

Grafos y Optimización Discreta

Colaboración estelar entre Paco Mora y Chito Belmonte

9 de noviembre de 2021

Índice general

I	Grafos y optimización discreta	3
1	Conceptos básicos en Teoría de Grafos	5
1.1	Primeros conceptos	5
2	Conexión	11
2.1	Algoritmo basado en las potencias de la matriz de adyacencia	11
2.2	Algoritmo basado en sumas lógicas (Tarjan)	11
3	Arboles	14
3.1	Algoritmo de Kruskal	17
3.2	Algoritmo de Prim	17
4	Camino más corto, recorridos por aristas y vértices	19
4.1	Algoritmo de Dijkstra	21
4.2	Algoritmo de Fleury	21
4.3	Algoritmo de Hierholzer	21
4.4	Algoritmo de Dantzig	22
5	Grafos planos. Coloración de grafos.	25
5.1	Algoritmo austero para colorear (no es óptimo)	26
6	Modelo de Programación Entera (Optimización Discreta)	30
6.1	Problemas de optimización sobre grafos	36
7	Ramificación y acotación. Hiperplanos de corte	39
7.1	Algoritmo de ramificación y acotación	39
7.2	Algoritmo de Ramificación y Acotación (Branch-and-bound)	40
7.3	Hiperplanos de corte	40
7.4	Hiperplano de corte todo entero	40
7.5	Hiperplano de corte mixto	41
8	Ejercicios	43

ASIGNATURA

Grafos y optimización discreta

IMPORTANTE

Estos apuntes son totalmente complementarios, podéis usarlos para estudiar la asignatura bajo vuestro propio criterio. Adicionalmente, la imposibilidad de tomar apuntes en LaTeX en directo nos ha obligado a utilizar un software externo para pasar los PDFs a LaTeX. Como el programa es inestable, es probable que haya erratas.

Agradecemos el contacto a la dirección *eduardo.belmonteg@um.es*

Conceptos básicos en Teoría de Grafos

1.1 - Primeros conceptos

Definición 1.1.1. Grafo

Un **grafo** (simple no dirigido) es un par (V, E) formado por un conjunto finito $V \neq \emptyset$, a cuyos elementos denominaremos **vértices** o **nodos** y E , un conjunto de pares (no ordenados y formados por distintos vértices) de elementos de V a los que llamaremos **aristas**.

Definición 1.1.2. Extremos

A los nodos v_1 y v_2 que forman una arista $e = (v_1, v_2)$ se les llama **extremos** de e .

Un grafo se puede representar mediante un diagrama en el que cada vértice es asociado a un objeto (típicamente un círculo o una circunferencia) y se traza una línea entre los dos vértices que componen una arista.

Ejemplo 1. Dado un tablero de 4×4 escaques, vamos a construir un grafo que represente los pares de casillas en las que dos caballos de ajedrez se amenazan.

Por el movimiento de los caballos de ajedrez, el grafo quedaría de esta forma:

Definición 1.1.3. Adyacencia, orden, tamaño, incidencia...

A dos nodos v_1 y v_2 que forman parte de una misma arista se les llama **adyacentes** (a veces vecinos)

Se llama **orden** de un grafo $|V|$ (se suele denotar por **n**)

Se llama **tamaño** de un grafo $|E|$ (se suele denotar por **m**)

Un grafo con $|V| = 1$ (y consecuentemente, $|E| = 0$) se llama **trivial**

Se dice que una arista es **incidente** con un vértice v cuando v es uno de sus extremos.

Un vértice de grado 0 se llama **aislado**

Un vértice con grado par (impar) se llama **vértice par (impar)**

Proposición 1.1.1. *El número de vértices de grado impar de cualquier grafo ha de ser par*

Demostración

Sea $G = (V, E)$ un grafo. Sumemos los grados de cada una de las aristas:

$$\sum_{v \in V} g(v) = 2|E| = 2m \text{ (par)}$$

La suma de los grados siempre par, ya que cada arista incrementa en 2 esta suma, la cual se puede descomponer según la paridad. Veamos:

$$\sum_{v \in V, g(v) \bmod 2 = 0} g(v) + \sum_{v \in V, g(v) \bmod 2 \neq 0} g(v)$$

La suma de los grados de nodos impares es diferencia de números pares, esto evidencia que se trata de un número par. Sin embargo, busquemos demostrar que el número de sumandos es par.

Como todos los sumandos son impares, si hubieran impares sumandos, el resultado sería impar. Por lo tanto, el número de sumandos es obligatoriamente par.

□

Definición 1.1.4. Grafo regular

*Si todos los vértices de un grafo tienen el mismo grado, el grafo se llama **regular***

*Si el grado de los vértices es k , el grafo es **k -regular**.*

Definición 1.1.5. Subgrafo

*Dado un grafo $G = (V, E)$, se dice que otro grafo $G' = (V', E')$ es **subgrafo** de G si $V' \subseteq V$ y $E' \subseteq E$.*

*Dado un grafo $G = (V, E)$ y dado $B \subseteq V$, se llama **subgrafo de G inducido por B** a $G_B = (B, E_B)$, con*

$$E_B = \{(i, j) \in E : i \in B, j \in B\}$$

*Dado un subgrafo $G = (V, E)$, se llama **subgrafo parcial** a $G' = (V, E')$ con $E' \subseteq E$.*

Definición 1.1.6. Complementario

*Se llama **grafo complementario** de G a $\overline{G} = (V, \overline{E})$, con $\overline{E} = \{(i, j) : i \neq j, (i, j) \notin E\}$.*

Definición 1.1.7. Isomorfía y derivados

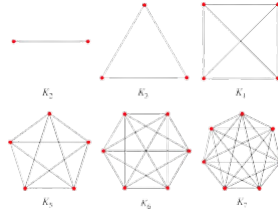
Dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ se dice que son **isomorfos** (y en tal caso escribiremos $G_1 \simeq G_2$) si existe una aplicación biyectiva $f : V_1 \rightarrow V_2$ tal que $\forall u, v \in V_1$

$$(u, v) \in E_1 \iff (f(u), f(v)) \in E_2$$

Tal aplicación recibe el nombre de **isomorfismo**. Si $G_1 = G_2$, llamaremos a f **automorfismo**.

Se llama grafo **completo** a $G = (V, E)$ tal que $v_1, v_2 \in V, v_1 \neq v_2 \implies (v_1, v_2) \in E$.

Al grafo completo K_3 se le llama **triángulo**.



Una colección de grafos que es cerrada bajo isomorfismos se llama una **propiedad del grafo**.

Una aplicación

$$g : \text{Conjunto de grafos} \rightarrow \mathbb{R}$$

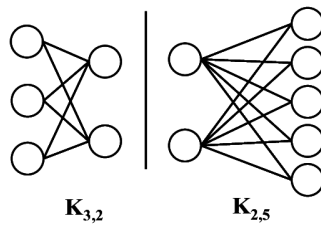
que asigna los mismos valores a dos grafos isomorfos se llama un **invariante del grafo**.

Definición 1.1.8. Grafos bipartitos

Un grafo $G = (V, E)$ es **bipartito** si $V = V_1 \cup V_2$, $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$. y

$$\forall (v_1, v_2) \in E \implies \text{se cumple una de } \begin{cases} v_1 \in V_1, v_2 \in V_2 \\ v_1 \in V_2, v_2 \in V_1 \end{cases}$$

Un **grafo completo bipartito** es un grafo bipartito con $E = V_1 \times V_2$, es decir, que contiene a todas las aristas posibles con un extremo en V_1 y otro en V_2 . Si $|V_1| = r$ y $|V_2| = s$, al correspondiente grafo bipartito se le llama $K_{r,s}$.



Definición 1.1.9. Paseo

Llamamos **paseo** (o *tour*) en un grafo $G = (V, E)$ a una sucesión alternante de aristas de E y vértices de V , de la forma

$$(v_1, (v_1, v_2), v_2, (v_2, v_3), v_3, \dots, v_{k-1}, (v_{k-1}, v_k), v_k)$$

que comienza en un vértice y termina en otro, tal que cada arista incide en el nodo anterior y posterior de la sucesión.

Se dice que el paseo **conecta** (o *une*) el primer y el último vértice, v_1 y v_k .

Si $v_1 = v_k$, se dice que el paseo es **cerrado**.

Definición 1.1.10. Cadena

Una **cadena** (o **camino**), que a veces se le llama *cadena elemental*, es un grafo $P = (V, E)$ definido por $V = \{v_1, \dots, v_n\}$ y $E = \{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)\}$, es decir, una sucesión alternante de aristas y vértices distintos que comienza en un vértice y termina en otro, tal que cada arista incide en el nodo anterior y posterior de la sucesión.

Se dice que p conecta (o *une*) v_1 con v_n . Al camino con m aristas se le suele denominar P_m .

Llamamos **longitud de la cadena** al número de aristas que la componen. Un subgrafo de G que sea cadena (camino) se denominará *cadena (camino)* en el grafo G .

Definición 1.1.11. Ciclo

Un **ciclo** (o *ciclo elemental*) es un grafo $C = (V, E)$ definido por $V = \{v_1, \dots, v_n\}$ ($n \geq 3$) y $E = \{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1)\}$, es decir, una cadena más una arista entre los vértices unidos por la cadena. Al ciclo con n nodos (y aristas) se le suele denominar C_n .

Llamamos **longitud del ciclo** al número de aristas que lo componen.

Un subgrafo de G que sea un ciclo se denominará *ciclo* en el grafo G .

