$$\sum_{v \in V} o(v) = \sum_{v \text{ hoja}} o(v) + \sum_{v | o(v) \geq 2} o(v)$$

y si suponemos que hay h hojas, entonces es

$$\sum_{v \text{ hoja}} o(v) = \sum_{v \text{ hoja}} 1 = h$$

Y es

$$2n - 2 = h + \sum_{v|o(v) \geq 2} o(v) \geq h + 2(n - h) = h + 2n - 2h = 2n - h$$

luego $h \geq 2$. □

Definition 3.6. Un **árbol binario** es aquel cuyos nodos tienen grado 1 ó 3 excepto un único nodo que tiene grado 2.
Llamamos **raíz del árbol binario** al nodo de grado 2.

Proposition 3.7. El número de vértices de un árbol binario es impar

Proof. $m = n - 1$ y

$$\sum_{v \in V} o(v) = 2n - 2 \text{ (par)}$$

y

$$\sum_{v \in V} o(v) = o(r) + \sum_{v|o(v) \text{ impar}} o(v) = 2 + \sum_{v|o(v) \text{ impar}} o(v)$$

luego la cantidad de nodos de grado impar ha de ser par, para que esta suma de par, por tanto hay una cantidad total par de nodos □

Proposition 3.8. El número de hojas de un árbol binario es $\frac{n+1}{2}$

Proof. Sea h la cantidad de hojas, entonces

$$o(r) + h + 3 \cdot (n - h - 1) = 2n - 2 \implies 2 + 3n - 2h - 3 = 2n - 2 \implies n + 1 = 2h \implies h = \frac{n + 1}{2}$$

□

Proposition 3.9. El número de nudos de un árbol binario es $\frac{n-3}{2}$

Proof. Si lo llamamos k , debe verificarse

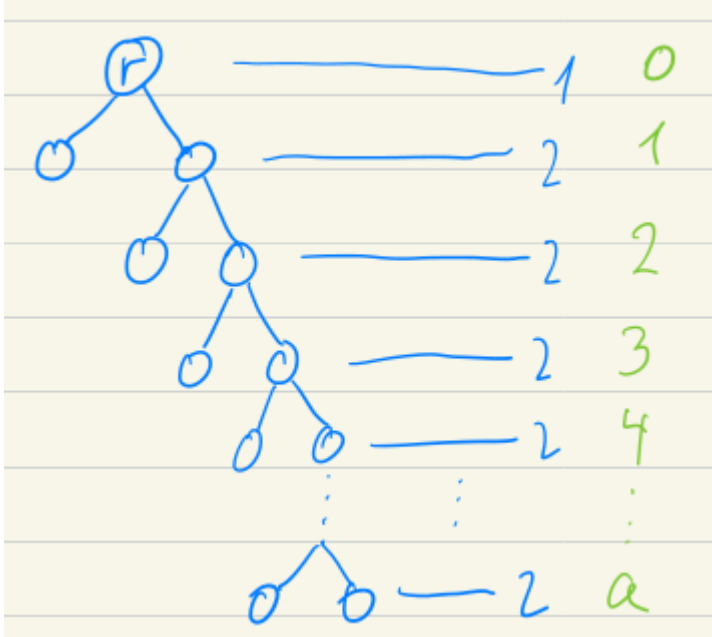
$$1 + \frac{n+1}{2} + k = n \implies 2 + n + 1 + 2k = 2n \implies 2k = n - 3 \implies k = \frac{n-3}{2}$$

□

Proposition 3.10. La altura de un árbol binario está comprendida entre $\lceil \log_2(n+1) - 1 \rceil$ y $\frac{n-1}{2}$

Proof. Primero veamos la altura máxima:

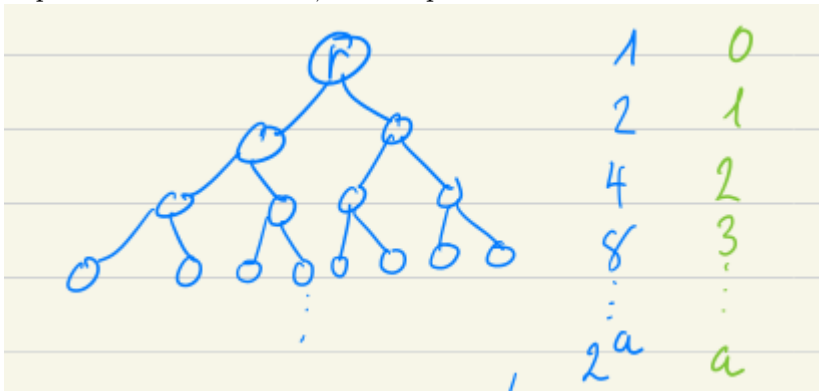
El árbol de altura máxima es de la siguiente forma:



y se tiene

$$n = 2a + 1 \implies a = \frac{n-1}{2}$$

Y para la altura mínima, vemos que el árbol de altura mínima tiene la forma



y tenemos

$$1 + 2 + 2^2 + \dots + 2^{a-1} < n \leq 1 + 2 + 2^2 + \dots + 2^a$$

esto son sumas geométricas de razón 2, y queda

$$2^a - 1 < n \leq 2^{a+1} - 1 \implies 2^a < n + 1 \leq 2^{a+1}$$

por lo que

$$a < \log_2(n + 1) \leq a + 1$$

por lo tanto

$$a = \lceil \log_2(n + 1) - 1 \rceil$$

□

Definition 3.11. Una red es una terna (V, E, l) formada por un grafo (V, E) y una función $l : E \rightarrow \mathbb{R}$ llamada **función de peso (longitud)**.

Dada una red (V, E, l) y un árbol generador $T = (V, E_T)$, se define el **peso del árbol** como

$$l(T) = \sum_{e \in E_T} l(e)$$

La **búsqueda del árbol generador de peso mínimo** en G es la resolución del problema

$$\begin{aligned} \min & \quad l(T) \\ \text{s.a.} & \quad T \text{ es árbol generador de } G \end{aligned}$$

Theorem 3.12. Sea una red $G = (V, E, l)$ y un árbol generador de G , $T = (V, E_T)$. Las siguientes afirmaciones son equivalentes:

1. T es un árbol generador minimal
2. Dada cualquier arista $a \in E \setminus E_T$ y cualquier otra arista $e \in E_T$ perteneciente al único ciclo del grafo $(V, E_T \cup \{a\})$ se verifica $l(e) \leq l(a)$
3. Para cualquier arista $e \in E_T, l(e) \leq l(a), \forall a \in w(V_1^e)$, este último es el cociclo en G asociado a una componente conexa cualquiera de $(V, E_T \setminus \{e\})$ (un cociclo es un corte, o sea $w(V_1^e) = [V_1^e, V \setminus V_1^e]$)

Proof. Veamos cada implicación:

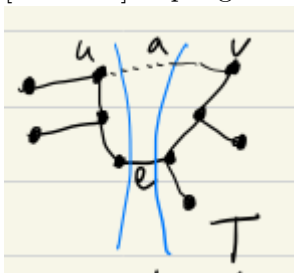
[1 \implies 2] Supongamos que $l(a) < l(e)$. Si hacemos

$$E'_T = E_T \setminus \{e\} \cup \{a\}$$

obtenemos un nuevo árbol generador $T' = (V, E'_T)$ y

$$l(T') < l(T) \#$$

[2 \implies 3] Supongamos que $\exists e \in E_T, a \in w(V_1^e) \mid l(e) > l(a)$

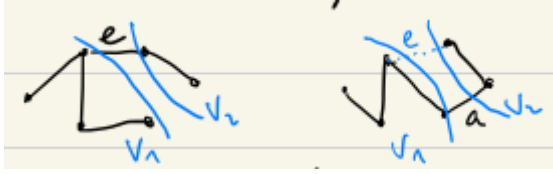


En T existe un único camino entre u y v , γ .

Haciendo $\gamma \cup \{a\}$, entonces tenemos un ciclo, y por hipótesis se verifica $l(e) \leq l(a) \#$ contradicción, habíamos supuesto lo contrario.

[3 \implies 1] Sea T el árbol generador que verifica (3). Sea T^* un árbol generador de peso mínimo:

- Si $T = T^*$, FIN, T es de peso mínimo
- Si no, sea $e \in E_T \setminus E_{T^*}$. Al añadir e a T^* se forma un ciclo. Si quitamos e de T , entonces se forman dos componentes conexas V_1, V_2 .



$w(V_1^e)$ es el cociclo

Al hacer $T^* \cup \{e\}$ se forma un ciclo que contiene una arista $a \in w(V_1^e) \implies l(e) \leq l(a)$. Si hacemos

$$\overline{T^*} = T^* - \{a\} \cup \{e\}$$

entonces

$$l(\overline{T^*}) \leq l(T^*)$$

pero T^* es minimal, luego

$$l(\overline{T^*}) = l(T^*)$$

y entonces

$$l(e) = l(a)$$

Repetiendo este proceso, vemos como toda arista de T tiene asociada una arista en T^* que pesa lo mismo. Por tanto

$$l(T) = l(T^*)$$

y T es minimal.

□

Algoritmo de Kruskal para encontrar un árbol generador minimal

Paso 1

Ordenar las aristas de E en orden ascendente por peso

Paso 2

Añadir $n - 1$ aristas a T , por orden de peso, sin formar ciclos

Algoritmo de Prim para encontrar un árbol generador minimal

Paso 1

Tomamos un vértice $r \in V$, hacemos $V_1 = \{r\}$ y $V_2 = V \setminus \{r\}$

Paso 2

Añadimos al árbol la arista de menor peso de $[V_1, V_2]$. Si esta es (v_1, v_2) con $v_1 \in V_1$ y $v_2 \in V_2$, hacemos $V_1 = V_1 \cup \{v_2\}$ y $V_2 = V_2 \setminus \{v_2\}$

Paso 3

Si $|V_1| = n$, FIN

Si no, ir al Paso 2

Algoritmo de Sollin para encontrar un árbol generador minimal

Es una variante de Prim, en la que hacemos Prim en todos los nodos simultáneamente.

