7 Teoría de grafos

7.1 Introducción

En numerosos problemas cuantificables, en las organizaciones, intervienen una serie de elementos entre los que se establecen unas relaciones: por ejemplo, los problemas relacionados con posibilidades de comunicación (redes de comunicación y de transporte), relaciones de orden entre actividades (planificación de proyectos mediante PERT) o estructuras de producto complejas (gestión de inventarios mediante MRP).

Los grafos son una herramienta que permite modelizar relaciones de esta naturaleza, de modo que se puedan resolver problemas asociados a esas circunstancias, frecuentemente de forma menos costosa que utilizando otras técnicas como la programación lineal.

Una buena comprensión de la teoría de grafos pasa por dominar la nomenclatura y conceptos asociados a estas representaciones de relaciones entre elementos, así como sus diversas formas de representación.

Seguidamente se definirá formalmente un grafo, pasando posteriormente a mostrar las diversas representaciones que admite. Con estos elementos podemos definir cómodamente los diversos elementos conceptuales asociados a los grafos

7.1.1 Definición de grafo

Un grafo G (x, E) consta de un conjunto de elementos “x”, denominados *nodos* o *vértices*, y un listado de parejas de vértices E que expresa las relaciones entre dichos elementos.

Si no se considera el orden de los *vértices* en cada pareja, dichos pares se denominan *aristas*, y decimos que el grafo es *no orientado*.

Si se consideran las relaciones, el par de aristas se llama *arco* y el grafo es *orientado*. Un grafo no orientado puede siempre convertirse en orientado, expresando la doble relación entre los vértices.

7.1.2 Representación de grafos

Existen múltiples maneras de representar un grafo. Tomemos un grafo orientado G(x, E) definido como con un conjunto de vértices y arcos:

$$X = (1,2,3,4,5)$$
$$E = \{(1,5), (1,2), (2,5), (5,4), (3,4), (3,2), (2,3), (4,5)\}$$

Esta representación, pese a cumplir con los requerimientos de la definición, resulta poco práctica para la interpretación del grafo y la comprobación de propiedades relevantes de éste. Por este motivo, existen diferentes representaciones de los grafos:

1. La *representación gráfica*, adecuada para la interpretación y resolución de problemas en grafos pequeños o medianos.

2. La representación mediante *matriz asociada o de adyacentes*, especialmente útil para el tratamiento de problemas de grafos con programas informáticos.

3. Otras representaciones, como el *diccionario de grafo*, buscan definir el grafo de forma más compacta, en términos de posiciones de memoria. Pueden ser útiles para representar grafos de gran tamaño.

Representación gráfica

Tal como puede apreciarse, consiste en un gráfico en que los vértices se representan mediante puntos. Las conexiones se representarán de diferentes maneras, dependiendo de que el grafo sea orientado o no:

a) Si es relevante para la representación determinar cuál es el vértice origen y cuál el destino, las conexiones entre vértices se representan mediante flechas (denominadas *arcos*): tendremos entonces un *grafo orientado*.

b) Si no es relevante determinar el sentido de la relación entre vértices, tendremos un *grafo no orientado*: los vértices se unirán mediante segmentos, denominados *aristas*. El grafo definido anteriormente admite esta representación gráfica.

El grafo definido anteriormente se trata de un grafo orientado, que admite la siguiente representación gráfica:

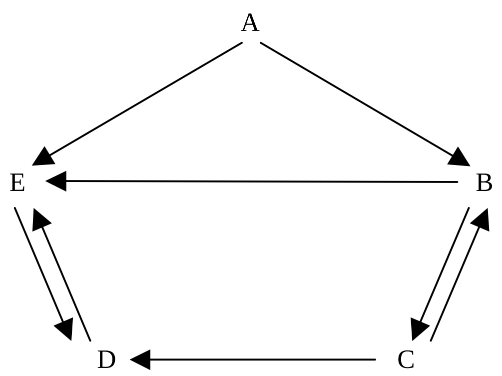


Figura 1.2.a

La figura 1.2.b muestra la representación de un grafo no orientado, en el que las conexiones no tienen una dirección establecida (o si se prefiere, una conexión entre dos vértices está definida en los dos sentidos posibles:

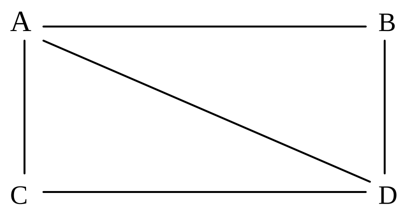


Figura 1.2.b

Matriz asociada al grafo o de adyacentes

Es una matriz **G** de $n \times n$, donde n es el número de vértices. Cada una de los componentes de la matriz representa una posibilidad de conexión: así, el componente g_{ij} representa la posibilidad de conexión existente entre el nodo origen i y el nodo destino j.

A continuación se muestra la matriz asociada al grafo del ejemplo. En este caso, se ha optado por representar por unos las conexiones existentes, y por ceros la ausencia de conexión.

$$\mathbf{G} = \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

Puede ocurrir que para la situación que deseemos representar, no sea relevante determinar cuál de los dos vértices del grafo asociados a una conexión es el origen y cuál es el destino: tendremos entonces un *grafo no orientado*. En estos grafos, se cumplirá que $g_{ij} = g_{ji}$. Tendremos entonces una matriz **G** simétrica.