Definición 1.1.12. Cociclo

Dado un grafo $G = (V, E)$ y un conjunto $B \subseteq V$, se llama **cociclo** asociado a B al conjunto de aristas que inciden en B y en $V \setminus B$

$$\omega(B) = \{(v_1, v_2) \in E : v_1 \in B, v_2 \notin B \text{ o bien } v_2 \in B, v_1 \notin B\}$$

Habitualmente se usa la notación $\omega(v)$ para referirse a $\omega(\{v\})$

Definición 1.1.13. Matriz de adyacencia

Dado $G = (V, E)$ con $V = \{v_1, \dots, v_n\}$, la **matriz de adyacencia** $M_{n \times n} = m_{i,j}$ con

$$m_{ij} = \begin{cases} 1 & \text{si } v_i \text{ es extremo de } e_j \\ 0 & \text{en otro caso.} \end{cases}$$

Definición 1.1.14. Matriz de incidencia

Dado un grafo $G = (V, E)$ con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$, la **matriz de incidencia**

(vértice-arista) es $M_{n \times m} = (m_{ij})$ con

$$m_{ij} = \begin{cases} 1 & \text{si } v_i \text{ es extremo de } e_j \\ 0 & \text{en otro caso} \end{cases}$$

También se puede representar un grafo mediante una matriz de enteros, de tamaño $m \times 2$, cuyas filas contienen los extremos inicial y final de las aristas del grafo.

Proposición 1.1.2. $G = (V, E)$ es un grafo bipartito si y sólo si no contiene ciclos de longitud impar.

Demostración

\Rightarrow Suponiendo G bipartito

Sea C un ciclo en G , tomamos $v \in G$ (sin pérdida de generalidad) $v_1 \in V_1$ y v_2 vecino de v_1 . Si $v_1 \in N(v_1) \Rightarrow v_2 \in V_2$, y vamos buscando de manera iterada vecinos.

En algún momento alcanzamos V_k (el último nodo), lo cual nos obliga a que $v_k \in V_2$ y eso sólo puede ocurrir si $k = 2$ (no entiendo muy bien esta dem sinceramente)

\Leftarrow Suponiendo que no contiene ciclos de longitud par

Elegimos un nodo cualquiera $v \in V$ y lo llamamos '+'. A todos los nodos que sean vecinos de v , los llamamos '-', y nos olvidamos de v . Sucesivamente iteramos con este método, cambiando de signo.

Al final, o bien se nos acaban los nodos o bien un nodo recibe el signo más y menos simultáneamente. Si alguno tiene longitud impar, entonces nos encontraremos con este caso. Los signos positivos serían un grupo del bipartito y los negativos los otros, al haber un elemento que exista en ambos, se tiene una contradicción.

* INCISO, ESTA IMPLICACIÓN SE HA COMPLICADO MUCHO, ESTA ES UNA IDEA PERO ESTÁ INCOMPLETA

□

Definición 1.1.15. Totalmente unimodular

Una matriz es **totalmente unimodular (TU)** si todos sus menores valen 0, 1 o -1.

Proposición 1.1.3. Un grafo es bipartito si y sólo si sus matrices de incidencia son totalmente unimodular

Demostración

\Rightarrow

Vamos a demostrar que no contienen ciclos impares y, por lo tanto, será bipartito. Llamemos M a una matriz de incidencia genérica (el resto se obtienen permutando filas y columnas, con lo que no perdemos la generalidad).

Supongamos que M contienen un ciclo impar. La submatriz obtenida de los nodos y aristas del nodo es

$$\begin{bmatrix} 1 & 0 & & 0 & 1 \\ 1 & 1 & \vdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & & \cdots & & \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & & 1 & 1 \end{bmatrix}$$

□

Definición 1.1.16. Entorno

Dado $G = (V, E)$, llamamos **entorno (o vecindario)** de un nodo v al conjunto $N(v) := \{u \in V : (u, v) \in E\}$. Decimos que un vértice v **incide** en una arista e si v es uno de los extremos de e .

Conexión

Definición 2.0.1. Unión de grafos

Se define la unión de los grafos (V_1, E_1) y (V_2, E_2) como el grafo $(V_1 \cup V_2, E_1 \cup E_2)$.

Definición 2.0.2. Conexión

Un grafo es **conexo** si cualquier par de vértices (distintos) están conectados por una cadena. Un grafo que no es conexo suele llamarse **disconexo** (o desconectado).

Diremos que un grafo trivial es también conexo.

Definimos en V la relación siguiente:

$$\forall v_1, v_2 \in V, \quad v_1 \mathcal{C} v_2 \Leftrightarrow \begin{cases} v_1 = v_2 \\ \text{existe una cadena en } G \text{ que conecta } v_1 \text{ con } v_2 \end{cases}$$

\mathcal{C} es una **relación de equivalencia**, de forma que V / \mathcal{C} constituye una partición de V , cada una de cuyas clases de equivalencia se denomina **componente conexa**.

Proposición 2.0.1. Un grafo conexo $G = (V, E)$ verifica $m \geq n - 1$.

2.1 - Algoritmo basado en las potencias de la matriz de adyacencia

Si se calcula $C = \sum_{k=1}^{n-1} M^k$, donde M es la matriz de adyacencia de G , la entrada c_{ij} será positiva si y sólo si v_i y v_j pertenecen a la misma componente conexa. La existencia de ceros en la matriz C indicaría que G no es conexo.

2.2 - Algoritmo basado en sumas lógicas (Tarjan)

Se define la suma lógica \oplus de bits de la forma siguiente:

\oplus	0	1
0	0	1
1	1	1

Paso 1

Inicialización: $S := M, i := 1, c := 1$.

Paso 2

$C(c) := \{v_i\}$

Paso 3

Si $s_{ij} = 0 \quad \forall j > i$, ir al paso 4.

En otro caso, encontrar el primer $j > i$ tal que $s_{ij} = 1$, sustituir la fila i y la columna i de S por la suma lógica de las filas i y j , marcar la fila j y la columna j de S y sustituir todas sus entradas por 0, hacer $C(c) := C(c) \cup \{v_j\}$ y volver al Paso 3 .

Paso 4

Hacer $c := c + 1$, encontrar $k > i$ que no haya sido marcada, hacer $i := k$ y volver al paso 2.

PARAR si tal k no existe.

Las componentes conexas del grafo vienen dadas por los conjuntos $C(q), q = 1, \dots, c - 1$, que se calculan en el algoritmo.

Proposición 2.2.1. *Los vértices de un grafo conexo pueden ser numerados de forma que los subgrafos $G_{\{v_1, \dots, v_i\}}, i = 1, \dots, n$, sean todos grafos conexos.*

Definición 2.2.1. Separador

Dado $G = (V, E)$ conexo y un subconjunto de nodos $B \subseteq V$, si el subgrafo inducido $G_{V \setminus B}$ es desconexo, B se llama un **separador** de G .

Definición 2.2.2. Vértice de corte y puente

Si un conjunto de un solo vértice $B = \{v\}$ es separador, v se llama **vértice de corte**.

Dado $G = (V, E)$ conexo, si $(V, E \setminus \{e\})$ es desconexo, la arista e se denomina **puente**.

Definición 2.2.3. k -conexidad

Un grafo $G = (V, E)$ es **k -conexo** con $k \in \{1, \dots, n - 1\}$, si todos los subgrafos inducidos por conjuntos de nodos B con $|V \setminus B| < k$ son conexos.

Definición 2.2.4. Conectividad

Se llama **conectividad** de G y se denota $\kappa(G)$ al mayor entero k tal que G es k -conexo. Un grafo desconexo tendrá conectividad 0.

Proposición 2.2.2. *Un grafo G es 2-conexo si y sólo si puede construir una sucesión de grafos $G_0 = C_n, G_1, \dots, G_a = G$ en la cual $G_i = (V_i, E_i)$ es obtenido de $G_{i-1} = (V_{i-1}, E_{i-1})$ mediante la unión de un camino $(v_{i_1}, \dots, v_{i_i})$ con $\{v_{i_1}, \dots, v_{i_{k_i}}\} \cap V_{i-1} = \{v_{i_1}, v_{i_{k_i}}\}$.*

Definición 2.2.5. Arista conectividad

El grafo $G = (V, E)$ es **k -arista-conexo** (o k -conexo por aristas) si todos los subgrafos $(V, E \setminus F)$ con $|F| < k$ son conexos.

Se llama **arista-conectividad** (o **conectividad por aristas**) de G conexo y se denota $\lambda(G)$ al mayor entero k tal que G es k -arista-conexo.

Proposición 2.2.3. En todo grafo (conexo) G se verifica $\kappa(G) \leq \lambda(G)$.

Definición 2.2.6. Separar

Dados dos conjuntos de vértices A y B en $G = (V, E)$, otro conjunto de vértices W **separa** A de B si cada camino que une un vértice de A y uno de B contiene un vértice de W .

Teorema 1. De Menger

Sea $G = (V, E)$ un grafo y $A, B \subseteq V$. Entonces el mínimo número de vértices que separan A de B en G es igual al máximo número de caminos disjuntos que conectan A y B .

Proposición 2.2.4. Sean u y v dos vértices diferentes de G . Si $(u, v) \notin E$, el mínimo número de vértices distintos de u y v que separan u de v en G es igual al número máximo de caminos con interior disjunto que conectan u y v en G .

Teorema 2. De Menger II

Un grafo es k -conexo si y sólo si contiene k caminos de interior disjunto entre cualquier par de vértices distintos.