Paso 1

Para cada $i \in V$, hacer $N_i = \{i\}$, $T = \emptyset$

Paso 2

Para árbol N_i , determinar el eje (u_i, v_i) tal que

$$l(u_i, v_i) = \min \{l(u, v) : (u, v) \in E, u \in N_i, v \notin N_i\}$$

Paso 3

Para cada N_i , si u_i, v_i pertenecen a árboles diferentes, unir dichos árboles en uno solo y hacer

$$T = E \cup \{(u_i, v_i)\}$$

Paso 4

Si $|T| = n - 1$, FIN

Si no, ir al Paso 2

4 Caminos más cortos. Recorridos por aristas y vértices

Definition 4.1. Sea (V, E, l) una red y P un camino en G . Se define la **longitud del camino** P como

$$l(P) = \sum_{a \in P} l(a)$$

El **problema del camino más corto entre v_i y v_j** en G es

$$\begin{aligned} \min & \quad l(P) \\ \text{s.a.} & \quad P \text{ es un camino entre } v_i \text{ y } v_j \end{aligned}$$

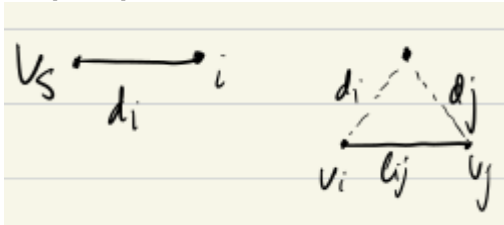
En lo que sigue para una arista $e = (u, v)$ denotaremos indistintamente $l(u, v)$, l_e , l_{uv} . También supondremos que la red es conexa y que $l_e \geq 0$, $\forall e \in E$.

Remark 4.2. Dado un camino más corto de v_a a v_b , todo subcamino de este que conecte v_i con v_j es un camino más corto entre v_i y v_j (o sea, los subcaminos de los caminos mínimos, son caminos mínimos entre los nodos que unen).

Theorem 4.3. El vector $d = (d_1, \dots, d_n)$ que contiene longitudes de caminos que conectan el nodo v_s con todos los nodos v_1, \dots, v_n de una red, con $d_s = 0$, mide las longitudes de los caminos más cortos entre v_s y todos los demás nodos si, y solo si

$$d_j - d_i \leq l_{ij}, \quad \forall (v_i, v_j) \in E$$

Proof. [\Rightarrow]



$\forall (v_i, v_j) \in E$, $d_i + l_{ij}$ mide la longitud del camino entre v_s y v_j , pasando justo antes por v_i , como d_j es mínimo, entonces

$$d_i + l_{ij} \geq d_j \implies d_j - d_i \leq l_{ij}$$

[\Leftarrow] Sea P un camino cualquiera de v_s a v_j

$$P = v_s v_{i1} v_{i2} \dots v_{ik} v_j$$

entonces

$$\begin{aligned} d_{i1} - d_s &\leq l(v_s, v_{i1}) \\ d_{i2} - d_{i1} &\leq l(v_{i1}, v_{i2}) \\ &\dots \\ d_j - d_{ik} &\leq l(v_{ik}, v_j) \end{aligned}$$

si los sumamos todos, se van cancelando los sumandos de la izquierda, y queda ($d_s = 0$)

$$d_j = d_j - d_s \leq l(v_s, v_{i1}) + \dots + l(v_{ik}, v_j) = l(P)$$

luego d_j es la longitud del camino más corto entre v_s y v_j . □

Theorem 4.4. Supongamos que el vector $d = (d_1, \dots, d_n)$ contiene longitudes de caminos que conectan el nodo v_s con todos los nodos v_1, \dots, v_n de una red, con $d_s = 0$. Sea V' un conjunto de vértices para el cual

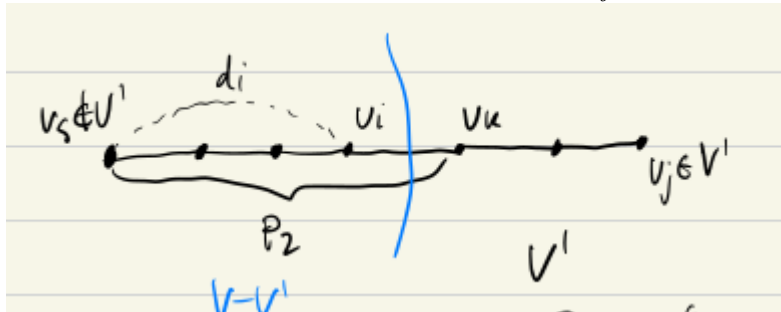
1. $v_s \notin V'$
2. si $v_i \notin V'$, entonces d_i es la longitud de un camino más corto de v_s a v_i
3. si $v_i \in V'$, entonces $d_i = \min \{d_j + l_{ji} : v_j \notin V', v_j \in N(v_i)\}$

Sea $v_j \in V'$ tal que

$$d_j = \min \{d_i : v_i \in V'\}$$

Entonces d_j es la longitud de un camino más corto de v_s a v_j

Proof. Sea $V' = V \setminus \{v_s\}$ y P un camino de v_s a v_j con $v_s \notin V'$ y $v_j \in V'$.



Entonces

$$l(P) \geq l(P_2) \stackrel{2.}{\geq} d_i + l_{ik} \stackrel{3.}{\geq} d_k \geq \min d_j$$

y se tiene que d_j es la longitud de un camino más corto de v_s a v_j . □

Algoritmo de Dijkstra

Sirve para obtener el vector de distancias mínimas de v_s a los demás nodos de la red y el vector de predecesores que permite reconstruir los caminos.

Paso 1

Elegimos v_s

$D = (d_1, \dots, d_n)$ (vector de distancias)

$$d_i = \begin{cases} l_{si} & i \in N(v_s) \\ 0 & i = s \\ \infty & \text{otro caso} \end{cases}$$

$P = (p_1, \dots, p_n)$ (vector de predecesores)

$$p_i = \begin{cases} s & i \in N(v_s) \\ 0 & \text{otro caso} \end{cases}$$

$V' = V \setminus \{v_s\}$ (nodos que faltan por visitar)

Paso 2

Buscar $v_j \in V'$ tal que $d_j = \min \{d_i : v_i \in V'\}$

Hacemos $V' = V' \setminus \{v_j\}$

Paso 3

$\forall v_i \in N(v_j) \cap V'$, si $d_j + l_{ji} < d_i$ hacer

$$d_i = d_j + l_{ji}$$

$$p_i = j$$

Si $V' \neq \emptyset$, volver al Paso 2

Si no, FIN

Definition 4.5. Dado un grafo $G = (V, E)$ se llama **tour euleriano** a un paseo cerrado que atraviesa cada arista de E exactamente una vez.

Un grafo que admite un tour euleriano se denomina **grafo euleriano**.

Theorem 4.6. Un grafo conexo es euleriano si, y solo si, todos los vértices son pares

Algoritmo de Fleury

Para obtención de tours Eulerianos

Paso 1

Hacer una copia $G' = (V', E')$ del grafo.

Tomar cualquier $i \in V'$

Paso 2

- Si $\exists (i, j) \in E'$ que no es puente en G' , elegimos ese eje.
- Si no, elegimos cualquier eje de E' .

Paso 3

Añadimos (i, j) al paseo y hacemos $E' = E' \setminus \{(i, j)\}$.

Si $E' = \emptyset$, FIN

Si no, hacemos $i := j$ y volvemos al Paso 2

Algoritmo de Hierholzer

Para obtención de tours eulerianos, una mejora del anterior.

Paso 1

Comenzando con una copia del grafo $G' = (V', E')$

Elegimos $i \in V'$

Paso 2

- Si se puede elegir $(i, j) \in E'$, anotar (i, j) en el paseo y eliminarla.
Si $E' = \emptyset$, FIN
Si no, hacemos $i := j$ y volvemos al Paso 2
- Si no se puede elegir (i, j) de E' , reemplazamos i por un vértice ya perteneciente al paseo en el que incida una arista de E' .
Sea (j, i) o (i, k) un tramo del paseo. Hacer $e = (j, i)$.
Volvemos al Paso 2, anotando la arista en el interior del paseo a continuación de e y reemplazando e por la arista anotada.

Definition 4.7. Dada una red (V, E, l) a la determinación del paseo cerrado de mínima longitud que incluye todas las aristas de E al menos una vez se le llama **problema del cartero chino**

Algoritmo de Dantzig

Para la obtención de los caminos mínimos entre todo par de vértices

Paso 1

D es la matriz de distancias y P la matriz de predecesores

Hacemos $k = 1$ y $\forall v_i, v_j \in V$ hacemos

$$D(i, j) = \begin{cases} 0 & i = j \\ \infty & \text{otro caso} \end{cases}$$

$$P(i, j) = i$$

Paso 2

Para $i = 1, \dots, k$

$$D(i, k+1) = \min_{j=1, \dots, k} \{D(i, j) + l_{j, k+1}\} = D(k+1, j)$$

$$P(i, k+1) = \arg \min_{j=1, \dots, k} \{D(i, j) + l_{j, k+1}\}$$

es decir, seleccionamos en esta última expresión el j que nos da el mínimo en la expresión anterior.

- Si $P(i, k+1) \neq i$, entonces

$$P(k+1, i) = P\left(\arg \min_{j=1, \dots, k} \{D(i, j) + l_{j, k+1}\}, i\right)$$

- Si $P(i, k+1) = i$, entonces

$$P(k+1, i) = k+1$$

Paso 3

Para todo par $i, j = 1, \dots, k$ hacer

$$D(i, j) = \min \{D(i, j), D(i, k+1) + D(k+1, j)\}$$

es decir, actualizamos la matriz de distancias, la distancia mínima será la menor entre la que había antes y la suma de las distancias mínimas correspondientes al añadir el nuevo nodo.

Si el valor cambia respecto al anterior, entonces $P(i, j) = P(k+1, j)$.

Hacer $k = k+1$.

Si $k < n$, volver al Paso 2

Resolución del problema del cartero chino

1. Usar el algoritmo de Dantzig para calcular las longitudes de los caminos más cortos entre cada par de nodos de la red
2. Calcular el coste de todos los emparejamientos (de nodos de grado impar) posibles sumando las longitudes obtenidas en el paso anterior
3. Seleccionar el emparejamiento de menor coste
4. Duplicar, en la red original, todas las aristas pertenecientes a los caminos más cortos para el emparejamiento elegido
5. En el multigrafo euleriano así obtenido, encontrar un ciclo euleriano

Definition 4.8. Un ciclo en un grafo G es un **ciclo hamiltoniano** si contiene a todos los vértices del grafo.