Si los arcos tienen asociado un valor (p.e. distancias entre elementos), puede representarse en la matriz **G**. La matriz siguiente, por ejemplo, muestra las distancias por carretera entre Barcelona (BCN), Madrid (MAD), Bilbao (BIL) y Valencia (VAL). Por ser la distancia de ida igual a la de vuelta, se trata de un grafo no orientado.

	BCN	*MAD*	*BIL*	*VAL*
BCN		621	620	349
MAD	621		395	352
BIL	620	395		633
VAL	349	352	633	

Otras representaciones

Podemos tener representaciones más compactas, a precio de perder la posibilidad de representar valores asociados a los arcos.

Por ejemplo, en el *diccionario de grafo* se enumeran los destinos de los arcos que parten de cada nodo. El *diccionario inverso* enumera los orígenes de los arcos que inciden en cada nodo. Para el grafo orientado del ejemplo, tenemos los siguientes diccionarios:

1	2	5	
2	3	5	
3	2	4	
4	5		
5	4		

diccionario de grafo

1			
2	1	3	
3	2		
4	3	5	
5	1	2	4

diccionario inverso

La *lista de arcos*, próxima a la definición formal de grafo expresada más arriba, permite una representación compacta del grafo. Nuevamente para el ejemplo, la lista de arcos para el grafo del ejemplo queda como:

Origen	1	1	2	2	3	3	4	5
Destino	2	5	3	5	2	4	5	4

lista de arcos

7.2 Topología de grafos

La determinación de las condiciones que ha de cumplir un grafo para representar determinadas situaciones, o para que le sean aplicables determinados algoritmos, exige definir determinadas propiedades topológicas a cumplir por los grafos. Seguidamente se exponen algunas de estas propiedades, tanto para grafos orientados como para no orientados.

7.2.1 Grafos orientados

Grado y semigrados interior y exterior de un vértice

El *grado* de un vértice se define como el número total de arcos que inciden en dicho vértice, y evalúa su grado de conexión con el resto de vértices del grafo. Para un grafo orientado, podemos definir también los semigrados interior y exterior:

a) El *semigrado interior* de un vértice es el número de arcos con destino en el vértice. Para determinadas situaciones, un vértice con semigrado interior cero puede ser un origen del grafo.

b) El *semigrado exterior* de un vértice es el número de arcos con origen en el vértice. Un vértice con semigrado exterior cero puede representar, en determinadas situaciones, un destino del grafo.

Para el grafo de la figura 2.1.a, el vértice α tiene semigrado interior cero, y representa el origen del grafo. El vértice ω tiene semigrado exterior cero, y representa el destino del grafo.

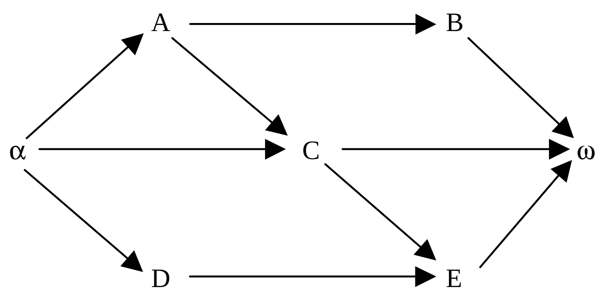


Figura 2.1.a

Caminos, bucles y circuitos

Dos vértices pueden no estar conectados directamente, pero sí indirectamente a través de un *camino*. Más formalmente, un camino es una sucesión de arcos tal que el vértice extremo de cada uno (exceptuando el último), coincide con el vértice extremo del siguiente en la sucesión.

Un vértice puede estar conectado consigo mismo de forma directa a través de un bucle, que no es más que un arco cuyos vértices origen y destino coinciden. También puede estar conectado a través de otros vértices mediante un circuito, definido como un camino cuyos vértices origen y destino coinciden. Dichos conceptos pueden ilustrarse en el grafo de la figura 2.1.b:

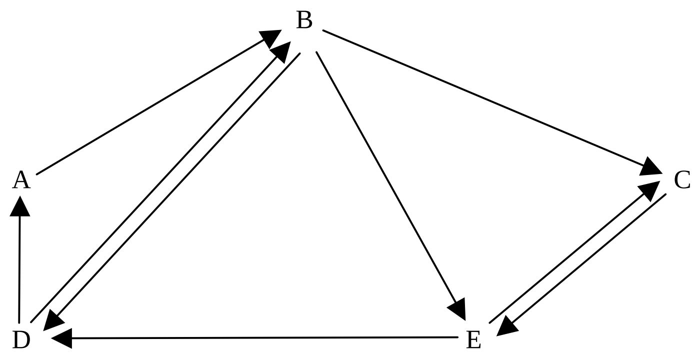


Figura 2.1.b

Se trata de un grafo orientado (por ejemplo, existe conexión directa entre A y D, pero no entre D y A). Uno de los *caminos* del grafo es el que une los vértices A, B y C. Por otra parte, encontramos un circuito que une A, B, C, E y D.

7.2.2 Grafos no orientados

Aunque algunos conceptos definidos para grafos orientados son válidos también para grafos no orientados: así, se definen de la misma manera los conceptos de *bucle* y de *grado* de un vértice. Sin embargo, no tiene sentido para grafos no orientados distinguir entre *semigrado interior* y *semigrado exterior*. Finalmente, los conceptos relacionados con la conexión entre vértices tienen definición específica en grafos no orientados.