Arboles

Definición 3.0.1. Árboles y propiedades

Diremos que un grafo $G = (V, E)$ es un **árbol** si es **conexo y no contiene ciclos**.

Un **árbol generador** de un grafo $G = (V, E)$ es un subgrafo parcial conexo y sin ciclos. Un **bosque** es un grafo $G = (V, E)$ sin ciclos.

En un árbol, los nodos con grado de incidencia 1 se denominan **hojas**.

Teorema 1. Teorema de 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 una única cadena.
- (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 que se obtiene contiene un único ciclo.

Demostración

$$1 \implies 2$$

Tengo que G es un grafo conexo y con ciclos. Vamos a demostrar que existe una única cadena entre u y v .

Como es conexo, existe una cadena entre u y v . Para comprobar la unicidad, usaremos la ausencia de ciclos. Si existieran dos cadenas diferentes, su yuxtaposición contendría (al menos) un ciclo, con lo cual no pueden existir.

$$2 \implies 3$$

Si se tiene que entre cada par de vértices distintos existe una única cadena, se tiene por definición que G es conexo. Sólo hay que demostrar que $m = n - 1$.

Utilizaremos una proposición que dice que G *conexo* $\implies m \geq n - 1$ y procederemos con inducción sobre el número de nodos.

Para $n = 1$, hay cero aristas. Para $n = 2$, sólo puede haber una arista. Para $n > 2$, asumamos que se cumple para $n - 1$. Eliminamos una arista (cualquiera) del grafo y dado que esa cadena es la única que conecta u con v , ahora cada uno pertenece a componentes conexas con n_1 y n_2 nodos y m_1 y m_2 aristas respectivamente, que siguen cumpliendo la propiedad de hipótesis, luego $m_1 = n_1 - 1$ y $m_2 = n_2 - 1$, con lo que en G , $n = m + n_2 = m_1 + 1 + m_2 + 1 = (m_1 + m_2 + 1) + 1 = m + 1$.

$$3 \implies 4$$

Si existiera un ciclo en G , retirando una arista del ciclo no podría desconectar el grafo. Tendría un grafo conexo con n nodos y $m - 1$ aristas. Como $m = n - 1$, me quedaría que $m - 1 = n - 1 \implies m = n$, que es una contradicción.

$$4 \implies 5$$

Basta demostrar que el grado es conexo, porque como se tiene que $m = n - 1$ y todo grafo conexo cumple $m \geq n - 1$, no podría estar conectado.

Supongamos que G tiene s componentes conexas, $(V_1, E_1), \dots, (V_s, E_s) \rightarrow n_i$ *nodos*, m_i *aristas*.

Cada componente conexa es acíclica y conexa, luego cumple la propiedad 1 del teorema, cumpliendo también 2 y 3, y cumpliendo 3 cumple que $m_i = n_i - 1 \forall i$.

Se tiene que $n = \sum_{i=1}^s n_i = \sum_{i=1}^s (m_i + 1) = s + \sum_{i=1}^s m_i = m + s$

$$5 \implies 6$$

Añadir una arista cierra un ciclo con la cadena que conecta sus extremos. Si cerráramos dos o más ciclos, retirar una de las arista no desconectaría el grafo de partida.

$$6 \implies 1$$

Supongamos que el grado es acíclico y la adición de una arista da un único ciclo. Por hipótesis, no se contienen ciclos, y la condición de que añadir una arista crea un ciclo implica que, entre esos dos nodos, ya había previamente una cadena, con lo que el grafo es conexo.

□

Definición 3.0.2. Raíz y árboles con raíz

En un árbol podemos distinguir un nodo cualquiera que pasará a llamarse (**nodo**) **raíz** r . El par compuesto por árbol y raíz se llamará entonces **árbol con raíz** o enraizado. El resto de nodos con grado de incidencia mayor que uno se llamarán **nodos**.

Definición 3.0.3. Predecesores y comparables

Cada nodo $v \neq r$ del árbol estará conectado a r mediante un único camino (r, v_1, \dots, v_k, v) .

Diremos entonces que los nodos r, v_1, \dots, v_k son **predecesores** de v en el árbol y que v es sucesor de estos nodos. La relación de precedencia $u \leq v \iff u$ precede a v o bien $u = v$ induce un orden

parcial^a en los nodos del árbol enraizado. Si $u \leq v$ o $v \leq u$, diremos que u y v son **comparables**.

^areflexivam antisimétrica y transitiva

Definición 3.0.4. Alturas

Diremos que la raíz del árbol enraizado está a **nivel cero**. Los vecinos de la raíz decimos que están a **nivel uno**. Cuando suben al nivel seis, pillan ulti y te pueden oneshotear.

Los vecinos de los de nivel uno son de nivel dos... etc. Llamaremos **altura** del árbol enraizado al máximo de los niveles de sus nodos.

Proposición 3.0.1. Todo árbol con $n \geq 2$ tiene al menos dos hojas

Demostración

Sea h el número de hojas del árbol G , sabemos que $m = n - 1$

$$h + \sum_{\text{no hojas}} g(v)_{\geq 2} = \sum_{\text{hojas}} g(v) + \sum_{\text{no hojas}} g(v) = \sum_{v \in V} g(v) = 2 * m = 2n - 2$$

$$h = 2n - 2(\geq 2(n - h)) \implies h \geq 2$$

□

Definición 3.0.5. Árbol binario

Un **árbol binario** es aquél cuyo grado tienen 1 o 3 excepto un único nodo que tiene grado 2. Llamaremos **raíz del árbol binario** al nodo de grado 2.

Proposición 3.0.2. El número de vértices de un árbol binario es impar.

Demostración

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

$$\sum_{v \in V} g(v) = h + 2 + 3(n - h - 1)$$

$$2 - n - 2 = 2 + 3n - 2h - 3$$

□

Proposición 3.0.3. La altura de un árbol binario está comprendida entre $\lceil \log_2(n + 1) - 1 \rceil$ y $(n - 1)/2$

Demostración

Es fácil comprobarlo haciendo una representación de los posibles árboles de máxima altura y de mínima altura.

Para hacerlo ha utilizado un trick **bastante importante**,

$$\sum_{i=0}^k 2^i = 2^{k+1} - 1$$

□

Definición 3.0.6. Red

Una **red** es una terna (V, E, \mathcal{L}) formada por un grafo (V, E) y una función $\mathcal{L} : E \rightarrow R$, llamada **función de peso** (*longitud, coste... tiene muchos nombres*).

Definición 3.0.7. Dada una red (V, E, \mathcal{L}) y un árbol generador, $T = (V, E_T)$, se define el **peso del árbol** como

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

La búsqueda del árbol generador de peso mínimo (o árbol generador minimal) en G es la resolución del problema.

Proposición 3.0.4. Considérese una red $G = (V, E, \mathcal{L})$ y un árbol generador de G , $T = (V, E_T)$. Las siguientes afirmaciones son equivalentes:

- T es un árbol generador minimal.
- Dada cualquier arista $a \in E \setminus E_T$ y cualquier otra arista $e \in E_T$ perteneciente al único ciclo del grado $(V E_T \cup \{a\})$ se verifica $\mathcal{L}(a) \leq \mathcal{L}(e)$.
- Para cualquier arista $e \in E_T$, $\mathcal{L}(e) \leq \mathcal{L}(a) \quad \forall a \in \omega(V_1^e)$, donde este último es el cociclo en el grafo G asociado a una componente conexa cualquiera de $(V, E_T \setminus \{e\})$

Demostración

$$1 \implies 2$$

$$2 \implies 3$$

□

3.1 - Algoritmo de Kruskal

Paso 1

Ordenar las aristas de E en orden ascendente de su peso:

$$V = \{v_1, \dots, v_n\}, \quad T^* = (V, \emptyset) \quad E := \{e_1, \dots, e_m\} : \mathcal{L}(e_i) \leq \mathcal{L}(e_i + 1) \forall i < m$$

Paso 2

Añadir $n - 1$ aristas a T^* sucesivamente (en el orden de sus pesos) sin que se formen ciclos.

3.2 - Algoritmo de Prim

Paso 1

Elegir un vértice $r \in V$ y hacer $V_1 = \{r\}$, $V_2 = V \setminus \{r\}$.

Paso 2

Añadir al árbol la arista de menor peso de $w(V_1)$, digamos (v_1, v_2) con $v_1 \in V_1$ y $v_2 \in V_2$. Añadir v_2 a V_1 y borrar v_2 de V_2 .

Paso 3

Si $|V_1| = n$ parar. Si no, volver al Paso 2.

Caminos más cortos, recorridos por aristas y vértices

Definición 4.0.1. Longitud de un camino

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

$$\ell(P) := \sum_{a \in P} \ell(a)$$

El problema del camino más corto (CMC) entre v_i y v_j en G es

$$\begin{aligned} &\text{mín } \ell(P) \\ &\text{s.a } P \text{ es un camino que une } v_i \text{ con } v_j \end{aligned}$$

En lo que sigue, para una arista $e = (v_i, v_j)$ notaremos indistintamente $\ell(v_i, v_j)$, ℓ_e , ℓ_{ij} . También supondremos que la red es conexa y que $\ell_e \geq 0 \forall e \in E$.

Proposición 4.0.1. Dado un camino más corto de v_a a v_b , todo subcamino de éste que conecte v_i con v_j es un camino más corto entre v_i y v_j

Demostración

Si hubiera un camino más corto P_1 entre v_i y v_j que el subcamino P_2 entre v_i, v_j , reemplazando P_2 por P_1 obtenemos o bien 1. o 2.:

1. Un camino más corto que el camino más corto con lo que tenemos una contradicción.
2. Un paseo que contiene ciclos, la eliminación de estos ciclos nos deja un camino más corto que el camino más corto, de nuevo una contradicción.