Si tal ciclo existe, se dice que G es un **grafo hamiltoniano**.

Un **camino hamiltoniano** es aquel que contiene todos los vértices del grafo.

Definition 4.9. Dado un grafo G , se obtiene el **grafo clausura**, $[G]$ mediante el proceso iterativo siguiente:

1. Se define $G_0 = (V, E_0) := G$, $j = 0$
2. Mientras exista un par de nodos u, v en G_j tales que $(u, v) \notin E_j$ y $o(u) + o(v) \geq n$:
 - (a) hacer $j = j + 1$
 - (b) hacer $E_j = E_{j-1} \cup \{(u, v)\}$
 - (c) hacer $G_j = (V, E_j)$, volver a 2.

Theorem 4.10. Un grafo G es hamiltoniano si, y solo si, su grafo clausura $[G]$ es hamiltoniano.

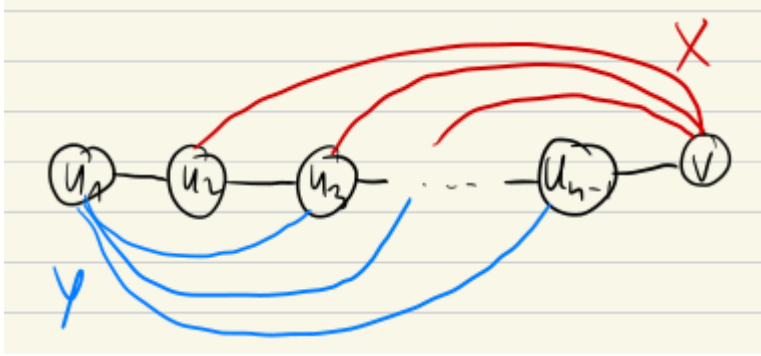
Proof. [\implies] Evidente, si añadimos ejes a un grafo hamiltoniano sigue siendo hamiltoniano.

[\impliedby] Supongamos que G no es hamiltoniano y $[G]$ sí que lo es. Tendremos entonces una primera arista (u, v) entre vértices no adyacentes en G_j , que no es hamiltoniano, tal que G_{j+1} es hamiltoniano. Entonces existía un camino hamiltoniano de u a v en G_j :

$$u = u_1, \dots, u_n = v$$

de forma que u, v verifican que $g(u) + g(v) \geq n$ (por eso los habíamos elegidos para añadir (u, v) a G_j). Sean los conjuntos

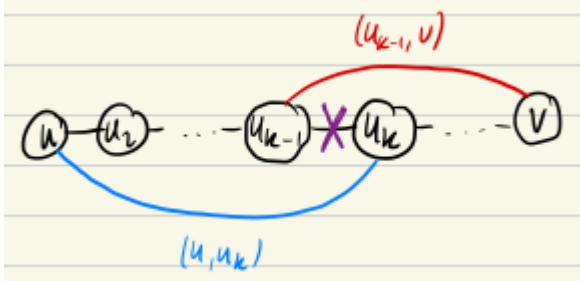
$$X = \{i \in \{3, \dots, n-1\} : (u_{i-1}, v) \in E_j\}$$



$$Y = \{i \in \{3, \dots, n-1\} : (u, u_i) \in E_j\}$$

entonces $|X| = g(v) - 1$ e $|Y| = g(u) - 1$, por lo que $|X| + |Y| = g(u) + g(v) - 2 \geq n - 2$.

Como $X, Y \subset \{3, \dots, n-1\}$ que tiene $n-3$ elementos, entonces $X \cap Y \neq \emptyset$, o sea $\exists k \in X \cap Y$



Y si eliminamos (u_{k-1}, u_k) entonces tendremos un ciclo hamiltoniano, luego G_j es hamiltoniano# pero esto contradice lo que habíamos supuesto. Por tanto, G debe ser hamiltoniano. \square

Definition 4.11. Una n -tupla de naturales (a_1, \dots, a_n) es una **sucesión hamiltoniana** si todos los grafos de n nodos de grados $g_1 \leq \dots \leq g_n$ que verifican $g_1 \geq a_1, \dots, g_n \geq a_n$ son hamiltonianos.

Theorem 4.12. Para cualquier $n \geq 3$, una n -tupla de naturales (a_1, \dots, a_n) con $a_1 \leq \dots \leq a_n < n$ es una sucesión hamiltoniana si, y solo si, para cada $i < \frac{n}{2}$ se satisface

$$a_i \leq i \implies a_{n-1} \geq n - i$$

Proof. [\Leftarrow] Sea G no hamiltoniano con el mayor número posible de aristas (es decir, si añadiésemos una arista pasaría a ser hamiltoniano), con grados $(g_1, \dots, g_n) \geq (a_1, \dots, a_n)$ de forma que $a_i \leq g_i \leq i \implies g_{n-i} \geq a_{n-i} \geq n - i$.

Sean u, v no adyacentes en G tales que $g(u) + g(v)$ es máximo y podemos suponer que $g(u) \leq g(v)$.

Entonces $G \cup \{(u, v)\}$ es hamiltoniano (G tenía el mayor número posible de aristas sin ser hamiltoniano). Por tanto, en G existe un camino hamiltoniano de u a v

$$u = u_1, u_2, \dots, u_n = v$$

Sean

$$S = \{i : (u, u_{i+1}) \in E\}$$

$$T = \{i : (u_i, v) \in E\}$$

Entonces $S \cap T = \emptyset$, porque si no encontramos un ciclo hamiltoniano, como en la demostración anterior, y $S \cup T \subset \{1, \dots, n-1\}$, por lo que $g(u) + g(v) = |S| + |T| < n$.

Como $g(u) + g(v)$ es maximal, entonces $g(u_j) \leq g(u) < \frac{n}{2}$, $\forall j \in S$ (habíamos supuesto que $g(u) \leq g(v)$).

Así, hay al menos $g(u) = |S|$ vértices cuyos grados son menores o iguales que $g(u)$, por tanto

$$g_{g(u)} \leq g(u) \stackrel{\text{hipótesis}}{\implies} g_{n-g(u)} \geq n - g(u)$$

Por lo que hay al menos $g(u) + 1$ vértices con grado mayor o igual que $n - g(u)$.

Como u tiene grado $g(u)$, no puede estar conectado a todos ellos. Por tanto, $\exists w | g(w) \geq n - g(u)$ no adyacente a u .

Entonces

$$g(u) + g(w) \geq g(u) + n - g(u) = n$$

Lo cual contradice que $g(u) + g(v) < n$ fuese maximal.

Por tanto, G debe ser hamiltoniano y la sucesión es hamiltoniana. \square

Definition 4.13. Dada una red, la determinación del ciclo hamiltoniano de mínima longitud se llama **problema del viajante de comercio**

Algoritmo de Christofides

Heurística para el problema del viajante.

Dada una red con G no dirigido, completo y cuyas distancias verifican la desigualdad triangular:

Paso 1

Encontrar en G un árbol generador minimal, T

Paso 2

Sea V_I el conjunto de los vértices de grado impar en T .

Sea M el emparejamiento mínimo asociado a V_I .

Paso 3

Sea $H = (V, T \cup M)$, que es euleriano.

Sea C un ciclo euleriano en H

Paso 4

Obtener C' ciclo hamiltoniano asociado a C , saltando los vértices ya visitados hasta un vértice no visitado, o hasta el vértice de partida.

El algoritmo de Christofides verifica que, si C^* es la solución óptima, entonces

$$l(C') \leq \frac{3}{2}l(C^*)$$

5 Coloración de grafos. Grafos planos

5.1 Coloración

Definition 5.1. Sea $G = (V, E)$ un grafo simple y S un conjunto de colores distintos. Una **coloración propia por vértices** o coloración de G es una aplicación $f : V \rightarrow S$ que asigna colores distintos a vértices adyacentes

$$(u, v) \in E \implies f(u) \neq f(v)$$

Cuando $|S| = k$ se dice que es una **k -coloración**
Un grafo es **k -coloreable** si admite una k -coloración

Nótese que una k -coloración induce una partición del conjunto de vértices V en subconjuntos V_1, \dots, V_k independientes.

Definition 5.2. El **número cromático** de un grafo G , $\mathcal{X}(G)$ es el menor número de colores necesario para colorear G

Remark 5.3. Algunos números cromáticos:

- $\mathcal{X}(K_n) = n$
- $\mathcal{X}(G) = 1 \iff G$ no tiene ejes, $|E| = 0$
- $\mathcal{X}(G) = 2 \iff G$ es bipartito y $|E| \geq 1$
- Si T es un árbol, $\mathcal{X}(T) = 2$
- Si C_n es el ciclo de tamaño n

$$\mathcal{X}(C_n) = \begin{cases} 2 & n \text{ par} \\ 3 & n \text{ impar} \end{cases}$$

- Si H es un subgrafo de G , entonces $\mathcal{X}(H) \leq \mathcal{X}(G)$
- Si K_q es subgrafo de G , entonces $\mathcal{X}(G) \geq q$

El número cromático de un grafo coincide con el mínimo número de conjuntos independientes en que se puede particionar V .

Algoritmo heurístico Greedy para colorear un grafo

Sea G un grafo simple

Paso Inicial

Ordenar los vértices de $V = \{v_1, \dots, v_n\}$.

Ordenar los colores $S = \{1, 2, \dots\}$.

Hacer $k = 1$.

Paso general

Asignar a v_k el primer color en S el primer color en S que no ha sido asignado previamente a ninguno de sus nodos adyacentes del conjunto $\{v_1, \dots, v_k\}$.

Hacer $k = k + 1$.

Este algoritmo no garantiza que la coloración sea óptima, y el resultado es dependiente de la forma en la que se ordenan los nodos y los colores en el paso inicial.

Dado que en cada paso del algoritmo se comparan los colores de los vecinos de un nodo y se elige el primero disponible, nunca harán falta más colores que el grado mayor más 1. Así, parece preferible ordenar los nodos en orden decreciente de grado.

Proposition 5.4. Para todo grafo G

$$\chi(G) \leq \Delta_G + 1$$

Theorem 5.5. de Brooks

Sea G un grafo conexo que no es ni un grafo completo ni un ciclo de longitud impar. Entonces

$$\chi(G) \leq \Delta_G$$

5.1.1 Grafos críticos

Definition 5.6. Un grafo G es **crítico** si para cada subgrafo H de G se tiene que $\chi(H) < \chi(G)$.