Cadenas y ciclos

Una *cadena* puede definirse como un camino no orientado: es una sucesión de aristas tal que el vértice extremo de cada una (exceptuando la última) coincide con el vértice extremo de la siguiente en la sucesión. Dos vértices que no están conectados directamente pueden estarlo indirectamente mediante una cadena.

Asimismo, un vértice puede estar conectado consigo mismo directamente mediante un *bucle*, o bien, mediante un *ciclo*, definido como un camino que se inicia y termina en el mismo vértice. Para los grafos orientados, puede definirse también ciclo como un conjunto de arcos que unen una serie de vértices, prescindiendo de su orientación.

En el grafo de la figura 2.2.a, podemos observar que todos sus vértices están conectados mediante cadenas: el vértice A está conectado con el E a través de la cadena A – B – E. También podemos observar el ciclo A – B – C – D.

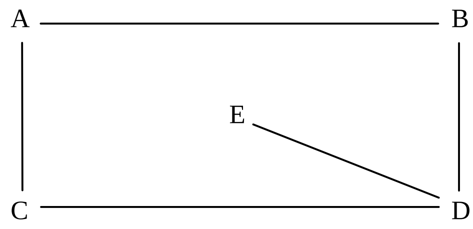


Figura 2.2.a

7.2.3 Conectividad en grafos

A diferencia de los conceptos anteriores, que afectaban a vértices del grafo, la conectividad es una propiedad del grafo en su conjunto. Un grafo orientado puede ser conexo o fuertemente conexo. Un grafo orientado sólo puede ser conexo. Más concretamente, tenemos:

Diremos que un grafo es *conexo* si existe al menos un ciclo entre toda pareja de vértices. El concepto es aplicable tanto a grafos orientados como para no orientados: obsérvese que se ha definido ciclo tanto para grafos orientados como para no orientados.

Diremos que un grafo orientado es *fuertemente conexo* si existe al menos un camino entre toda pareja de vértices. Todo grafo orientado fuertemente conexo será también conexo.

El grafo de la figura 2.3.a, con el que se ejemplificó el concepto de camino, es un grafo fuertemente conexo: podemos encontrar un camino entre cualquier pareja de vértices:

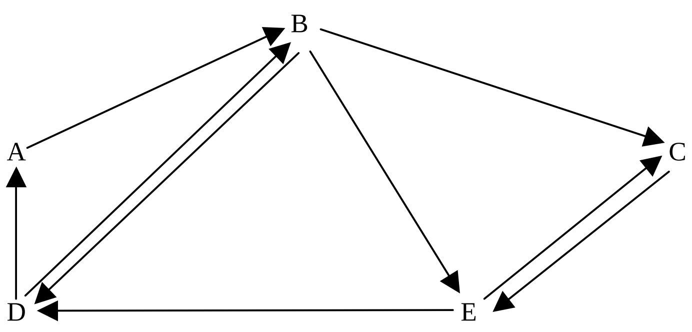


Figura 2.3.a

El grafo de la figura 2.3.b es un ejemplo de grafo conexo, pero no es, sin embargo, fuertemente conexo: No existe camino entre los vértices A y D.

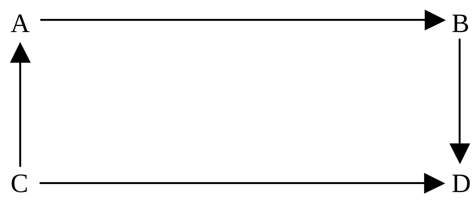


Figura 2.3.b

El grafo siguiente es un ejemplo de grafo conexo no orientado: podemos encontrar una cadena entre cualquier pareja de vértices:

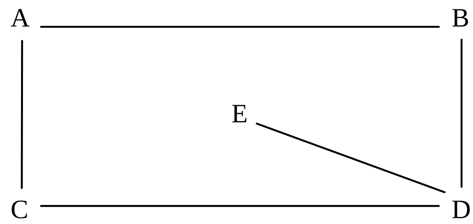


Figura 2.3.c

Finalmente, en la figura 2.3.d tenemos un grafo no orientado que no es conexo: D y E no están conectados con A, B y C. Puede apreciarse el ciclo de los vértices A, B y C (no es ciclo, sin embargo, la conexión existente entre D y E).

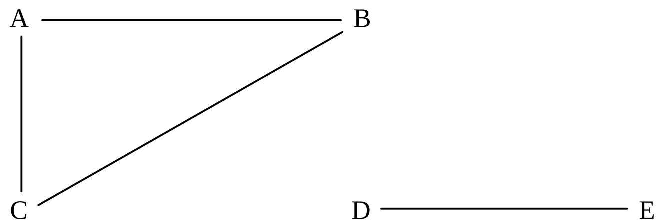


Figura 2.3.d

7.2.4 Árboles y arborescencias

Los árboles y arborescencias son un tipo particular de grafos, útil para representar determinadas situaciones.

Un *árbol* puede ser tanto orientado como no orientado, y no es más que un grafo conexo sin circuitos (grafos orientados) o ciclos (grafos no orientados).

Las figuras siguientes representan dos grafos no orientados. El ciclo A – B – C – D hace que el grafo de la figura 2.4.a no sea un árbol. El grafo de la figura 2.4.b, al no ser conexo y sin ciclos, sí es un árbol.