□

Teorema 1. El vector $d = (d_1, d_2, \dots, d_n)$ que contiene longitudes de caminos que conectan el nodo v_s con todos los nodos v_1, v_2, \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 sólo si

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

Demostración

\Leftarrow

Supongamos $d_j > d_i + \ell_{ij}$, entonces, podemos crear un camino más corto a j yuxtaponiendo el camino a i y la arista que une i con j si no se forman ciclos, si se formaran, basta con quitarlo y aún así tendríamos un camino más corto a j . En ambos casos tenemos una contradicción.

\Rightarrow

Sea $j \in V$ cualquiera, sea P un camino cualquiera de s a j , ¿se cumple que $l(p) \geq d_j$? Si $P = (s = i_0, i_1, \dots, i_q = j)$ tenemos que:

$$d_{i_1} - d_{i_0} \leq \ell_{i_0 i_1} \quad d_{i_2} - d_{i_1} \leq \ell_{i_1 i_2} \quad \dots$$

$$\text{Sumando todo obtenemos que } d_j - \underbrace{d_s}_0 = d_{i_q} - d_{i_0} \leq \sum_k \ell_{i_k i_{k+1}} = l(P)$$

□

Teorema 2. Supongamos que el vector $d = (d_1, d_2, \dots, d_n)$ contiene longitudes de caminos que conectan el nodo v_s con todos los nodos v_1, v_2, \dots, v_n de una red, con $d_s = 0$. Sea V' un conjunto de vértices para el cual

- (a) $v_s \notin V'$
- (b) si $v_i \notin V'$, d_i es la longitud de un camino más corto de v_s a v_i ,
- (c) si $v_i \in V'$, $d_i = \min \{d_j + \ell_{ji} : v_j \notin V', v_j \in N(v_i)\}$ 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 .

Demostración

Sea P camino entre s y j , ¿ $l(p) \geq d_j$? De este camino $(v_s, v_a, v_b, \dots, v_j)$ sabemos que:

- $v_s \notin V'$
- $v_j \in V'$

Si P_2 es el subcamino desde s hasta el primer nodo de V' con último coste $(v_{i_1}, v_{i_2}) \in w(V')$, entonces:

$$l(P) \geq l(P_2) = \underbrace{l(P_1)}_{\text{longitud del subcamino que une } s \text{ y } v_{i_j}} + \ell_{i_1 i_2} \geq d_{i_1} + \ell_{i_1 i_2} \geq_{(c)} d_{i_2} \geq d_j$$

□

4.1 - Algoritmo de Dijkstra

Paso 1

$d_s := 0$ $d_i := \ell_{si} \forall v_i \in N(v_s)$ $d_i := \infty$ en otro caso. $p(i) := s \quad \forall v_i \in N(v_s)$, $p(i) := 0$ en otro caso, $V' := V \setminus \{v_s\}$

Paso 2

Buscar un nodo $v_j \in V'$ tal que $d_j = \min \{d_i : v_i \in V'\}$. Hacer $V' := V' \setminus \{v_j\}$.

Paso 3

Para todo $v_i \in N(v_j) \cap V'$: si $d_j + \ell_{ji} < d_i$ entonces:

$$d_i := d_j + \ell_{ji}, \quad p(i) := j$$

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

Definición 4.1.1. Tour euleroiano y grafo euleriano

Dado un grafo $G = (V, E)$, se llama **tour euleroiano (o recorrido euleroiano)** a un paseo cerrado que atraviesa cada arista de E exactamente una vez. Un grafo que admite un recorrido euleroiano se denomina **grafo euleriano**.

Teorema 3. Un grafo conexo es euleriano si y sólo si todos los vértices son pares.

4.2 - Algoritmo de Fleury

Paso 1

Comenzando con una copia del grafo, $G' = (V', E')$, elegir cualquier vértice $i \in V'$.

Paso 2

Si existe una arista $(i, j) \in E'$ que no sea puente en G' , elegirla. En otro caso, elegir cualquier arista (i, j) de E'

Paso 3

Anotar la arista (i, j) en el paseo y eliminarla de E' . Si $E' = \emptyset$, FIN. En otro caso, hacer $i := j$ y volver al paso 2 .

4.3 - Algoritmo de Hierholzer

Paso 1

Comenzando con una copia del grafo, $G' = (V', E')$, elegir cualquier vértice $i \in V'$.

Paso 2

Si se puede elegir una arista (i, j) de E' , anotar la arista (i, j) en el paseo y eliminarla de E' . Si $E' = \emptyset$, FIN. Si $E' \neq \emptyset$ hacer $i := j$ y volver al paso 5

Si no se puede elegir ninguna arista (i, j) de E' , reemplazar i por un vértice ya perteneciente al paseo en el que incida una arista de E' . Sea $(j, i), i, (i, k)$ un tramo de este paseo. Hacer $e = (j, i)$.

Volver al paso 5 pero anotando la arista en el interior del paseo a continuación de e y reemplazando e por la arista anotada.

Definición 4.3.1. *Cartero chino*

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

4.4 - Algoritmo de Dantzig

Paso 1

Hacer $k := 1$ y obtener para cada $v_i, v_j \in V$:

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

Paso 2

Para $i = 1, 2, \dots, k$

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

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

Si $\arg \min_{j=1, \dots, k} \{\ell_{j, k+1} + D(i, j)\} \neq i$,

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

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

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

$$D(k+1, i) := D(i, k+1)$$

Paso 3

Para cada $i, j = 1, \dots, k$ hacer

$$D(i, j) := \min \underbrace{\{D(i, j)\}}_A, \underbrace{\{D(i, k+1) + D(k+1, j)\}}_B;$$

Si el mínimo es B, hacer $P(i, j) := P(k+1, j)$. Hacer $k := k+1$. Si $k < n$, volver al Paso 2

El problema del cartero chino se resuelve mediante el siguiente procedimiento:

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 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 (multi)grafo euleariano así obtenido, encontrar un ciclo euleariano usando el algoritmo de Fleury.

Definición 4.4.1. Hamiltoniano Un ciclo en un grafo G es **hamiltoniano** si contiene a todos los vértices del grafo. Si tal ciclo existe, diremos que el grafo es hamiltoniano.

En un grafo G un **camino hamiltoniano** es aquél que contiene a todos los vértices del grafo.

Dado un grafo G se obtiene el grafo clausura de G , denotado por $[G]$, mediante el proceso iterativo siguiente:

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

Teorema 4. Un grafo G es hamiltoniano si y sólo si su grafo clausura $[G]$ es hamiltoniano.

Demostración

\Rightarrow

Directo.

\Leftarrow

Reducción al absurdo: supongamos que existe un grafo no hamiltoniano cuya clausura sí es hamiltoniana.

Sea $G_0, \dots, G_k = [G]$ con j el primer índice tal que G_j no es hamiltoniano pero G_{j+1} sí lo es. La última arista añadida, digamos que es $(v_1, v_n) \notin E_j$ y $(v_1, v_n) \in E_{j+1}$, tiene que estar en el ciclo hamiltoniano de G_{j+1} . Sea este ciclo v_1, \dots, v_n, v_1 .

Retirando (v_1, v_n) de nuevo, se sigue que en G_j existía un camino hamiltoniano (v_1, v_2, \dots, v_n) .

Sean $Y = \{i \in \{3, \dots, n-1\} : (v_1, v_i) \in E_j\}$ (algunos de los vecinos de v_1) y $X = \{i \in \{3, \dots, n+1\} : (v_n, i) \in E_j\}$ (los nodos a la derecha de algunos de los vectores de v_n). Entonces $|Y| = g_{G_j}(v_1) - 1$ y $|X| = g_{G_j}(v_n) - 1$ y $|X| + |Y| \geq_{\text{por la construcción de } G_{j+1}} n - 2$.

Como $X, Y \subset \{3, 4, \dots, n-1\}$ con cardinal $n-3$ y sus cardinales suman $n-2 \Rightarrow X \cap Y \neq \emptyset \Rightarrow \exists v_k \in X \cap Y \Rightarrow$

$$\Rightarrow \begin{cases} (v_k, v_1) \in E_j \\ (v_{k-1}, v_n) \in E_j \end{cases}$$

Y tenemos que $(v_1, \dots, v_{k-1}, v_n, v_{n-1}, \dots, v_k, v_1)$ es un ciclo hamiltoniano.

□

Definición 4.4.2. Sucesión hamiltoniana

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

Teorema 5. de Chvátal

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 sólo si para cada $i < n/2$ se satisface

$$a_i \leq i \Rightarrow a_{n-i} \geq n - i.$$

Demostración

\Rightarrow

Utilizaremos el contrarrecíproco, si tenemos un $i < \frac{n}{2}$ tal que $a_i \leq 1$ y $a_{n-1} < n - 1$

\Leftarrow

Supongamos que no es hamiltoniano. Sea $G(V, E)$ un grafo con el mayor número posible de aristas que no es hamiltoniano con grafos $g_1 \leq \dots \leq g_n$. Esta grafo satisface la condición

$$g_i \leq i \Rightarrow a_i \leq g_i \leq 1 \Rightarrow a_{n-i} \geq n - 1 \Rightarrow g_{n-i} \geq a_{n-i} \geq n - i$$

De todos los G en estas condiciones elegimos uno con número máximo de aristas. Sean $u, v \in V$ con $(u, v) \notin E$ con máxima suma de $g(u) + g(v)$. Sea u el de menor grado: $g(u) \leq g(v)$. Como G es maximal, al añadir la arista (u, v) pasa a ser hamiltoniano, luego \exists camino hamiltoniano $u \sim v$ en G :

$$u = v_1 - v_2 - \dots - v_n = v$$

Como vimos ayer si u es vecino de v_i , no puede darse que v sea vecino de v_j , $j < i$.