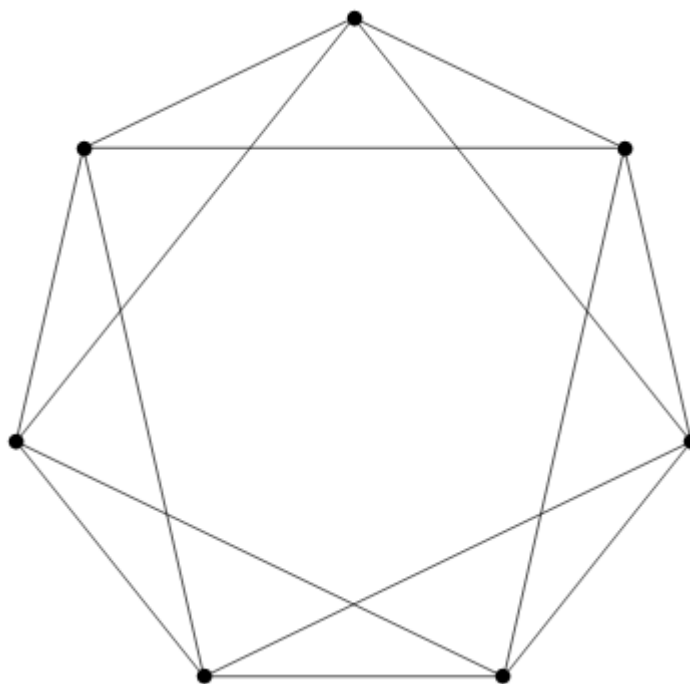
G es **crítico por ejes** si $\chi(G - e) < \chi(G)$, $\forall e \in E$.

G es **crítico por vértices** si $\chi(G - v) < \chi(G)$, $\forall v \in V$

Un grafo crítico con $\chi(G) = k$ se denomina **k -crítico**

Se verifica:

- G crítico $\iff G$ crítico por ejes y por vértices
- G crítico por ejes y no tiene vértices aislados $\implies G$ crítico por vértices
- G crítico por vértices no implica crítico por ejes, como contraejemplo se tiene el grafo circulante $C(7, 2)$



Proposition 5.7. Cada grafo G de orden $n \geq 2$ contiene un subgrafo crítico H tal que $\chi(G) = \chi(H)$

Proof. Si G es crítico, ya está.

Si no es crítico, entonces existe $e_1 \in E : \chi(G - e_1) = \chi(G)$. Llamamos $H_1 = G - e_1$:

- Si H_1 es crítico, ya está
- Si no es crítico, entonces existe $e_2 \in E : \chi(G - e_2) = \chi(G)$. Llamamos $H_2 = G - e_2$:
- ...

Como E es finito, en algún momento se alcanza H_n un subgrafo de G crítico y tal que $\chi(H) = \chi(G)$ □

Theorem 5.8. Si G es un grafo crítico, entonces $\chi(G) \leq \delta_G + 1$

Proof. Supongamos que G es k -crítico.

Supongamos que $\delta_G \leq k - 2$

Sea $v \in V$ un vértice de menor grado.

Como G es crítico

$$\chi(G - v) = \chi(G) - 1$$

pues al quitar un vértice, quitamos sus ejes adyacentes, por ser crítico este número debe ser menor que el original. Pero como solo quitamos un vértice, como mucho disminuye en una unidad.

Así, en una coloración de $G - v$ se utilizarían, a lo sumo, $k - 2$ colores para colorear los vértices adyacentes a v en G .

Luego hay, al menos, un color que no se ha utilizado. Pintando v de ese color, tendríamos una $k - 1$ -coloración de G # contradicción. Por tanto, debe ser

$$\delta_G > k - 2 \implies k < \delta_G + 2 \implies k \leq \delta_G + 1$$

□

5.1.2 Polinomio cromático

Definition 5.9. Dado un grafo G , su **polinomio cromático**, $p(G, k)$ se define como el número de formas de colorear el grafo G con a lo sumo k colores.

Claramente, $p(G, k) = 0, \forall k < \chi(G)$.

Remark 5.10. Algunos polinomios cromáticos:

- $p(K_n, k) = k \cdot (k - 1) \cdot \dots \cdot (k - n + 1) = \left[\frac{k!}{n!} \right], \forall k \geq 0$
- si P_n es el grafo cadena de n nodos, entonces $p(P_n, k) = k(k - 1)^{n-1}, \forall k \geq 0$

Definition 5.11. Sea $G = (V, E)$ un grafo simple y $e = (u, v) \in E$. El grafo $G \circ e$ se denomina **grafo contrario** y es el grafo resultante de contraer los nodos u y v en uno solo y reducir a uno los posibles ejes paralelos.

En concreto, $G \circ e = (V', E')$ donde

- $V' = (V \setminus \{u, v\}) \cup \{uv\}$
- $E' = E_{V \setminus \{u, v\}} \cup \{(uv, i) : (u, i) \in E \text{ o } (v, i) \in E\}$

Theorem 5.12. de Reducción

Sea G un grafo simple y u, v dos vértices no adyacentes en G . Sea $e = (u, v)$. Entonces

$$p(G, k) = p(G + e, k) + p(G \circ e, k)$$

Corollary 5.13. Sea G un grafo simple y $(u, v) \in E$. Entonces

$$p(G, k) = p(G - e, k) - p(G \circ e, k)$$

Proposition 5.14. El polinomio cromático $p(G, k)$ de un grafo G de orden n tiene grado n . El coeficiente de término k^n es 1, el término independiente es 0 y los coeficientes de los términos intermedios son enteros y se alternan en signo.

5.2 Planaridad

Definition 5.15. Un grafo es un **grafo planar** si existe una representación de G en el plano de modo que ningún par de ejes se cruza entre sí.

Una representación concreta de un grafo planar en el que ningún par de ejes se cruza entre sí se denomina **grafo plano** (o embutido).

Remark 5.16. Algunos grafos planares:

- Cualquier ciclo de orden n , C_n , es planar
- Cualquier grafo cadena, P_n , es planar
- Cualquier estrella es planar
- Cualquier árbol es planar

5.2.1 Identidad de Euler

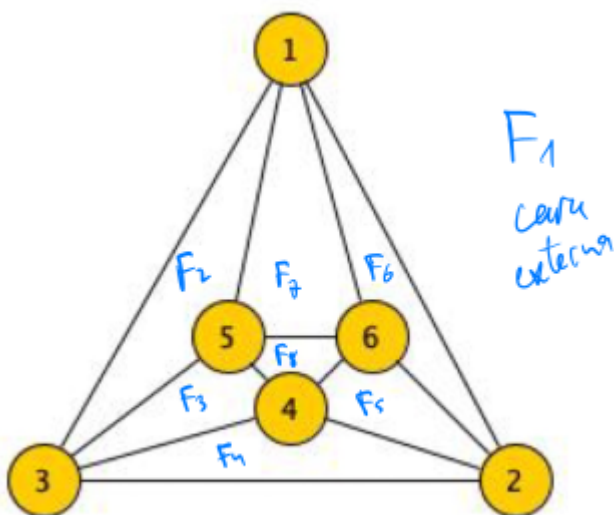
Cualquier grafo plano divide \mathbb{R}^2 en un conjunto de regiones denominadas **caras**. Como el grafo es acotado, una de las caras es no acotada y se denomina **cara externa**. El **contorno** de una cara es la unión de los ejes y vértices incidentes a dicha cara.

En un grafo plano se verifican las siguientes propiedades:

- Dos caras son disjuntas y sus contornos solo se intersecan en los ejes
- Hay una única cara externa

- El interior de cada ciclo de G contiene al menos una cara interna
- Un puente pertenece al contorno de una sola cara (y viceversa)
- Un eje que no es puente pertenece al contorno de dos caras (y viceversa)
- Se define el **grado de una cara F** , $d(F)$, como el número de ejes incidentes a la misma, teniendo en cuenta que los puentes cuentan con grado 2. Por tanto

$$\sum_F d(F) = 2m$$



Theorem 5.17. Identidad de Euler

Si G es un grafo plano conexo con n vértices, m ejes y c caras, se cumple que $n - m + c = 2$

Proof. Inducción sobre c :

- $c = 1$, entonces G no tiene ciclos y es conexo, por lo que es un árbol y entonces $m = n - 1$, por tanto

$$n - m + c = n - n + 1 + 1 = 2 \checkmark$$

- Supongamos que es cierto para $c - 1$ caras y que G tiene $c \geq 2$ caras.

Como $c \geq 2$, entonces G tiene al menos un ciclo C .

Sea $e \in C$, que no es un puente. Entonces $G - e$ tiene $n - 1$ caras y $G - e$ sigue siendo conexo. Por la inducción, tenemos que

$$n - (m - 1) + (c - 1) = 2 \implies n - m + c = 2$$

□

Corollary 5.18. *Todos los grafos planos de un grafo planar tienen el mismo número de caras*

Proposition 5.19. *Si G es un grafo planar con $n \geq 3$, entonces $m \leq 3n - 6$.
Si G no contiene a K_3 , entonces $m \leq 2n - 4$.*

Proof. Veamos primero el resultado general.

Si $m \leq 2$, es trivial ($n \geq 3$ por hipótesis)

Si $m \geq 3$, entonces el contorno en cada cara tiene, al menos, 3 ejes:

$$3c \leq \sum_F d(F) = 2m \implies 3(2 - n + m) \leq 2m \implies 6 - 3n + 3m \leq 2m \implies m \leq 3n - 6$$

Y tenemos el resultado.

Si ahora G no contiene a K_3 , entonces cualquier cara está contorneada por, al menos, 4 ejes

$$4c \leq 2m \implies 2c \leq m \implies 2(2 - n + m) \leq m \implies 4 - 2n + 2m \leq m \implies m \leq 2n - 4$$

□

Remark 5.20. Este resultado proporciona condiciones necesarias, pero no suficientes. Como contraejemplo encontramos el grafo de Petersen.

5.2.2 Grafo planar maximal

Definition 5.21. Un grafo planar es **maximal** si la adición de cualquier arista lo convierte en no planar.