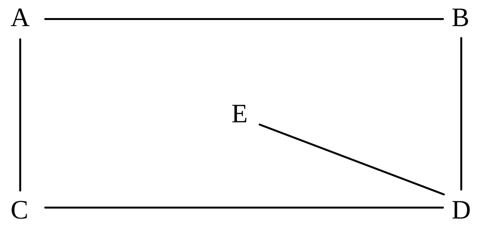


Figura 2.4.a

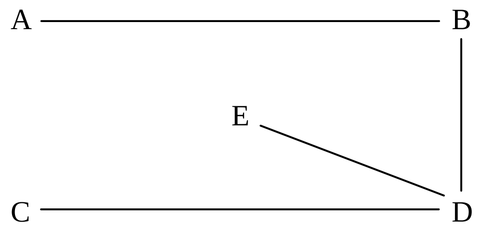


Figura 2.3.c

Una arborescencia es un grafo orientado, fuertemente conexo, sin ciclos ni bucles, en que todos los vértices tendrán semigrado interior igual a la unidad, excepto uno, raíz de la arborescencia, cuyo semigrado interior es 0. Las arborescencias son útiles para representar, por ejemplo, procesos decisionales. El grafo de la figura 2.3.d es un ejemplo de arborescencia, en la que la raíz es el vértice A.

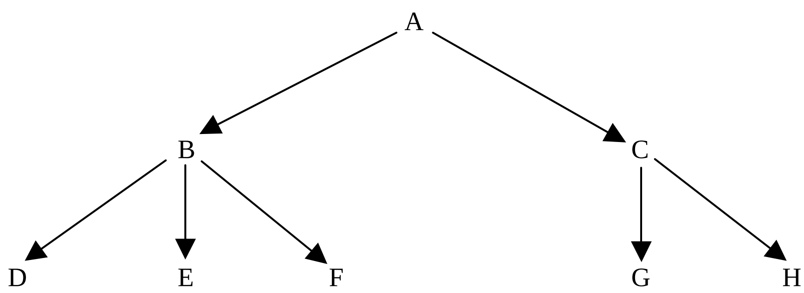


Figura 2.3.d

7.3 Modelización mediante grafos

Mediante la teoría de grafos pueden representarse gran número de situaciones que supongan relaciones entre diversos elementos. Seguidamente se exponen algunas de ellas.

7.3.1 Posibilidades de comunicación

De manera bastante natural, un grafo puede representar las posibilidades de comunicación existentes entre diferentes puntos. Lo más frecuente es que los puntos estén representados en el grafo mediante vértices, y las posibilidades de comunicación mediante arcos (o en ocasiones aristas, si la comunicación entre dos nodos es siempre igual entre los dos sentidos). La representación de las posibilidades de comunicación se completa asociando a cada arco una magnitud relevante para la representación (distancia, tiempo, etc.).

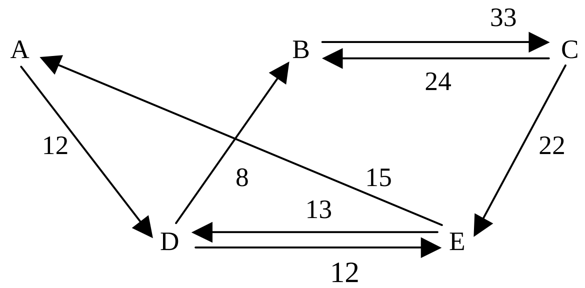


Figura 3.1.a

El grafo de la figura 3.1.a representa las diferentes posibilidades de comunicación entre cinco ciudades (A, B, C, D y E). Los valores sobre los arcos representan los tiempos de traslado de un punto a otro.

Mediante grafos de este tipo pueden resolverse problemas de conectividad (árbol parcial mínimo) o problemas de caminos, en los que se trata de encontrar el camino de mínima distancia entre dos vértices del grafo.

7.3.2 Redes de transporte

También es posible representar mediante un grafo orientado los flujos que circulan a través de una red de transporte. Una red de transporte deberá tener uno o varios *vértices origen* (de semigrado interior cero) y uno o varios *vértices destino* (de semigrado exterior cero). El resto de vértices, con semigrados interior y exterior no nulos, serán *vértices de transbordo*.

El grafo de la figura 3.2.a puede representar una red de distribución de agua entre diferentes puntos. El vértice A es el origen de la red, y el vértice E el destino.

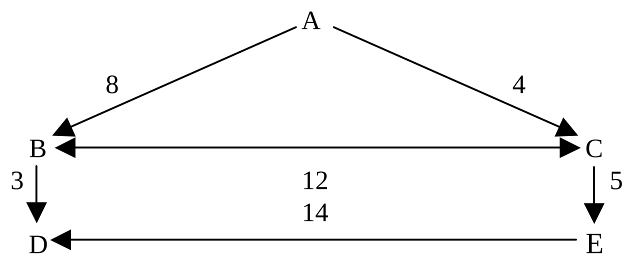


Figura 3.2.a

En una red de transporte, se asociarán a los arcos los valores del flujo que circula por dicha comunicación. Dichos flujos cumplirán algunas propiedades importantes:

1. En un *vértice de transbordo*, el total de los flujos entrantes (valores de los arcos que tienen como destino el vértice) será igual al de los flujos salientes (valores de los arcos que tienen como origen el vértice).

2. La suma de los valores de los flujos de los arcos que tienen como origen *vértices origen* será igual al de la suma de los valores de los flujos de los arcos que tiene como destino *vértices destino*. Dicho valor será el *flujo total* de la red de transporte.