Consideremos el conjunto de vecinos de u ($N(u) \subset \{v_1, \dots, v_{n-1}\}$) y el conjunto de nodos de v_i que preceden inmediatamente en el camino a un vecino de $u \subset \{v_1, \dots, v_{n-1}\}$. La existencia de un nodo en la intersección de estos 2 conjuntos implica la existencia de un ciclo ham. en G , luego son disjuntos.

La unión de estos dos conjuntos está incluida en $\{v_1, \dots, v_{n-1}\}$. Uno de los conjuntos tiene cardinal $g(u)$ y el otro $g(v)$, como la unión es disjunta el cardinal es $g(u) + g(v)$. Por tanto, tenemos que $g(u) + g(v) < n \Rightarrow g(u) < \frac{n}{2}$

Sea $x \in V$ un nodo que precede en el camino a un vecino de u (y que por tanto no es vecino de v). Como tomamos u, v que no fueran vecinos con máximo $g(u) + g(v)$, tenemos que $g(x) \leq g(u)$. En las condiciones de x tengo tantos nodos como $|N(u)| = g(u)$. Sea $i = g(u)$, entonces $i < \frac{n}{2}$, $g(i) \leq i \Rightarrow g_{n-i} \geq n - i$

Luego al menos hay $i + 1$ nodos con grado $\geq n - i$. Como $i = g(u)$ no todos estos nodos pueden ser vecinos de n . Sea $w \notin N(u)$ con hgrado $\geq n - i$.

$$g(w) \geq n - g(u) \Rightarrow \text{Contradicción con la elección de } v$$

□

Definición 4.4.3. Problema del viajante

Dada una red (V, E, ℓ) , a la **determinación del ciclo hamiltoniano de mínima longitud** se le llama **problema del viajante de comercio** sobre esa red.

Grafos planos. Coloración de grafos.

Definición 5.0.1. Coloración

Una **coloración propia** (o simplemente coloración) de un grafo es una aplicación

$$c : V \longrightarrow \{1, 2, \dots, \ell\}$$

que hace corresponder valores diferentes a vértices adyacentes:

$$\forall (u, v) \in E \rightarrow c(u) \neq c(v)$$

El valor $c(v_i)$ es el **color** correspondiente al nodo v_i .

Definición 5.0.2. Número cromático

Llamaremos **número cromático** del grafo G , $\chi(G)$, al **mínimo valor** de ℓ que **permite una coloración** de G , es decir, al mínimo número de colores necesarios para colorear los vértices de forma que los extremos de cada arista tengan colores distintos.

Definición 5.0.3. Crítico

Se dice que un grafo G es **crítico** si para cualquier subgrafo propio H se verifica $\chi(H) < \chi(G)$.

Proposición 5.0.1. Metralleta de resultados *Los siguientes tres resultados se cumplen:*

1. Cada grafo G con $n \geq 2$ contiene un subgrafo crítico H tal que $\chi(G) = \chi(H)$.
2. Si $G = (V, E)$ es crítico, G es conexo.
3. Si $G = (V, E)$ es crítico, $\forall v \in V, g(v) \geq \chi(G) - 1$

Demostración

1.

Si G es crítico, hemos terminado. Si G no es crítico, contiene un subgrafo (propio) G_2 con el mismo número cromático. Si G_2 es crítico, hemos terminado, si no, iteramos el proceso.

Finalmente agotaremos la sucesión hasta un nodo o encontraremos uno crítico.

2

En un grafo no conexo, el número cromático es el máximo de sus componentes conexas. El subgrafo inducido de una de las cc.cc. que de ese máximo tiene por tanto el número cromático de gG .

3.

Si existe $v \in B$ con un número cromático $g(v) \leq \chi(G) - 2$, retirando v de G , dado que G es crítico, $\chi(G_{V \setminus \{v\}}) \leq 1$. Por tanto, existe una coloración c propia de $G_{V \setminus \{v\}}$.

□

5.1 - Algoritmo austero para colorear (no es óptimo)

Paso 0

Inicialización. Ordenar los elementos de $V : v_1, \dots, v_n$.

Paso 1

Asignar $c(v_1) := 1$.

Paso 2

Si $(v_1, v_2) \notin E$, asignar $c(v_2) := 1$; en otro caso, asignar $c(v_2) := 2$.

Paso 3

- Si habíamos coloreado $c(v_1) = c(v_2) = 1$:

Si $(v_1, v_3) \notin E$ y $(v_2, v_3) \notin E$, asignar $c(v_3) := 1$

En otro caso, asignar $c(v_3) := 2$ - Si habíamos coloreado $c(v_1) = 1$ y $c(v_2) = 2$:

Si $(v_1, v_3) \notin E$, asignar $c(v_3) := 1$

En otro caso:

Si $(v_2, v_3) \notin E$, asignar $c(v_3) := 2$

En otro caso, asignar $c(v_3) := 3$.

Y así sucesivamente.

Proposición 5.1.1. Dos resultaditos

El algoritmo de coloreado utiliza a lo sumo $1 + \max_{v \in V} \{g(v)\}$ colores.

En todo grafo G se verifica $\chi(G) \leq 1 + \max_{v \in V} \{g(v)\}$.

Dado un grafo G , consideremos la función

$$P_G : \mathbb{N}^* \longrightarrow \mathbb{N}^*$$

que asigna a cada valor k el número de distintas coloraciones del grafo G que no usen más de k colores.

Está claro que $P_G(0) = P_G(1) = \dots = P_G(\chi(G) - 1) = 0$, mientras que $P_G(\chi(G)), P_G(\chi(G) + 1) \dots \geq 1$. También está claro que $P_G(k) \leq P_G(k + 1) \forall k$.

Definición 5.1.1. Polinomio cromático

Se llama *polinomio cromático* del grafo G al polinomio de grado n resultante de interpolar $(k, P_G(k))$ para $k = 0, \dots, n$

Definición 5.1.2. Identificar

Dado un grafo $G = (V, E)$ con $V = \{v_1, \dots, v_n\}$ y un par de nodos $v_i \neq v_j \in V$, se llama *identificar* v_i y v_j a construir un nuevo grafo $G = (V', E')$ con $V' = V \setminus \{v_i, v_j\} \cup \{v_{n+1}\}$ y $E' = E \setminus (\{(v_i, v_j)\} \cup \omega(\{(v_i, v_j)\})) \cup E''$, donde $E'' = \{(v_k, v_{n+1}) : (v_k, v_i) \in E \text{ o } (v_k, v_j) \in E \text{ ambas}\}$.

Teorema 1. Dado un grafo $G = (V, E)$ y un arista $(u, v) \in E$, se satisface $P_G(k) = P_{G_1}(k) - P_{G_2}(k)$ donde $G_1 = (V, E \setminus \{(u, v)\})$ y G_2 es el grafo obtenido de G identificando los dos vértices u y v .

Definición 5.1.3. Grafo plano

Un **grafo plano** es (V, E) , un par de conjuntos finitos llamados *vértices* y *aristas* que satisfacen

1. $V \subseteq \mathbb{R}^2$
2. Cada arista de E es un arco entre dos vértices distintos.
3. Aristas diferentes tienen extremos diferentes.
4. El interior de una arista no contiene ni vértices ni puntos de otra arista.

Para cada grafo plano G , el conjunto $\mathbb{R}^2 \setminus (V \cup E)$ es abierto. Sus regiones se llaman **caras**. Como G es acotado, una de sus caras será no acotada y se denomina **cara externa**, a diferencia de las caras internas.

Definición 5.1.4. Planar

Un grafo isomorfo a un grafo plano se denomina **planar**.

Cada representación de un grafo planar como grafo plano se llama un **embutido**.

Admitiremos las siguientes propiedades de cualquier embutido de G :

- Dos caras diferentes son disjuntas, y sus contornos tienen intersección sólo en las aristas.
- Hay una única cara exterior.
- El interior de cada ciclo de G contiene al menos una cara interna.

- Un puente pertenece al contorno de una sola cara.
- Una arista que no es puente pertenece exactamente al contorno de dos caras.

Proposición 5.1.2. *Un embutido de un grafo planar G no tiene caras internas si y sólo si G es un bosque.*

Teorema 2. *Sea G un grafo planar conexo. Consideremos un embutido de G con c caras. Entonces*

$$n - m + c = 2$$

Proposición 5.1.3. *Si G es un grafo planar con $n \geq 3$, entonces $m \leq 3n - 6$.*

Si además ningún subgrafo inducido de G es K_3 , entonces $m \leq 2n - 4$.