Lemma 5.22. *Sea F una cara de un grafo plano G con al menos cuatro ejes en su contorno. Entonces hay dos vértices no adyacentes en F .*

Corollary 5.23. *Si G es un grafo planar maximal con $n \geq 3$, el contorno de cada cara de un grafo plano de G tiene exactamente 3 aristas.*

Theorem 5.24. *En todo grafo planar maximal se verifica que $m = 3n - 6$*

5.2.3 Teorema de Kuratowski

Definition 5.25. Una **subdivisión de un eje** $e = (u, v)$ de un grafo G se obtiene introduciendo en G un nuevo vértice w y reemplazando el eje e por los ejes (u, w) , (w, v) . El grafo resultante de realizar sucesivas subdivisiones sobre los ejes de un grafo G se denomina **subdivisión** de G .

Theorem 5.26. de Kuratowski

Un grafo es planar si, y solo si, no contiene como subgrafo ninguna subdivisión de K_5 ni de $K_{3,3}$

5.3 Coloración en grafos. El teorema de los 4 colores

Theorem 5.27. de los 4 colores

Todo grafo planar se puede colorear con, a lo sumo, 4 colores

6 Programación lineal entera

6.1 El modelo de programación lineal entera

Definition 6.1. Un **modelo de programación lineal entera** es un modelo de programación lineal en el que algunas o todas las variables de decisión deben tomar valores enteros. El modelo general con n variables de decisión y m restricciones se puede expresar del siguiente modo:

$$\begin{aligned} (P_E) \max & \quad \sum_{i=1}^n c_i x_i \\ \text{s.a.} & \quad \sum_{i=1}^n a_{ji} x_i \leq b_j, \quad \forall j = 1, \dots, m \\ & \quad x_i \geq 0, \quad \forall i = 1, \dots, n \\ & \quad x_i \in \mathbb{Z}, \quad \forall i \in I \subset \{1, \dots, n\} \end{aligned}$$

Cuando todas las variables de decisión son enteras, el problema se denomina **entero puro**, cuando no son todas, se dice **entero mixto**.

En este tipo de problemas se pueden aplicar las mismas transformaciones que en un problema de programación lineal:

- Modificar el sentido de optimización del problema
- Cambiar el sentido de una restricción
- Convertir una restricción de igualdad en dos restricciones de desigualdad
- Transformar variables negativas en positivas
- Transformar una variable no restringida en dos variables positivas

Recordemos que, en programación lineal, la región factible era una región convexa, concretamente un poliedro. Esto no es así ahora, en los modelos de PLE la región factible no tiene por qué ser un poliedro, incluso puede no ser convexa.

Definition 6.2. Dado un problema de programación lineal entera (P_E) , denotaremos por (P_L) al problema lineal asociado resultante de eliminar de las variables las restricciones de integridad. Este problema se denomina **relajación lineal** del problema (P_E)

$$\begin{aligned} (P_E) \max & \quad \sum_{i=1}^n c_i x_i \\ \text{s.a.} & \quad \sum_{i=1}^n a_{ji} x_i \leq b_j, \quad \forall j = 1, \dots, m \\ & \quad x_i \geq 0, \quad \forall i = 1, \dots, n \end{aligned}$$

Remark 6.3. El valor óptimo z_L de la relajación lineal de un modelo PLE es al menos tan bueno como el valor óptimo z_E del problema entero.

Remark 6.4. En los modelos de PLE se pueden contruir infinidad de problemas con distintos conjuntos de restricciones, pero con la misma región factible.

El siguiente resultado pone de manifiesto que si los coeficientes de las restricciones son racionales, la envoltura convexa del conjunto factible es un poliedro.

Theorem 6.5. Dadas matrices de coeficientes racionales A , G y un vector de coeficientes racionales b , sea $P = \{(x, y) : Ax + Gy \leq b\}$ y sea $S = \{(x, y) \in P : x \text{ entero}\}$.
Entonces, existen matrices racionales A' , G' y un vector de coeficientes racionales b' tales que

$$\text{conv}(S) = \{(x, y) : A'x + G'y \leq b'\}$$

Los problemas lineales enteros pueden ser no acotados o infactibles. El siguiente resultado indica que si un problema lineal entero es factible, podemos establecer si es acotado o no mediante su relajación lineal.

Proposition 6.6. Sea $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ un poliedro de coeficientes racionales tal que el poliedro $P_I = \text{conv}(P \cap \mathbb{Z}^n)$ sea no vacío. Sea $c \in \mathbb{R}^n$.
El problema $\max \{c'x : x \in P_I\}$ es acotado si, y solo si, $\max \{c'x : x \in P\}$ es acotado.

6.2 Matrices totalmente unimodulares

En algunos PLE, la solución del correspondiente problema lineal relajado da directamente una solución entera.

Cuando un poliedro tiene sus puntos extremos con coordenadas enteras, independientemente de la función a optimizar, la solución será entera. Tiene sentido preguntarse si podemos establecer condiciones bajo las cuales todos los puntos extremos de un poliedro tengan coordenadas enteras.

Definition 6.7. Una matriz A se dice que es **totalmente unimodular** si el determinante de cualquier submatriz cuadrada de A vale 1, -1 ó 0 .
Una matriz es **entera** si todas sus entradas son números enteros

Esto implica, en particular, que todas las entradas de una matriz totalmente unimodular tienen que ser 1, -1 ó 0 .

Definition 6.8. Sea A de tamaño $m \times n$ y rango m , A es **unimodular** si sus coeficientes son enteros y el determinante de cualquier submatriz invertible de orden m es 1 ó -1

Lemma 6.9. Lema 1

Una matriz A es totalmente unimodular si y solo si $[I, A]$ es unimodular

Proof. El determinante de cualquier submatriz cuadrada de A es ± 1 si, y solo si, el determinante de cualquier submatriz de orden m de $(I|A)$ es ± 1 □

Definition 6.10. Un poliedro $P \subset \mathbb{R}^n$ se dice **entero** si todos sus puntos extremos son enteros

Lemma 6.11. Lema 2

x es un punto extremo de $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ si, y solo si, $(x, b - Ax)$ es un punto extremo de $Q = \{(x, y) \in \mathbb{R}^{n+m} : Ax + Iy = b, x, y \geq 0\}$

Proof. [\implies]

$$Ax \leq b \implies b - Ax \geq 0 \implies (x, b - Ax) \geq 0$$

$$Ax + I(b - Ax) = b$$

[\impliedby]

$$(x, b - Ax) \geq 0 \implies b - Ax \geq 0 \implies Ax \leq b$$

□

Lemma 6.12. Lema 3 (de Voinott y Dantzig)

Sea A una matriz entera de tamaño $m \times n$ y rango m .

A es unimodular si, y solo si, para cada $b \in \mathbb{Z}^m$ el poliedro $Q = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ es entero

Proof. [\implies] A unimodular. Sean $b \in \mathbb{Z}^m$ y x un punto extremo de Q .

Entonces x es SBF (solución básica factible) y se puede escribir

$$x = (B^{-1}b, 0)$$

donde B es una submatriz cuadrada de orden m de A , no singular (matriz básica).

Como $B^{-1} = \frac{adj(B)}{|B|}$ y B es entera, entonces B^{-1} es entera, y, por tanto, $B^{-1}b$ es entero. Así, x es entero, y entonces Q es entero.

[\impliedby] Sea B una submatriz (entera) no singular de A de orden m , ¿se tendrá $|B| = \pm 1$?

Existe $y \in \mathbb{Z}^m$ tal que $z = y + B^{-1}e^i \geq 0 \implies b = Bz = By + e^i \in \mathbb{Z}$

Por tanto, $(z, 0)$ es SBF del sistema

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

y, entonces $(z, 0)$ es punto extremo de Q (expresado respecto de la matriz básica B). Por tanto, como Q es entero, $z \in \mathbb{Z}^m \implies z - y = B^{-1}e^i$ es entero, y eso quiere decir que la i -ésima columna de B^{-1} es entera, pero podemos hacerlo con cualquier i , por lo que B^{-1} es entera.

Eso quiere decir que $|B|, |B^{-1}| \in \mathbb{Z}$ y dado que $|B||B^{-1}| = 1$, debe ser

$$|B| = |B^{-1}| = \pm 1$$

□

Theorem 6.13. Hoffman y Kruskal

Sea A una matriz entera de tamaño $m \times n$. A es totalmente unimodular si, y solo si, para cualquier vector b de componentes enteras, los puntos extremos del poliedro $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ tienen componentes enteras

Proof. A es totalmente unimodular $\xLeftrightarrow{\text{Lema 1}} [I, A]$ es unimodular $\xLeftrightarrow{\text{Lema 3}} Q = \{x : Ax + Iy = b, x, y \geq 0\}$ es entero, para todo $b \in \mathbb{Z}^m$ $\xLeftrightarrow{\text{Lema 2}} \{x : Ax \leq b, x \geq 0\}$ es entero, para todo $b \in \mathbb{Z}^m$ □

6.2.1 Resultados y caracterizaciones de matrices totalmente unimodulares

Theorem 6.14. *Para una matriz A las siguientes condiciones son equivalentes:*

- A es totalmente unimodular
- A' es totalmente unimodular
- $[A, I]$ es totalmente unimodular
- La matriz obtenida al eliminar o añadir de A un vector fila o columna unitario (un vector de la matriz identidad) es totalmente unimodular
- La matriz obtenida de multiplicar una fila o columna de A por -1 es totalmente unimodular
- La matriz obtenida de intercambiar dos filas o columnas de A es totalmente unimodular
- La matriz obtenida de duplicar columnas o filas de A es totalmente unimodular

Definition 6.15. Una **matriz euleriana** es aquella en la que la suma de los elementos de cada fila y cada columna es par

Theorem 6.16. *Una matriz A es totalmente unimodular si, y solo si*

- $a_{ij} \in \{0, 1 - 1\}$, $\forall i, j$, y
- la suma de los elementos de cada submatriz cuadrada euleriana es múltiplo de 4

Theorem 6.17. *Una matriz A de tamaño $m \times n$ es totalmente unimodular si, y solo si, para cada conjunto de filas $F \subset \{1, \dots, m\}$ existe una partición $F = F_1 \cup F_2$ tal que*

$$\left| \sum_{i \in F_1} a_{ij} - \sum_{i \in F_2} a_{ij} \right| \leq 1, \quad \forall j = 1, \dots, n$$

Remark 6.18. Como una matriz es totalmente unimodular si, y solo si, su traspuesta lo es, el teorema anterior se puede reescribir en relación al conjunto de columnas.