Además del flujo que circula por el arco, podemos asociar otros valores a los arcos:

a) El flujo máximo y el flujo mínimo que pueden y deben, respectivamente, circular por un arco. Con dichos valores podremos resolver el problema del *flujo total máximo* que circula por una red de transporte.

b) El coste de vehicular una unidad de flujo por cada uno de los arcos. Dichos valores permitirán resolver el problema del *flujo de coste mínimo*, que consiste en vehicular por la red un determinado valor de flujo total.

7.3.3 Relaciones de orden

También pueden representarse mediante un grafo las relaciones de sucesión entre las actividades de un proyecto.

a	Pelar patatas	-
b	Batir huevos	-
c	Pelar cebolla	-
d	Freír patatas+cebolla	a,c
e	Añadir huevo y freír	b,d
f	Servir	e

El cuadro muestra la secuencia de actividades a realizar para elaborar una tortilla de patatas. La columna de la derecha representa las actividades inmediatamente anteriores para poder realizar la actividad en cuestión: por ejemplo, antes de iniciar deben haberse acabado a y b.

Es usual representar estas situaciones en grafos en los que las actividades pueden estar representadas por arcos (grafo PERT) o por nodos (grafo ROY):

Grafo PERT:

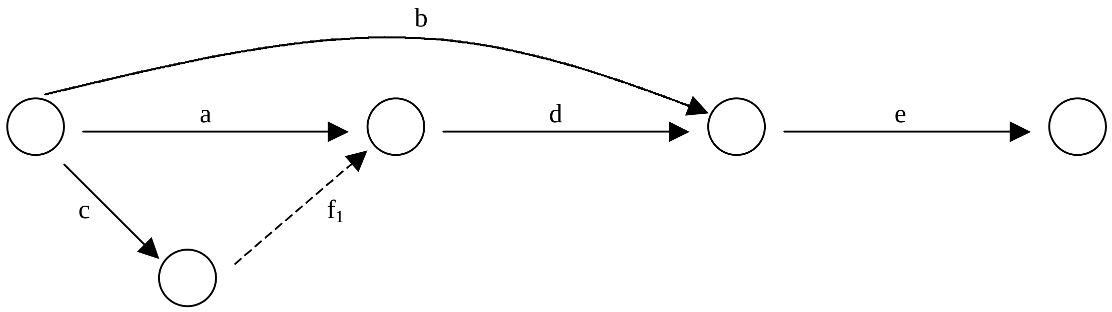


Figura 3.3.a: grafo PERT

Los vértices del grafo representan etapas (instantes de tiempo en que se termina o empieza una actividad). La actividad ficticia f_1 se ha añadido para respetar las restricciones propias del grafo PERT (entre dos etapas sólo puede haber una actividad).

El grafo ROY suministra la misma información que el PERT, diferenciándose de éste en que las actividades están representadas por vértices. Deben añadirse dos vértices representativos del inicio (I) y del final (F) del proyecto.

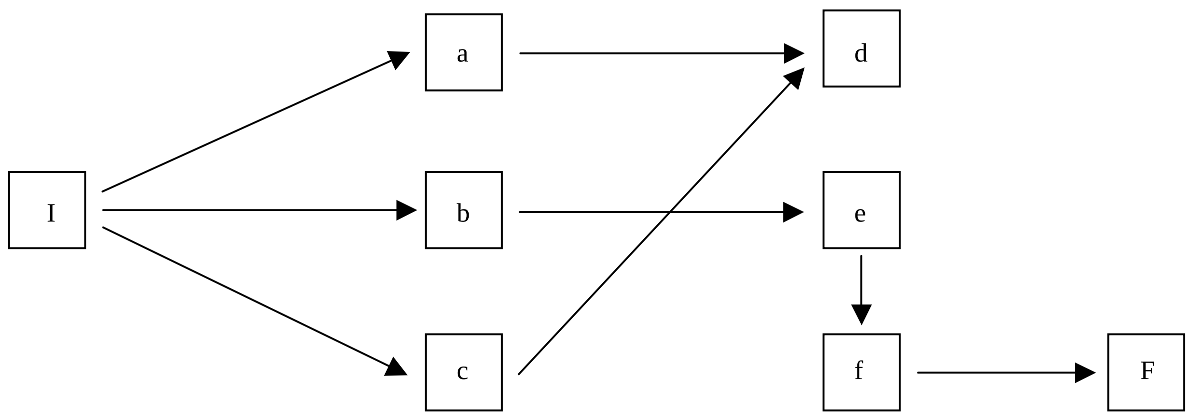


Figura 3.3.b: grafo ROY.

Estos grafos se emplean frecuentemente para la gestión de proyectos. Mediante su uso podemos determinar entre otros parámetros:

1. Tiempo en que puede finalizarse el proyecto.
2. Holgura de las actividades (retraso máximo que no afecta a la duración del proyecto).
3. Camino crítico (conjunto de actividades de holgura cero, que determina la duración del proyecto).

7.3.4 Estructura de un producto

Puede representarse mediante un grafo la estructura de un producto complejo. El grafo representa la estructura de dos productos relacionados: una estantería de tres anaqueles y otra de 6 anaqueles.