Proposición 5.1.4. *K_5 y $K_{3,3}$ no son grafos planares.*

Proposición 5.1.5. *Si G es planar, el grado medio de sus nodos es menor que 6 .*

Definición 5.1.5. Maximal

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

Proposición 5.1.6. *Sea F una cara de un embutido del grafo G con al menos cuatro aristas en su contorno. Entonces hay dos vértices no adyacentes entre los extremos del contorno de F .*

Proposición 5.1.7. *Si G es un grafo planar maximal con $n \geq 3$, el contorno de cada cara del embutido plano de G tiene exactamente tres aristas.*

Proposición 5.1.8. *En todo grafo planar maximal con $n \geq 3$ se verifica $m = 3n - 6$.*

Definición 5.1.6. Subdividir

*Dado un grafo $G = (V, E)$ y una arista $(u, v) \in E$, se llama **subdividir** la arista (u, v) a añadir un nuevo nodo v_{n+1} al grafo y reemplazar (u, v) por (u, v_{n+1}) y (v, v_{n+1}) .*

Definición 5.1.7. Subdivisión

*Se llama **subdivisión del grafo** G al grafo obtenido de G subdividiendo sucesivas aristas.*

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

Teorema 4. *Si G es un grafo planar, entonces $\chi(G) \leq 4$.*

(aunque demostraremos el teorema del mapa de los cinco colores)

Modelo de Programación Entera

(Optimización Discreta)

Un problema de programación lineal es aquél que se puede establecer mediante la formulación

$$\begin{cases} \text{mín}_x & c \cdot x \\ \text{s.a} & Ax \leq b \\ & x \in \mathbb{R}_+^n \end{cases}$$

con $c \in \mathbb{R}^n$, $A_{m \times n} = (a_{ij})$, $a_{ij} \in \mathbb{R} \forall i, j$, $b \in \mathbb{R}^m$.

Es decir,

$$\begin{cases} \text{mín} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{s.a} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ & x_1, x_2, \dots, x_n \geq 0 \end{cases}$$

O, equivalentemente,

$$\begin{cases} \text{mín} & \sum_{j=1}^n c_j x_j \\ \text{s.a} & \sum_{j=1}^n a_{ij} x_j \leq b_i \forall i = 1, \dots, m \\ & x_j \geq 0 \quad \forall j = 1, \dots, n \end{cases}$$

Definición 6.0.1. *Infinitas cantidades de texto*

Los valores a_{ij} se denominan **coeficientes** de las restricciones.

Las variables $x = (x_1, \dots, x_n)$ se denominan **variables de decisión**.

Los valores b_i se denominan **lados derechos o términos independientes**.

Las desigualdades $\sum_{j=1}^n a_{ij}x_j \leq b_i$ se denominan **restricciones lineales**.

Las desigualdades $x_j \geq 0$ se denominan **restricciones de no negatividad**.

Los valores c_j se denominan **costes** o simplemente **coeficientes de la función objetivo**.

La función lineal $\sum_{j=1}^n c_j x_j$ se denomina **función objetivo**.

A minimizar la función objetivo se le llama **objetivo**.

El conjunto de soluciones (valores de las variables) que satisfacen todas las restricciones se denomina **región factible**.

El valor mínimo que se busca determinar recibe el nombre de **valor óptimo** del problema. **Puede no existir** si (i) no se puede dar valores a las variables que satisfagan todas las restricciones (la región factible es vacía), en cuyo caso se dice que **el problema es infactible** o (ii) para cualquier $h \in \mathbb{R}$ existe un punto en la región factible x con $cx < h$. En este segundo caso se dice que el problema es **no acotado**.

Definición 6.0.2. Solución óptima

Las soluciones que corresponden al valor óptimo se llaman **soluciones óptimas**.

Téngase en cuenta que el problema puede asimismo ser de maximización, con desigualdades ≥ 0 igualdades, ya que $\max f(x) = -\min(-f(x))$, que $f(x) \geq f_0 \equiv -f(x) \leq -f_0$ y una igualdad puede descomponerse en dos desigualdades.

También pueden incluirse variables sin restricción de signo, sin más que reescribirlas como la diferencia de dos variables no negativas.

Nótese que las restricciones lineales y de no negatividad definen un poliedro (acotado o no) en el que se encuentran las soluciones factibles del problema.

Una restricción del tipo $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$ puede transformarse en una igualdad introduciendo una **nueva variable (de holgura)** x_{n+i} :

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_{n+i} = b_i$$

$$x_{n+i} \geq 0$$

La variable de holgura no aparecerá en ninguna otra restricción ni en la función objetivo

Un problema de programación lineal en forma estándar

$$\begin{cases} \min_x & c \cdot x \\ \text{s. a} & Ax = b \\ & x \in \mathbb{R}_+^n \end{cases}$$

puede resolverse mediante varios métodos, entre los cuales nos interesa destacar el método **símplex dual**. Para ello considerábamos una tabla asociada a una base $B = \{x_1, \dots, x_m\}$ de la forma (salvo permutación de filas y/o columnas y/o variables)

	x_1	x_2	\cdots	x_m	x_{m+1}	x_{m+2}	\cdots	x_n	
x_1	1	0	\cdots	0	$\bar{a}_{1,m+1}$	$\bar{a}_{1,m+2}$	\cdots	\bar{a}_{1n}	b_1
x_2	0	1	\cdots	0	$\bar{a}_{2,m+1}$	$\bar{a}_{2,m+2}$	\cdots	\bar{a}_{2n}	\bar{b}_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
x_m	0	0	\cdots	1	$\bar{a}_{m,m+1}$	$\bar{a}_{m,m+2}$	\cdots	$\bar{a}_{m,n}$	\bar{b}_m
	0	0	\cdots	0	\bar{c}_{m+1}	\bar{c}_{m+2}	\cdots	\bar{c}_n	

con $\bar{c}_j \geq 0 \forall j$ (para detalles, ver apuntes de la asignatura correspondiente).

Si $\bar{b}_i \geq 0 \forall i$, la solución dada por la tabla es óptima:

$$x_i = \bar{b}_i \forall i \in B, \quad x_i = 0 \forall i \notin B$$

En otro caso, se elige una variable de índice ℓ con $\bar{b}_\ell < 0$ para sacarla de la base. Para entrar a la base se elige la variable de índice $h \notin B$ con $\bar{a}_{\ell h} < 0$ (si no existe, el problema es infactible) tal que

$$h = \arg \max \left\{ \frac{\bar{c}_j}{\bar{a}_{\ell j}} : \bar{a}_{\ell j} < 0 \right\}$$

Ejemplo 1. Si partimos de la tabla siguiente,

	x_1	x_2	x_3	x_4	x_5	
x_4	-2	-2	-1	1	0	-6
x_5	-1	-2	-3	0	1	-5
	3	4	5	0	0	

elegiremos la fila de x_4 por tener el lado derecho menor, y dado que

$$\frac{3}{-2} = \max \left\{ \frac{3}{-2}, \frac{4}{-2}, \frac{5}{-1} \right\}$$

la variable x_1 sustituye en la base a x_4 . La nueva tabla se obtiene pivotando en la intersección de la fila y la columna elegidas, es decir,

- dividiendo la primera fila por -2 ,
- restando a la fila de x_5 la fila de x_4 dividida por 2,
- sumando a la fila de abajo la fila de x_4 multiplicada por $3/2$.

El resultado es la tabla

	x_1	x_2	x_3	x_4	x_5	
x_1	1	1	1/2	-1/2	0	3
x_5	0	-1	-5/2	-1/2	1	-2
	0	1	7/2	3/2	0	

Pivotando de nuevo en la fila de x_5 , columna de x_2 , se llega a la tabla

	x_1	x_2	x_3	x_4	x_5	
x_1	1	0	-2	-1	1	1
x_2	0	1	5/2	1/2	-1	2
	0	0	1	1	1	

Al ser los lados derechos no negativos, la solución óptima es

$$x_1 = 1, x_2 = 2, x_3 = x_4 = x_5 = 0$$

Ejemplo 2. Partimos del problema siguiente:

$$\begin{array}{ll} \text{máx} & x_1 + 2x_2 \\ \text{s.a} & 2x_1 + 6x_2 \leq 15 \\ & 28x_1 + 8x_2 \leq 77 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array}$$

En primer lugar debemos transformarlo en un problema de minimización con restricciones de igualdad:

$$\begin{array}{ll} \text{mín} & -x_1 - 2x_2 \\ \text{s.a} & 2x_1 + 6x_2 + x_3 = 15 \\ & 28x_1 + 8x_2 + x_4 = 77 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \end{array}$$

Al construir la correspondiente tabla

	x_1	x_2	x_3	x_4	
x_3	2	6	1	0	15
x_4	28	8	0	1	77
	-1	-2	0	0	

observamos que algunos \bar{c}_j son negativos. Añadiremos entonces una restricción artificial de la forma

$$x_1 + x_2 \leq M$$

donde M es un número suficientemente grande. El lado izquierdo de la restricción es la suma de las variables j con $\bar{c}_j < 0$. Pivotaremos en la nueva restricción, en la columna h con mínimo valor \bar{c}_h :

	x_1	x_2	x_3	x_4	x_5	
x_3	2	6	1	0	0	15
x_4	28	8	0	1	0	77
x_5	1	1	0	0	1	M
	-1	-2	0	0	0	

	x_1	x_2	x_3	x_4	x_5	
x_3	-4	0	1	0	-6	$15 - 6M$
x_4	20	0	0	1	-8	$77 - 8M$
x_2	1	1	0	0	1	M
	1	0	0	0	2	

A partir de aquí se sigue el procedimiento habitual. Si alguna de las variables básicas distintas de x_5 (la variables de holgura de la restricción artificial), al final del procedimiento, tomara un valor

en función de M , el problema sería no acotado. En otro caso, la solución óptima se obtiene de la forma habitual. La tabla final tras pivotar dos veces es

	x_1	x_2	x_3	x_4	x_5	
x_1	1	0	$-1/19$	$3/76$	0	$9/4$
x_5	0	0	$5/38$	$-1/36$	1	$M - 4$
x_2	0	1	$-7/38$	$-1/76$	0	$7/4$
	0	0	$18/19$	$1/76$	0	

Un problema de programación (lineal) entera es aquél que se puede establecer mediante la formulación

$$\begin{cases} \text{(P) mín} & c \cdot x \\ \text{s. a} & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{cases}$$

con $c \in \mathbb{R}^n$, $A_{m \times n} = (a_{ij})$, $a_{ij} \in \mathbb{R} \forall i, j$, $b \in \mathbb{R}^m$.