Theorem 6.19. *Si una matriz A satisface que:*

- $a_{ij} \in \{0, 1, -1\}$, $\forall i, j$
- cada columna contiene a lo sumo dos elementos no nulos
- las filas de A se pueden particionar en dos conjuntos F_1 y F_2 tal que dos entradas no nulas en una misma columna pertenecen al mismo conjunto de filas si tienen signo diferente y están en conjuntos de filas distintos si tienen el mismo signo

Entonces A es totalmente unimodular

Corollary 6.20. Si una matriz A con coeficientes $\{1, -1, 0\}$ contiene en cada columna a lo sumo un coeficiente 1 y un coeficiente -1, entonces es unimodular

Definition 6.21. Una matriz A de tamaño $m \times n$ con coeficientes $\{1, 0\}$ se dice que es una **matriz de intervalo** si para cada $j \in \{1, \dots, n\}$ se cumple que si $a_{ij} = a_{kj} = 1$ con $i < k$, entonces $a_{hj} = 1$ para todo $i \leq h \leq k$

Corollary 6.22. Toda matriz de intervalo es totalmente unimodular

6.3 Formulación de modelos de programación lineal entera

- **Problema de asignación de tareas** (tiempo total y tiempo máximo)

Hay n operarios, cada uno puede realiar, a lo sumo, una tarea

Hay n tareas, que deben ser realizadas

El operario i realiza la tarea j en un tiempo $t_{ij} \geq 0$

Encontrar la asignación óptima de operarios a tareas de modo que el tiempo total sea mínimo.

Formulación:

Definimos

$$x_{ij} = \begin{cases} 1 & \text{si el operario } i \text{ es asignado a la tarea } j \\ 0 & \text{en otro caso} \end{cases}$$

y entonces es

$$\begin{aligned} \min & \quad \sum_{i,j} t_{ij} \cdot x_{ij} \\ \text{s.a.} & \quad \sum_{j=1}^n x_{ij} \leq 1, \quad \forall i = 1, \dots, n \\ & \quad \sum_{i=1}^n x_{ij} \geq 1, \quad \forall j = 1, \dots, n \\ & \quad x_{i,j} \in \{0, 1\} \end{aligned}$$

La primera restricción hace que cada operario solo pueda realizar una tarea (podría ponerse en igualdad) y la segunda hace que cada tarea deba ser hecha (también podría ponerse en igualdad).

La matriz asociada al problema es

$$A = \begin{pmatrix} x_{11} & \dots & x_{1n} & x_{21} & \dots & x_{2n} & \dots & x_{n1} & \dots & x_{nn} \\ 1 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & & \dots & \dots & & \dots & \dots & \dots & & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 & 0 & \dots & & 1 & 0 & \dots \\ 0 & 1 & \dots & 0 & 1 & \dots & & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 \end{pmatrix}$$

y la relajación lineal es

$$\begin{aligned} \min & \quad \sum_{i,j} t_{ij} \cdot x_{ij} \\ \text{s.a.} & \quad \sum_{j=1}^n x_{ij} \leq 1, \quad \forall i = 1, \dots, n \\ & \quad \sum_{i=1}^n x_{ij} \geq 1, \quad \forall j = 1, \dots, n \\ & \quad 0 \leq x_{i,j} \leq 1 \end{aligned}$$

- **Modificación del problema anterior: Minimax y Maximin**

– Minimax

$$\begin{aligned}
 & \min && z \\
 & \text{s.a.} && \\
 & && \sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \\
 & && \sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \\
 & && z \geq t_{ij}x_{ij} \\
 & && x_{i,j} \in \{0, 1\} \\
 & && z \geq 0
 \end{aligned}$$

– Maximin

$$\begin{aligned}
 & \max && z \\
 & \text{s.a.} && \\
 & && \sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \\
 & && \sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \\
 & && z \leq t_{ij}x_{ij} \\
 & && x_{i,j} \in \{0, 1\} \\
 & && z \geq 0
 \end{aligned}$$

- **Camino más corto entre dos vértices en una red**

Sea una red $R = (V, E, d)$, definimos, para todo $(i, j) \in E$

$$x_{ij} = \begin{cases} 1 & (i, j) \in C \\ 0 & \sim \end{cases}$$

donde C es el camino que vamos a dar como resultado. La formulación queda, tomando s como nodo inicial y t como final:

$$\begin{aligned}
 & \min && \sum d_{ij}x_{ij} \\
 & \text{s.a.} && \\
 & && \sum_{(s,j) \in E} x_{sj} = 1 \\
 & && \sum_{(i,t) \in E} x_{it} = 1 \\
 & && \sum_{(i,v) \in E} x_{iv} - \sum_{(v,j) \in E} x_{vj} = 0, \quad \forall v \neq s, t \\
 & && x_{i,j} \in \{0, 1\} \\
 & && \left(\sum_{(i,v) \in E} x_{iv} \leq 1 \right)
 \end{aligned}$$

La primera restricción nos dice que solo podemos salir una vez del nodo inicial.

La segunda que solo podemos entrar una vez al nodo final.

La tercera que debemos salir tantas veces como entramos a cada nodo intermedio.

La que está entre paréntesis hace que solo podamos usar una vez cada eje, aunque no es necesario añadirla, surge espontáneamente de las restricciones anteriores y de buscar minimización.

- **Árbol generador de mínimo coste**

Definimos, para todo $(i, j) \in E$

$$x_{ij} = \begin{cases} 1 & (i, j) \in AGMC \\ 0 & \sim \end{cases}$$

y el problema puede formularse como

$$\begin{aligned} \min & \sum c_{ij}x_{ij} \\ \text{s.a.} & \sum_{(i,j) \in E} x_{ij} = n - 1 \\ & \sum_{(i,j) \in E_S} x_{ij} \leq |S| - 1, \forall S \subset V \mid |S| \geq 3 \\ & x_{i,j} \in \{0, 1\} \end{aligned}$$

esta formulación se denomina **AUPM-1**.

La primera restricción nos dice que la solución debe tener tamaño $m = n - 1$

La segunda impide que se formen ciclos. Por tanto tenemos $m = n - 1$ y no hay ciclos, y es un árbol.

Una propiedad interesante de este problema es que la relajación lineal de este modelo produce soluciones enteras, a pesar de que A no es TU.

Otra formulación es la siguiente

$$\begin{aligned} \min & \sum c_{ij}x_{ij} \\ \text{s.a.} & \sum_{(i,j) \in E} x_{ij} = n - 1 \\ & \sum_{(i,j) \in [S, V \setminus S]} x_{ij} \geq 1, \forall S \subset V \mid S \neq \emptyset, V \\ & x_{i,j} \in \{0, 1\} \end{aligned}$$

que se denomina **AUPM-2**.

La primera restricción es la misma

La segunda nos dice que todo corte debe tener algún eje en la solución, lo que implica que la solución debe ser conexa. Por tanto tenemos $m = n - 1$ y conexo, por lo que es un árbol.

Esta vez la relajación lineal no produce necesariamente soluciones enteras.

- **Problema del viajante de comercio (Dantzig-Fullerson-Jhonson)**

Dos formulaciones:

– Definimos

$$x_{ij} = \begin{cases} 1 & \text{el recorrido visita } i \text{ y a continuación } j \\ 0 & \sim \end{cases}$$

y la formulación del modelo es

$$\begin{aligned} \min & \sum_{i \neq j} d_{ij}x_{ij} \\ \text{s.a.} & \sum_{j \in V, j \neq i} x_{ij} = 1, \forall i \in V \\ & \sum_{i \in V, i \neq j} x_{ij} = 1, \forall j \in V \\ & x_{i,j} \in \{0, 1\} \end{aligned}$$

La primera restricción nos dice que salimos una vez de cada nodo, y la segunda que entramos una vez a cada nodo.

– Definimos

$$x_e = \begin{cases} 1 & e \in \text{recorrido óptimo} \\ 0 & \sim \end{cases}$$

y el modelo queda

$$\begin{array}{ll} \min & \sum d_e x_e \\ \text{s.a.} & \sum_{e \in N(v)} x_e = 2, \quad \forall v \in V \\ & \sum_{e \in E_S} x_e \leq |S| - 1, \quad |S| \geq 3 \\ & x_e \in \{0, 1\} \end{array}$$

la primera restricción nos dice que para cada nodo tiene que haber dos aristas adyacentes a este en la solución, y la segunda que no deben formarse ciclos internos.

• **Formulación alternativa (Miller-Tuccker-Zemlin)**

$$x_{ij} = \begin{cases} 1 & i \rightarrow j \\ 0 & \sim \end{cases}$$

Sean $u_i \in \mathbb{R}^+$ y $V = \{0, \dots, n-1\}$ donde 0 es el origen. El modelo es

$$\begin{array}{ll} \min & \sum_{i \neq j} d_{ij} x_{ij} \\ \text{s.a.} & \sum_{j \in V, j \neq i} x_{ij} = 1, \quad \forall i \in V \\ & \sum_{i \in V, i \neq j} x_{ij} = 1, \quad \forall j \in V \\ & u_i - u_j + n x_{ij} \leq n - 1, \quad \forall i \neq j \in \{1, \dots, n-1\} \\ & x_{i,j} \in \{0, 1\} \\ & u_i \in \mathbb{R}^+ \end{array}$$

La nueva restricción añadida evita que haya ciclos que no contengan al nodo origen. Es decir, si x es solución factible del PVC, entonces cualquier ciclo de x contiene al nodo 0.