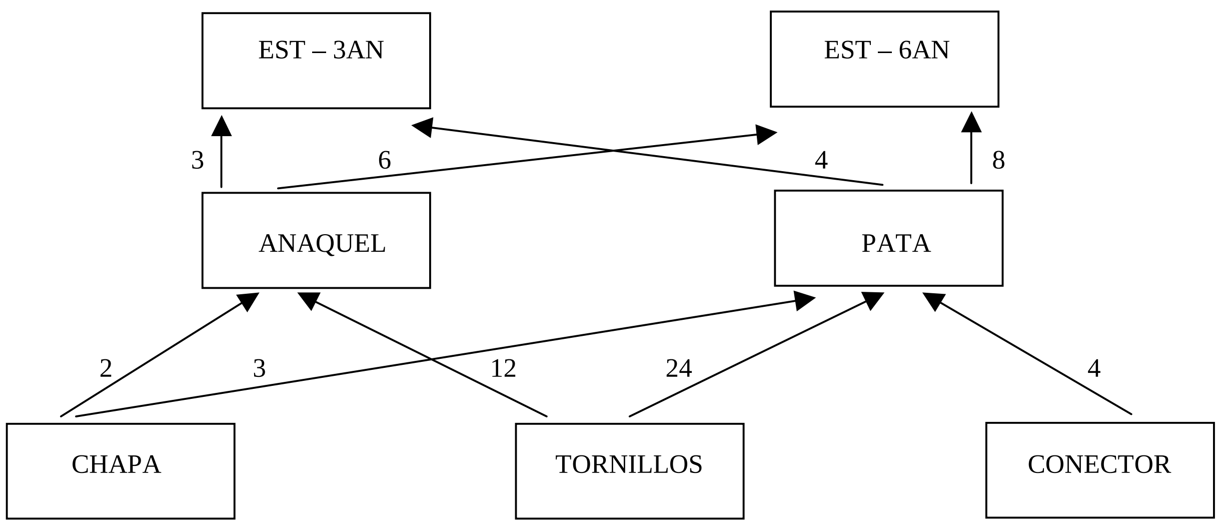


Figura 3.4.a

Este tipo de grafos se utiliza en procedimientos de previsión de la demanda dependiente (en este caso, de la chapa, tornillos y conectores) en el sistema de gestión de inventarios MRP I (*Material Requirements Planning*).

En este documento trataremos dos tipos de problemas:

a) *Problemas de caminos*: se trata de algoritmos que permiten establecer la conexión entre todos los nodos del grafo a mínimo coste (algoritmo de Prim), o bien encontrar el camino más corto entre dos nodos del grafo (algoritmos de Dijkstra y de Bellman – Kalaba).

b) *Problema de flujo total máximo*: se trata de determinar cuál es el flujo máximo que puede circular por una red de transporte (algoritmo de Ford – Fulkerson).

Finalmente, se presenta un problema más general (*flujo de costo mínimo*), que permite tratar los problemas anteriormente indicados mediante programación lineal.

7.4 Problemas de caminos

En los problemas de caminos, se trabaja con grafos cuyos arcos o aristas tienen asociado un determinado valor (que puede corresponder, aunque no necesariamente, a la distancia entre los vértices unidos por el arco o arista). Dichos problemas suelen reducirse a encontrar:

1. el conjunto de aristas que conecta todos los nodos del grafo, tal que la suma de los valores asociados a las aristas sea mínimo. Dado que dicho conjunto ha de ser necesariamente un árbol, suele llamarse a este problema de *árbol parcial mínimo*.

2. el *camino más corto* desde un vértice origen hasta otro extremo. Se denominará camino más corto a aquel cuya suma de los valores asociados de los arcos que lo componen sea mínimo.

A continuación se describen varios procedimientos para resolver estos dos problemas: el algoritmo de Prim, para el problema del árbol parcial mínimo; y para el problema del camino más corto, los algoritmos de Dijkstra y de Bellman – Kalaba.

7.4.1 Árbol parcial mínimo: algoritmo de prim

El problema del árbol parcial mínimo puede ser definido con precisión en los siguientes términos:

> Dado un grafo de referencia no orientado y conexo, con unos determinados valores asociados a sus aristas, se trata de hallar un subconjunto de aristas conexo tal que permita la conexión de toda pareja de vértices del grafo de referencia, de modo que la suma de los valores asociados al subconjunto de aristas sea mínima.

En otras palabras, se trata de conectar todos los vértices del grafo a un mínimo coste (suponiendo que los valores asociados a una arista representen el coste de conectar de manera directa los dos vértices que ésta une). La solución ha de ser forzosamente un árbol, dado su naturaleza de subgrafo conexo, y que la elección de una arista que haga ciclo con las restantes aumenta el coste del subgrafo, sin añadir nuevas posibilidades de conexión. De ahí que la solución al problema sea el *árbol parcial mínimo* del grafo.

El problema puede resolverse de manera sencilla con el algoritmo de Prim, que se detalla a continuación:

Paso 0	Iniciación: 1. Asociar a cada arista a_{ij} un valor d_{ij}. Si no hay posibilidad de conexión directa entre los vértices i, j dicho valor será infinito.