Es decir, añadiendo restricciones de integridad $x_j \in \mathbb{Z}$ sobre las variables de un problema lineal.

Ejemplo

$$\begin{cases} \text{mín} & -40x_1 - 55x_2 \\ \text{s.a} & x_1 \leq 4 \\ & 6x_1 + 11x_2 \leq 33 \\ & x_2 \leq 2 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

Nótese que, en este caso, las restricciones lineales definen un poliedro dentro del cual se encuentran las soluciones factibles del problema (aquéllas que toman valores enteros). Por esta razón, usando las mismas variables se pueden construir infinidad de problemas con distintos conjuntos de restricciones pero con la misma región factible:

Definición 6.0.3. Relajación lineal

Llamamos **relajación lineal** de (P) , (LP) , al problema de programación lineal que se obtiene suprimiendo las restricciones de integridad.

Denotaremos con $v(\bullet)$ al valor óptimo del problema (\bullet) y con $F(\bullet)$ a su región factible.

Proposición 6.0.1. Se satisface la siguiente relación entre los valores óptimos de un problema (de minimización) y el de su relajación lineal:

$$v(LP) \leq v(P)$$

Definición 6.0.4. Salto de dualidad

Se llama salto de dualidad a la diferencia $v(P) - v(LP)$.

Ejemplo 3. Se dispone de una mochila con volumen c y de n objetos para meter en ella. El j -ésimo objeto ocupa $c_j \geq 0$ unidades de volumen y reporta un beneficio $b_j \geq 0$. ¿Qué objetos deben ir en la mochila con el fin de obtener el máximo beneficio? Para formular el problema, pueden usarse las

siguientes variables:

$$x_j = \begin{cases} 1 & \text{si metemos a la mochila el objeto } j, \\ 0 & \text{en otro caso,} \end{cases} \quad j = 1, \dots, n$$

Ejemplo 4. Un empresario en crisis dispone de n empleados y debe llevar a cabo n tareas. La torpeza que el empleado i manifiesta al llevar a cabo la tarea j , t_{ij} , es medida por el empresario. El sindicato exige al empresario que asigne una tarea a cada empleado. Sacar al empresario del apuro lo menos dolosamente posible. Para formular este problema podemos definir las variables

$$x_{ij} = \begin{cases} 1 & \text{si se asigna la tarea } j \text{ al empleado } i, \\ 0 & \text{en otro caso,} \end{cases} \quad i, j = 1, \dots, n$$

Proposición 6.0.2. Si A es una matriz Totalmente Unimodular y $b \in \mathbb{Z}^m$, los vértices del poliedro $Ax = b$ son vectores enteros.

Veremos cómo se pueden formular algunos de los problemas de optimización que han ido apareciendo en los temas anteriores. Partiremos de la matriz de adyacencia $A = (a_{ij})$ y de la función de longitud ℓ .

¿Cuál es el número mínimo de colores necesarios para colorear los vértices del grafo? Se pueden utilizar variables binarias y_k que toman el valor 1 si se utiliza el k -ésimo color, con $k = 1, \dots, n$ (el número de variables puede reducirse utilizando una cota superior), y variables binarias x_{ik} que toman el valor 1 si el nodo i recibe el color k .

- **El problema de camino más corto entre dos vértices s y t .** Las variables (binarias) son

$$x_{ij} = 1 \text{ si el camino atraviesa el nodo } i \text{ y a continuación el } j.$$

- **El problema del árbol generador de peso mínimo.** Variables binarias $x_e = 1$ si la arista e pertenece al árbol $\forall e \in E$
- **El problema de convertir un grafo en euleriano** (para resolver el problema del cartero chino). x_{ij} = número de veces que se copia la arista (i, j) .
- **El problema del viajante de comercio.** $x_e = 1$ si la arista e pertenece al ciclo

Experimentar con distintas formulaciones es muy importante a la hora de usar un método de optimización discreta para resolver un problema. Una buena cota inferior (en caso de minimización) lo es todo. La cota puede ser realmente mala si no se presta atención a la formulación.

Ejemplo 5. Calcular la cota inferior que aporta la relajación lineal y el valor óptimo del problema siguiente:

$$\begin{cases} \text{mín} & -x_1 \\ \text{s.a} & x_3 - \sqrt{2}(x_1 - x_2) = 0 \\ & x_2 + x_4 = 1 \\ & x_i \geq 0 \text{ enteras } \forall i \end{cases}$$

Ejemplo 6. Encontrar desigualdades válidas para esta formulación en la que M es un número grande:

$$\begin{cases} \text{máx} & x_2 \\ \text{s.a} & x_2 \leq Mx_1 \\ & Mx_1 + x_2 \leq M + 1 \\ & x_i \geq 0 \text{ enteras } \forall i \end{cases}$$

Veremos cómo formular las siguientes situaciones:

- Consideremos dos restricciones sencillas, $x \leq a, y \leq b$. Queremos que toda solución factible cumpla al menos una de las dos.

$$- z \leq \min\{x, y\}$$

$$- z \geq \max\{x, y\}$$

$$- z \geq \min\{x, y\}$$

$$- z \leq \max\{x, y\}$$

$$- x \neq y.$$

$$- \text{Si } x < a \text{ entonces } y \geq b.$$

En muchas ocasiones las variables que se utilizan para formular son binarias. Veremos cómo se pueden expresar mediante restricciones lineales las situaciones siguientes (todas las variables son binarias):

$$- z = \max\{x, y\}$$

$$- z = xy$$

$$- z \text{ toma el valor 1 si exactamente una de las dos variables } x, y \text{ toma el valor 1 :}$$

6.1 - Problemas de optimización sobre grafos

Ejercicio 7. El problema del árbol generador del peso mínimo

$$x_e = 1 \text{ si la arista } e \text{ pertenece al árbol } \forall e \in E$$

$$\begin{cases} \min & \sum_{e \in E} l_e x_e \\ \text{s.a.} & x_e \in \{0, 1\} \forall e \in E \\ & \sum_{e \in E} x_e = n - 1 \\ & \sum_{e \in E(V^3)} x_e \leq 2 \quad \forall V^3 \subset V \quad |V^3| = 3 \\ & \sum_{e \in E(V^4)} x_e \leq 3 \quad \forall V^4 \subset V \quad |V^4| = 4 \\ & \vdots \\ & \sum_{e \in E(S)} x_e \leq |S| - 1 \quad \forall S \subset V \quad 3 \leq |S| \leq n - 1 \end{cases}$$

Haremos ahora la llamada formulación MTZ, que utiliza "una especie de árbol dirigido" comenzamos definiendo las variables:

$$u_i = \text{algo parecido al nivel del nodo } i \text{ en la arborescencia}$$

$$x_{ij} = \begin{cases} 1 & \text{si } i \text{ es el predecesor inmediato de } j \text{ en el árbol con raíz en } 1 \\ 0 & \text{oc} \end{cases}$$

$$\left\{ \begin{array}{ll} \min & \sum_i \sum_{j=(i,j) \in E} l_{ij} x_{ij} \\ \text{s.a.} & x_{ij} \in \{0, 1\} \quad \forall i, j : (i, j) \in E \\ & \sum_i x_{ij} = 1 \quad \forall j \neq 1 \\ & x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \in E \\ & u_i \in \mathbb{Z}^+ \quad \forall i \\ & u_1 = 0 \\ & u_j \geq u_i + 1 - M(1 - x_{ij}) \quad \forall i, j : (i, j) \in E \quad (x_{ij} = 1 \implies u_j \geq u_i + 1) \end{array} \right.$$

Podemos cambiar la M por $n - 1$ ya que $u_j \leq n - 1 \quad \forall j$, creando una mejor formulación del problema.

Ejercicio 8. El problema del camino más corto entre dos vértices s y j

Usaremos longitudes no negativas, y sea $x_{ij} = 1$ si el camino atraviesa el nodo i y a continuación el j .

$$\left\{ \begin{array}{ll} \text{Min} & \sum_{i \in V} \sum_{j \in V : (i,j) \in E} l_{ij} x_{ij} \\ \text{s.a.} & x_{ij} \in \{0, 1\} \quad \forall i, j : (i, j) \in E \\ & \sum_{j : (j,s) \in E} x_{sj} = 1 \\ & x_{js} = 0 \quad \forall j : (j, s) \in E \\ & \sum_{j : (j,t) \in E} x_{jt} = 1 \\ & x_{tj} = 0 \quad \forall j : (j, t) \in E \\ & \sum_{i : (i,j) \in E} x_{ij} = \sum_{i : (i,j) \in E} x_{ji} \quad \forall j \neq s, t \quad \text{Para todo nodo por el que entres, sales} \end{array} \right.$$

Podemos ver que la tercera y la cuarta restricción no son necesarias ya que se obtienen de las otras.