Proof. Para verlo, supongamos que no es así, es decir, que existe un ciclo alrededor de $\{1, \dots, k\}$, entonces

$$\begin{array}{rcl} u_1 - u_2 + n & \leq & n - 1 \\ u_2 - u_3 + n & \leq & n - 1 \\ \dots & & \dots \\ u_k - u_1 + n & \leq & n - 1 \end{array}$$

sumando todo esto, queda

$$kn \leq k(n-1) \#$$

lo cual es imposible. □

Además, se verifica que, si x es un circuito hamiltoniano, $u_0 = 0$ y $u_i = t \iff i$ es el t -ésimo vértice visitado. Entonces

- si $x_{ij} = 0 \implies u_i - u_j \leq n - 1$
- si $x_{ij} = 1 \implies u_i - u_j + n \leq n - 1$

Proof. La primera afirmación es obvia, pues $1 \leq u_i, u_j \leq n - 1$. Respecto a la segunda, se verificara si, y solo si

$$u_i - u_j \leq -1 \iff u_j \geq u_i + 1$$

pero, como $x_{ij} = 1$ y u_i indica el orden de visitas, entonces $u_j = u_i + 1 \checkmark$ □

Ahora vamos a ver cómo modelar algunas situaciones genéricas y habituales:

- **Una variable binaria cuyo valor depende de la relación entre otras dos variables enteras**

Supongamos que tenemos $x_1, x_2 \in \mathbb{Z}^+, y \in \{0, 1\}$ de tal forma que $y = 1 \iff x_1 > x_2$. Esto puede modelarse como

$$\begin{cases} x_1 - x_2 - My \leq 0 & (1) \\ x_2 - x_1 + (1 + M)y \leq M & (2) \end{cases}$$

donde M es un número suficientemente grande.

Proof. $[\implies] y = 1 \xrightarrow{(2)} x_2 - x_1 + 1 \leq 0 \iff x_1 \geq x_2 + 1 \xrightarrow{x_1, x_2 \in \mathbb{Z}} x_1 > x_2$

$[\impliedby] x_1 > x_2 \xrightarrow{(1)} y = 1$

□

- **Una variable es mayor que el mínimo de otras dos**

Si tenemos $z \geq \min\{x_1, x_2\}$, sabemos que esto se verifica si, y solo si, se cumple, al menos, una de $z \geq x_1$ ó $z \geq x_2$.

Puede modelarse como

$$\begin{cases} x_1 - z \leq My \\ x_2 - z \leq M(1 - y) \end{cases}$$

con $y \in \{0, 1\}$

- **Una variable es menor que el máximo de otras dos**

$z \leq \max\{x_1, x_2\} \iff z \leq x_1$ ó $z \leq x_2$. Se puede modelar

$$\begin{cases} z - x_1 \leq My \\ z - x_2 \leq M(1 - y) \end{cases}$$

con $y \in \{0, 1\}$

- **Mínimo de dos variables binarias**

Ahora es $x_1, x_2 \in \{0, 1\}$ e $y = \min\{x_1, x_2\}$. Podemos hacerlo como

$$\begin{cases} y \leq x_1 \\ y \leq x_2 \\ y \geq x_1 + x_2 - 1 \end{cases}$$

- **Máximo de dos variables binarias**

Ahora es $x_1, x_2 \in \{0, 1\}$ e $y = \max\{x_1, x_2\}$. Podemos hacerlo como

$$\begin{cases} y \geq x_1 \\ y \geq x_2 \\ y \leq x_1 + x_2 \end{cases}$$

- **Que se verifique al menos 1 entre 2 restricciones**

Tenemos 2 restricciones de las que queremos que se verifique al menos una. Podemos expresarlas como

$$g_1(x_1, \dots, x_n) \leq 0$$

$$g_2(x_1, \dots, x_n) \leq 0$$

Para que se verifique al menos una de ellas, añadimos una nueva variable $y \in \{0, 1\}$ y añadimos las restricciones

$$g_1(x_1, \dots, x_n) \leq My$$

$$g_2(x_1, \dots, x_n) \leq M(1 - y)$$

donde M es un número positivo suficientemente grande.

- **Que se verifiquen al menos k restricciones entre m posibles**

Ahora añadimos $y_i \in \{0, 1\}$, $i = 1, \dots, m$ variables binarias, así como las restricciones

$$g_i(x_1, \dots, x_n) \leq M(1 - y_i), \quad 1 \leq i \leq m$$

$$\sum_{i=1}^m y_i \geq k$$

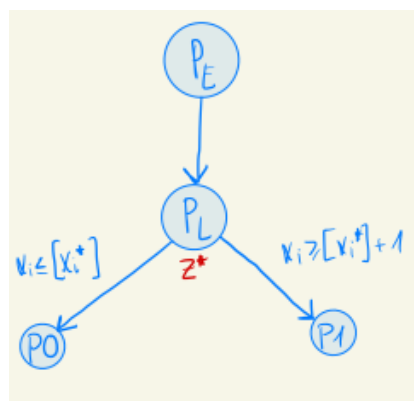
7 Resolución de problemas de programación lineal entera

7.1 Método de ramificación y acotación

Supongamos que queremos resolver un problema entero (P_E). El algoritmo de ramificación y acotación proporciona un método de resolución.

Lo que hacemos es relajar el problema y obtener (P_L), lo solucionamos (simplex), obteniendo una solución óptima z_L^* . Si esta solución, por casualidad, es también solución de (P_E), habremos terminado. Pero, por desgracia, esto ocurrirá en raras ocasiones.

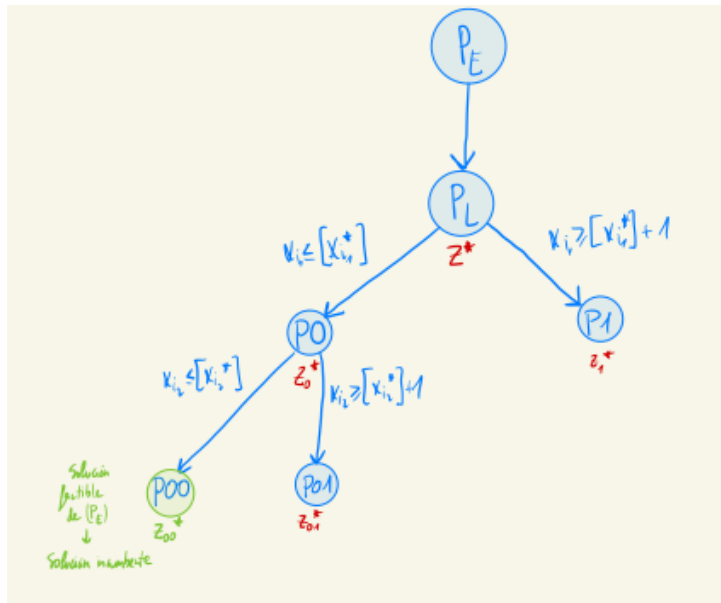
No obstante, podemos utilizar esta solución como punto de partida hacia una solución de nuestro problema. Elegimos alguna de las variables de decisión que deba ser entera, y generamos dos subproblemas, los correspondientes a añadir las restricciones $x_i \leq [x_i^*]$ y $x_i \geq [x_i^*] + 1$, que nos dan los subproblemas (P_0) y (P_1), respectivamente.



Resolvemos estos nuevos subproblemas y obtenemos nuevos valores z_0^*, z_1^* , que serán como mucho tan buenas como z_L^* . Y, de nuevo, puede que sean soluciones de (P_E), o no.

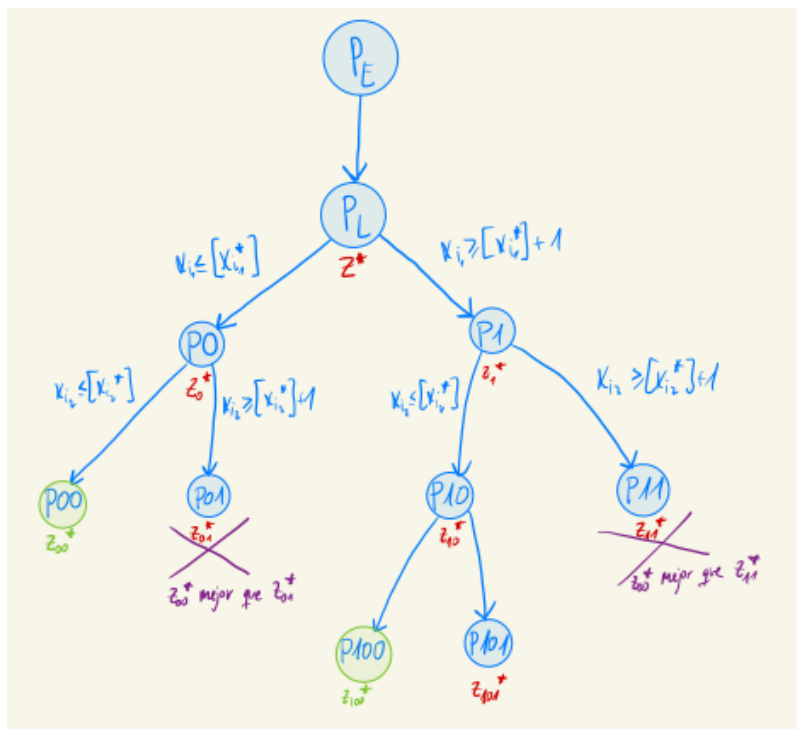
Una opción es reiterar este proceso hasta ramificar todas las variables, obteniendo todas las posibles soluciones de (P_E) y, entre todas ellas, elegir la mejor.

Sin embargo, podemos usar la naturaleza de los problemas de optimización lineal para hacer más eficiente el proceso de búsqueda de la solución óptima. Cuando vamos ramificando, la primera solución factible de (P_E) que encontremos se va a denominar **solución incumbente**, que pasa a ser candidata a solución óptima de nuestro problema original. Al valor óptimo de esta solución se le denomina **valor incumbente**.