2. Sea S el conjunto de vértices conectados en la iteración en curso. Inicialmente, S está vacío.

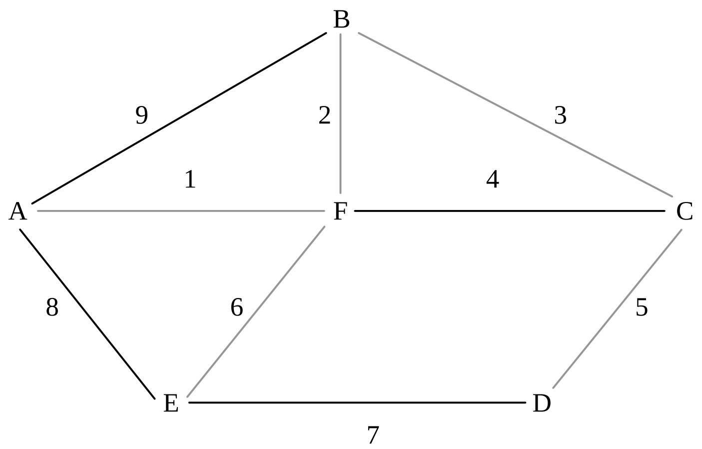
Paso 1 Tomar un vértice cualquiera e incorporarlo a S.

Paso 2 Buscar la arista a_{ij} de menor valor d_{ij} tal que: el vértice i pertenezca a S y el vértice j no pertenezca a S. Sean i^* y j^* los vértices de la arista así hallada. En caso de empate entre aristas, puede seleccionarse una cualquiera de ellas.

Paso 3 Incorporar el vértice j^* al conjunto S.

Paso 4 Si todos los vértices están conectados, finalizar.
Si no, ir a paso 2.

En el grafo que se indica a continuación se ha aplicado el algoritmo para obtener el árbol parcial mínimo, que está representado en rojo en el dibujo. El lector puede seguir el procedimiento cuando se comienza en el vértice A, observando el orden en que los vértices se incorporan al conjunto S.



S = { A, F, B, C, D, E }

7.4.2 Camino más corto entre dos vértices: algoritmo de Dijkstra

El problema del camino más corto puede formularse del siguiente modo:

> Dado un grafo conexo y orientado, a cuyos arcos se asocian determinados valores, encontrar un camino entre dos vértices origen (O) y final (F), si existe alguno[1], de forma que la suma de los valores asociados a los arcos que componen el camino sea mínima.

Para la resolución de este problema existen varios algoritmos, basados todos ellos en el *procedimiento de Ford*.

Fase 0 Asociar una etiqueta a cada vértice. Por ejemplo:
$P_O = 0$; $P_i = d_{O,i}$.

[1] Para asegurarnos de que el problema siempre tenga solución, debemos imponer que el grafo sea *fuertemente conexo*.

$d_{O,i}$ es el valor asociado al arco que une el vértice origen O con un vértice cualquiera i. Si no hay conexión directa, el valor de $d_{O,i}$ se considera infinito.

Fase 2 — Repetir las siguientes operaciones:
Buscar un arco tal que $P_i > P_j + d_{j,i}$
Hacer $P_i = P_j + d_{j,j}$

Fase 3 — Finalizar si no existe ningún arco que cumpla la condición.

La exploración puede realizarse de diferentes formas en la matriz de distancias origen (filas) / destino (columnas). Cada procedimiento de exploración dará lugar a un algoritmo, que será una variante del procedimiento de Ford.

Uno de los algoritmos que aplican este procedimiento es el algoritmo de Dijkstra. En dicho algoritmo, tenemos un *conjunto de vértices S* de los que ya se ha encontrado su distancia mínima al vértice origen O. Se exploran aquellos vértices en los que *inciden arcos con origen en algún vértice de S*. Se incorpora a S aquel vértice que resulte estar más cercano al origen después de la exploración.

La ventaja de este algoritmo es que puede finalizarse la exploración cuando F se haya incorporado al conjunto S. Su inconveniente es que se trata de un algoritmo lento: si el grafo cuenta con n vértices, son necesarias $n - 1$ exploraciones para tener todo el grafo cubierto.

Seguidamente se expone el contenido del algoritmo:

Algoritmo de Dijkstra

Paso 0 — Iniciación:

1. Asociar a cada arco a_{ij} un valor d_{ij}. Dicho valor será infinito si no hay conexión entre los vértices i y j.
2. Sea S el conjunto de vértices conectados en la iteración en curso. Incorporar a S el vértice origen O.
3. Hacer la etiqueta $P_O = 0$, y $P_i = \infty$, si i es diferente del origen.
4. Hacer la variable t_i = O.

Paso 1 — Sea y el último vértice incorporado a S. Para todo vértice z no incorporado a S, calcular:

$$P_z = \min \{ P_z, P_y + d_{yz} \}$$

Si , $P_y + d_{yz} < P_z$, hacer $t_z = y$.

Paso 2 — De entre todos los vértices z, determinar aquel z^* que cumple:

$$P_{z^*} = \min \{ P_z \}$$

Paso 3 — Si $P_{z^*} = \infty$, finalizar. No hay caminos de longitud finita entre O y los vértices no pertenecientes a S.
Si no, *continuar*.

Paso 4 — Incorporar z^* al conjunto S. P_{z^*} será el valor de la distancia mínima de z al vértice origen O.

Paso 5 — Si F = z^*, *finalizar*.
Si no, *ir a paso 1*.

En cada iteración, P_i representa el valor del mejor camino desde O hasta el vértice *i* hallado hasta el momento.