Veamos ahora otra formulación, si tenemos la estructura de árbol con raíz en s que contiene los caminos más cortos, supongamos que tenemos que enviar canicas desde la raíz de forma que llegue una a cada nodo. Las variables serán las canicas que pasan por cada arista.

$$x_{ij} = n^o \text{ de items (canicas) que circulan desde } i \text{ hasta } j \equiv$$

$$\equiv n^o \text{ de caminos más cortos desde } s = 1 \text{ que contienen el subcamino } i, (i, j), j$$

$$\left\{ \begin{array}{l} \min \quad \sum_i \sum_{j:(i,j) \in E} l_{ij} x_{ij} \\ \text{s.a.} \quad \sum_{j:(1,j) \in E} x_{1j} = n - 1 \\ \sum_{j:(i,j)} x_{ij} = \sum_{j:(j,i) \in E} x_{ji} - 1 \quad \forall i \neq 1 \\ x_{ij}, x_{ji} \in \mathbb{Z}^+ \quad \forall (i,j) \in E, i < j \end{array} \right.$$

Ejercicio 9. El problema de convertir un grafo en euleriano

$x_e \equiv$ número de veces que se recorre la arista e

$$\left\{ \begin{array}{l} \min \quad \sum_{e \in E} x_e \\ \text{s.a.} \quad x_e \in \mathbb{Z}^+ \quad \forall e \in E \\ x_e \geq 1 \quad \forall e \in E \\ \sum_{e \in E : e=(i,-)} x_e = 2z_i \quad \forall i \in V \\ z_i \in \mathbb{Z}^+ \quad \forall i \\ z_i \leq 2g(i) \end{array} \right. \quad \text{Opcional, es para tener una mejor formulación}$$

Ejercicio 10. El problema del viajante de comercio

$x_e = 1$ si la arista e pertenece al ciclo

$$\left\{ \begin{array}{l} \min \quad \sum_{e \in E} l_e x_e \\ \text{s.a.} \quad \sum_{e \in w(v)} x_e = 2 \quad \forall v \in V \\ \sum_{e \in w(S)} x_e \geq 2 \quad \forall S \subset V \quad 2 \leq |S| \leq n - 1 \\ x_e \in \{0, 1\} \quad \forall e \in E \end{array} \right.$$

Ramificación y acotación. Hiperplanos de corte

7.1 - Algoritmo de ramificación y acotación

Aunque nosotros lo utilizaremos para el problema de optimización (lineal) discreta, este tipo de algoritmo puede ser usado para resolver cualquier problema de la forma

$$\min\{f(x) : x \in P\}$$

Se basa en descomponer la región factible P en $\cup_{i=0}^{\ell} P_i$ y resolver los subproblemas

$$\min\{f(x) : x \in P_i\}$$

para cada $i \in \{1, \dots, \ell\}$, tomando luego la solución que dé el menor de los valores de $f(x)$. En la práctica, estos subconjuntos se van generando dinámicamente, intentando reducir su número en lo posible.

En el caso de un problema de Programación Lineal Entera (mixta)

$$\begin{aligned} (P) \quad & \min cx \\ & \text{s.a } Ax \leq b \\ & x_j \in \mathbb{Z}^+ \quad \forall j \in J \end{aligned}$$

se pueden considerar los subproblemas obtenidos al añadir a la formulación restricciones lineales del tipo $x_j \leq u$ o $x_j \geq v$ para $j \in J$ y $u, v \in \mathbb{Z}$. En particular, $P = P_0 \cup P_1$ con $P_0 = P \cap \{x_j \leq u\}$ y $P_1 = P \cap \{x_j \geq u + 1\}$, $u \in \mathbb{Z}$

Ejemplo 1.

$$(P_0) \quad (P) + \{x_1 \leq 4\}$$

$$P(1) \ (P) + \{x_1 \geq 5\} \left\{ \begin{array}{l} (P_{10}) \ (P_1) + \{x_2 \leq 0\} \\ (P_{11}) \ (P_1) + \{x_2 \geq 1\} \end{array} \right.$$

Supondremos por simplicidad que la región factible de la relajación lineal es acotada.

7.2 - Algoritmo de Ramificación y Acotación (Branch-and-bound)

Paso 1

Se construye la relajación lineal del problema, (P_0) . Se crea una lista $L = \{(P_0)\}$ de subproblemas sin resolver. Se fija la cota superior en $z_{UB} = +\infty$. El contador t se pone a 1.

Paso 2

Se elige un subproblema de L y se elimina. Se resuelve (típicamente con el método dual del símplex). Pueden darse los casos siguientes.

1. La solución x^* del subproblema es entera en J . Entonces si su valor óptimo es menor que z_{UB} , se fija z_{UB} a este valor y se anota x^* .

2. El subproblema es infactible. No se hace nada.

3. El subproblema es factible y su solución óptima no es entera en J . Si $cx^* \geq z_{UB}$, no se hace nada. En otro caso, se elige una variable $j \in J$ que tome un valor fraccional x_j^* .

(a) Se crea un nuevo subproblema (P_t) añadiendo al subproblema resuelto la restricción $x_j \leq \lfloor x_j^* \rfloor$ y se añade a L . Se aumenta t en 1.

(b) Se crea un nuevo subproblema (P_t) añadiendo al subproblema resuelto la restricción $x_j \geq \lceil x_j^* \rceil$ y se añade a L . Se aumenta t en 1.

Paso 3

Si $L = \emptyset$, fin. La solución óptima es x^* . En otro caso, se vuelve al paso anterior.

Obsérvese que en dos lugares del algoritmo se debe llevar a cabo una elección y que en los ejemplos anteriores no siempre hemos procedido de la misma forma al elegir. No hay una forma de actuar que sea siempre mejor. Para elegir el subproblema de la lista se puede optar por usar el de menor cota inferior, y para elegir la variable de ramificación es recomendable usar la que tome en el óptimo el valor con parte fraccional más cercana a 0.5.

7.3 - Hiperplanos de corte

Los métodos de hiperplanos de corte son métodos algebraicos que se utilizan en ocasiones para resolver el problema de programación lineal entera pero fundamentalmente para mejorar las cotas inferiores de las relajaciones lineales en los distintos nodos del árbol de ramificación.

7.4 - Hiperplano de corte todo entero

Partimos del problema de optimización lineal entero (puro) de la forma

$$\begin{aligned}
 (P) \text{ m\u00ednc} x \\
 \text{s.a } Ax = b \\
 x_j \in \mathbb{Z}^+ \quad \forall j
 \end{aligned}$$

Supondremos que las entradas de la matriz A y los lados derechos b son n\u00fameros enteros, de donde se sigue que las variables de holgura tambi\u00e9n tendr\u00e1n que tomar valores enteros.

Resolvemos la relajaci\u00f3n lineal de (P) y obtenemos la soluci\u00f3n \u00f3ptima \bar{x} . Sea x_k una variable b\u00e1sica que no toma valor entero en esta soluci\u00f3n (elegimos la que tenga parte fraccional m\u00e1s cercana a $1/2$). La k -\u00e9sima fila de la tabla final del algoritmo dual del \u00e1simplex expresa la igualdad, que satisfacen todas las soluciones factibles del problema,

$$x_k + \sum_{j \notin B} y_{kj} x_j = \bar{x}_k,$$

donde B es el conjunto de \u00edndices de las variables b\u00e1sicas e y son las entradas de la tabla.

Se separan las constantes de la igualdad en parte entera y parte fraccional:

$$\begin{aligned}
 x_k + \sum_{j \notin B} (\lfloor y_{kj} \rfloor + f_{kj}) x_j &= \lfloor \bar{x}_k \rfloor + f_k \\
 - \sum_{j \notin B} f_{kj} x_j &\leq -f_k
 \end{aligned}$$

Esta desigualdad no es satisfecha por la soluci\u00f3n \u00f3ptima de la relajaci\u00f3n lineal, que toma el valor 0 en las variables no b\u00e1sicas, pero s\u00ed por todas las soluciones enteras.

7.5 - Hiperplano de corte mixto

Partimos del problema de optimizaci\u00f3n lineal entero-mixto de la forma

$$\begin{aligned}
 (P) \text{ m\u00ednc} x \\
 \text{s.a } Ax = b \\
 x_j \in \mathbb{Z}^+ \quad \forall j \in J \\
 x_j \geq 0 \quad \forall j \notin J.
 \end{aligned}$$

Resolvemos la relajaci\u00f3n lineal de (P) y obtenemos la soluci\u00f3n \u00f3ptima \bar{x} .

Destacamos de la tabla final del m\u00e9todo dual del \u00e1simplex la fila correspondiente a la variable x_k :

$$x_k + \sum_{j \notin B} y_{kj} x_j = \bar{x}_k$$

de donde

$$x_k + \sum_{j \notin B} y_{kj} x_j = \lfloor \bar{x}_k \rfloor + f_k$$

donde como antes f_k es la parte fraccional de \bar{x}_k .

Llamamos

$$\bar{B}^+ := \{j \notin B : y_{kj} > 0\}$$

y

$$\bar{B}^- := \{j \notin B : y_{kj} < 0\}.$$

Esta desigualdad

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

no es satisfecha por la solución óptima de la relajación lineal, que toma el valor 0 en las variables no básicas, pero sí por todas las soluciones enteras.

Ejercicios

Ejercicio 1. (*Hoja 2, ejercicio 21*)

$$G = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 4), (2, 3), (3, 4)\})$$

$$L(G) = (\{12, 13, 14, 23, 34\}, \{(12, 13), (12, 23), (12, 14), (13, 34), (13, 23), (23, 34), (34, 14)\})$$