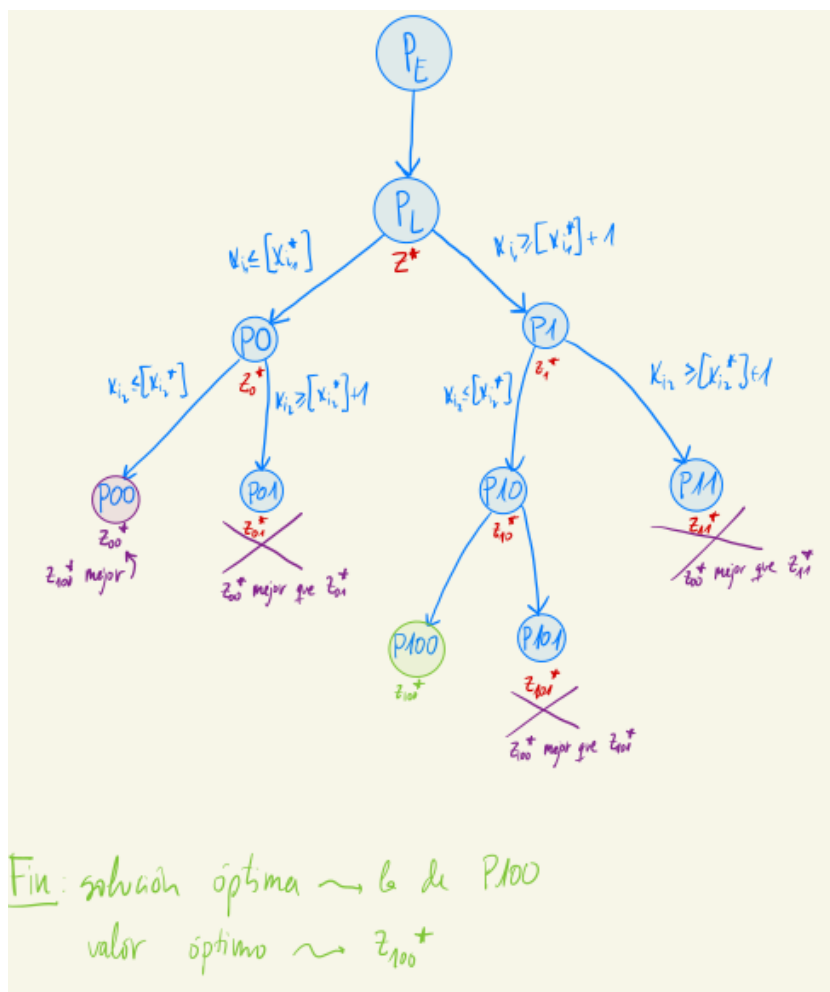


Una interesante observación es que, dado que las soluciones de los subproblemas son como mucho tan buenas como la del problema que las genera, entonces, si el valor óptimo en un nodo es peor que el valor incumbente, no es necesario ramificar ese nodo: no puede generar mejores soluciones.

Por tanto, cuando obtenemos una solución incumbente, podemos descartar (podar, acotar) ramas cuyas soluciones podemos asegurar que serán peores que la actual, que actualizaremos si en algún momento encontramos una mejor solución.



Así, terminaremos cuando no queden ramas por explorar ni subproblemas por resolver.



7.1.1 ¿Cómo seleccionamos el nodo activo a resolver?

No hay una forma sistemática y óptima de hacerlo, pero hay algunas reglas que son, generalmente, de utilidad:

- **Reglas a priori:**

- Búsqueda en profundidad + backtracking: consiste en que, una vez seleccionado un nodo, lo exploramos hasta llegar a las soluciones de (P_E), o sea, a los nodos hoja. Cuando llegamos a estos, tiramos hacia atrás, visitando los vecinos del nodo. Si pueden mejorar el valor previo, los exploramos, si no, pasamos al siguiente vecino. Si no quedan vecinos, subimos al padre, etc. Hasta que no queden ramas por explorar ni subproblemas por resolver.

* La experiencia apunta a que se encuentran más rápidamente soluciones factibles

- Búsqueda en anchura: se exploran todos los vecinos antes de ir a los hijos

- **Reglas adaptativas:** por ejemplo, elegir el nodo con mejor valor óptimo

7.1.2 ¿Cómo seleccionamos la variable a ramificar?

La experiencia demuestra que la elección adecuada de la variable de ramificación puede acelerar la resolución del problema. Sin embargo, como antes, no existen reglas robustas que indiquen cómo seleccionar la variable de ramificación. Algunas posibilidades son:

- Elegir aquella con mayor valor fraccionario
- Elegir aquella cuyo valor fraccionario queda más cerca de 0.5
- Elegir las variables más importantes (por ejemplo, las que más influyen en la función objetivo)

7.2 Método de los hiperplanos de corte

Definition 7.1. Una **desigualdad válida** o **hiperplano de corte** es una desigualdad que cumplen las soluciones factibles del problema lineal entero.

Las desigualdades válidas pueden utilizarse para:

- Obtener mejores cotas en los nodos del árbol de ramificación
- Resolver el problema de programación lineal entera (**método de hiperplanos de corte de Gomory**)

7.2.1 Hiperplano de corte entero

Estamos ante el problema

$$\begin{aligned} (P_E) \max \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

donde suponemos que A y b son enteros.

Sea x^* la solución óptima del problema lineal relajado y supongamos que existe un índice k tal que $x_k \notin \mathbb{Z}$ (si hay varios podemos tomar aquél con parte fraccionaria más cercana a 0.5).

x_k será una variable básica cuya representación en la base óptima B tendrá la forma

$$x_k + \sum_{j \notin B} y_{kj} x_j = x_k^*$$

La desigualdad

$$-\sum_{j \notin B} f_{kj} x_j \leq -f_k$$

, donde f_{kj} es la parte fraccional de y_{kj} y f_k es la parte fraccional de x_k^* , es una desigualdad válida de (P_E) que no satisface la solución óptima del problema lineal relajado.

Proof. Veamos esta última afirmación.

$$x_k + \sum_{j \notin B} y_{kj} x_j = x_k^*$$

puede reescribirse como

$$x_k + \sum_{j \notin B} ([y_{kj}] + f_{kj}) x_j = [x_k^*] + f_k$$

agrupando

$$x_k + \sum_{j \notin B} [y_{kj}] x_j - [x_k^*] = f_k - \sum_{j \notin B} f_{kj} x_j \leq f_k < 1$$

Ahora bien, si consideramos soluciones enteras, entonces los x_j y x_k son enteros, por lo que la parte izquierda es un entero menor que uno, o lo que es lo mismo, la parte izquierda es menor o igual que cero, por tanto

$$0 \geq f_k - \sum_{j \notin B} f_{kj} x_j \implies - \sum_{j \notin B} f_{kj} x_j \leq -f_k$$

Y que la solución óptima del problema lineal relajado no lo satisface es obvio, pues en esa solución los x_j son 0 (son no básicos), por lo que quedaría

$$0 \leq -f_k$$

y f_k es la parte entera de x_k , que no es entero, luego $-f_k < 0$, y es una contradicción. \square

7.2.2 Hiperplano de corte para un problema entero mixto

Ahora queremos resolver

$$\begin{aligned} (P_E) \min \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \\ & x_i \in \mathbb{Z}, \forall i \in I \end{aligned}$$

y suponemos que A y b son enteros.

Sea x^* la solución óptima del problema lineal relajado y supongamos que existe un índice $k \in I$ (de entre las variables enteras) tal que $x_k \notin \mathbb{Z}$ (si hay varios tomamos el que tenga parte fraccionaria más cercana a 0.5). Entonces x_k será una variable básica cuya representación en la base óptima B será

$$x_k + \sum_{j \notin B} y_{kj} x_j = x_k^*$$

Y la siguiente desigualdad es una desigualdad básica de (P_E) que no satisface la solución óptima del problema lineal relajado

$$\sum_{j \in N^-} \frac{f_k}{1 - f_k} y_{kj} x_j - \sum_{j \in N^+} y_{kj} x_j \leq -f_k$$

donde f_{kj} es la parte fraccional de y_{kj} , f_k es la parte fraccional de x_k^* , $N^- = \{j \notin B : y_{kj} < 0\}$ y $N^+ = \{j \notin B : y_{kj} > 0\}$.

Proof. Veamos esta última afirmación.

$$x_k + \sum_{j \notin B} y_{kj} x_j = x_k^* \iff x_k + \sum_{j \notin B} y_{kj} x_j = [x_k^*] + f_k \iff x_k - [x_k^*] = - \sum_{j \in N} y_{kj} x_j + f_k$$

Y nos encontramos con dos casos:

- $x_k \leq [x_k^*]$, entonces $x_k - [x_k^*] = - \sum_{j \in N} y_{kj} x_j + f_k \leq 0 \implies - \sum_{j \in N} y_{kj} x_j \leq -f_k \implies - \sum_{j \in N^+} y_{kj} x_j - \sum_{j \in N^-} y_{kj} x_j \leq -f_k$, y claramente, el segundo término de la parte izquierda es negativo, luego restamos algo negativo, y debe ser

$$- \sum_{j \in N^+} y_{kj} x_j \leq -f_k \quad (1)$$

- $x_k \geq [x_k^*] + 1$, en este caso $-\sum_{j \in N} y_{kj}x_j + f_k \geq 1 \implies -\sum_{j \in N} y_{kj}x_j \geq 1 - f_k \implies -\sum_{j \in N^+} y_{kj}x_j - \sum_{j \in N^-} y_{kj}x_j \geq 1 - f_k$, y el primer término de la parte izquierda es positivo, luego restamos algo positivo, entonces

$$-\sum_{j \in N^-} y_{kj}x_j \geq 1 - f_k \implies \sum_{j \in N^-} y_{kj}x_j \leq f_k - 1$$

multiplicando por $\frac{f_k}{1-f_k} > 0$ queda

$$\sum_{j \in N^-} \frac{f_k}{1-f_k} y_{kj}x_j \leq -f_k \quad (2)$$

Entonces, para ver que se verifica la desigualdad

$$\sum_{j \in N^-} \frac{f_k}{1-f_k} y_{kj}x_j - \sum_{j \in N^+} y_{kj}x_j \leq -f_k$$

observamos que:

- Si $x_k \leq [x_k^*]$, entonces se verifica (1), pero, además, en ese caso el primer término de la parte izquierda de la desigualdad es no positivo, por lo que se verifica la desigualdad
- Si $x_k \geq [x_k^*] + 1$, entonces se verifica (2), y, además, el segundo término de la parte izquierda es no negativo, y lo estamos restando, por lo que se verifica la desigualdad

□

7.2.3 Desigualdades de Chvátal-Gomory

Consideremos un problema de programación entera

$$\begin{aligned} (P_E) \max \quad & c^t x \\ \text{s.a.} \quad & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

donde A es una matriz $m \times n$.

Entonces, para cada $u \in \mathbb{R}_+^m$, la desigualdad

$$\sum_{j=1}^n \left[\sum_{i=1}^m u_i a_{ij} \right] x_j \leq \left[\sum_{i=1}^m u_i b_i \right]$$

es una desigualdad válida del problema (P_E) , y este conjunto de desigualdades son las **desigualdades de Chvátal-Gomory**.

Proposition 7.2. *Cualquier desigualdad válida del conjunto $\{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ se puede obtener aplicando el procedimiento de Chvátal-Gomory (es decir, añadiendo hiperplanos de corte mediante uso de desigualdades de Chvátal-Gomory) un número finito de veces.*