En el grafo de la figura 4.2.a se ha hallado el camino más corto desde el vértice A hasta el F. Han sido necesarias cinco iteraciones para encontrarlo. La tabla indica el valor de la etiqueta P_i para cada vértice en cada una de las iteraciones. En negrilla se indica la etiqueta del z^* considerado, que es definitiva y no vuelve a indicarse en la tabla.

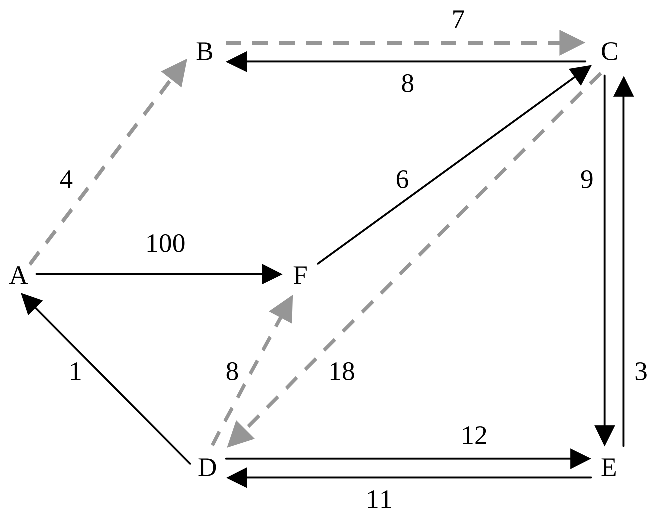


Figura 4.2.a

	k = 0	k = 1	k = 2	k = 3	k = 4	k = 5
A	**0**					
B	∞	**4 (A)**				
C	∞	∞	**11 (B)**			
D	∞	∞	∞	30 (C)	**29 (C)**	
E	∞	∞	∞	**20 (C)**		
F	∞	100 (A)	100 (A)	100 (A)	100 (A)	**37 (D)**
z^*	A	B	C	E	D	F

El camino más corto entre A y F tiene un valor de 38 y pasa por los vértices:

A – B – C – D – F

Así, el camino estará compuesto por los arcos siguientes, en el orden que se indica:

$\{a_{AB}, a_{BC}, a_{CD}, a_{DF}\}$

El camino se ha determinado a través de los valores entre paréntesis, que indican el origen del arco que forma parte del camino mínimo. Así, desde F, podemos ir reconstruyendo ese camino.

Nótese como, al estar todos los vértices del grafo en S, también se han hallado los caminos más cortos entre A y el resto de vértices del grafo: si el camino más corto entre A y D es A – B – C – D, entonces el camino más corto entre A y C es A – B – C.

Hemos utilizado la representación gráfica del grafo para la resolución. Podríamos haber usado la representación matricial, que nos habría obligado a explorar la matriz por filas: al explorar las fila *i* de la matriz **G**, estamos examinando todos los arcos con origen en el vértice *i*, con destino en el resto de nodos del grafo.

7.4.3 Camino más corto entre un vértice y todos los demás: algoritmo de Bellman–Kalaba

Si el algoritmo de Dijkstra exploraba la matriz del grafo por filas, el de Bellman–Kalaba lo hace por columnas: se determina la etiqueta de un vértice en una iteración a partir de las etiquetas de los vértices de los que proceden los arcos que llegan al vértice considerado.

Las etiquetas P_i y t_i, así como las distancias $d_{j,i}$ se definen de la misma manera que en algoritmo de Dijkstra. Sin embargo, ahora ninguna de las etiquetas se considera permanente, hasta que finalice el procedimiento. Esto ocurre cuando los valores de las etiquetas de todos los vértices no han variado en una iteración.

En definitiva, tendremos el algoritmo:

Paso 0 — Iniciación:

1. Asociar a cada arco a_{ij} un valor d_{ij}. Dicho valor será infinito si no hay conexión entre los vértices *i* y *j*.
2. Hacer $P_O = 0$, y $P_i = d_{Oi}$, para los *i* distintos de O.
3. Para todo vértice, hacer $t_i = O$.

Paso 1 — Cuerpo del algoritmo:

1. Inicializar el detector de cambios: $c = 0$
2. Para todo i hacer:
3. Para todo *j* determinar el min $\{ P_j + d_{ji} \}$. Sea j^* el vértice donde se ha hallado el mínimo.
4. Si $P_{j*} + d_{j*i} < P_i$, entonces hacer:
   - $P_i = P_{j*} + d_{j*i}$
   - $t_i = j^*$
   - ch = 1

Paso 2 — Si $c = 1$, *ir a Paso 1*.
Si $c = 0$, *finalizar*.

El resultado final es el camino más corto entre el origen O y el resto del grafo. Se expresa a partir de:

a) Un conjunto de etiquetas P_i que indican el valor del camino más corto del origen hasta el vértice *i*.

b) Un conjunto de etiquetas t_i que indica (excepto para el origen) el vértice que precede a *i* en el camino más corto de O a *i*. Dichas etiquetas nos permiten determinar el camino más corto de O a un vértice cualquiera, de manera similar a como se encontró en el apartado anterior.

El algoritmo puede aplicarse:

1. Sobre la representación gráfica: hay que considerar los arcos que llegan a cada uno de los vértices.

2. Sobre la matriz: hay que considerar los valores de las columnas de **G**: en la columna i encontraremos aquellos arcos cuyo destino sea el vértice *i*.

Seguidamente se resolverá el mismo problema con que se ejemplificó el algoritmo de Dijkstra, usando Bellman–Kalaba:

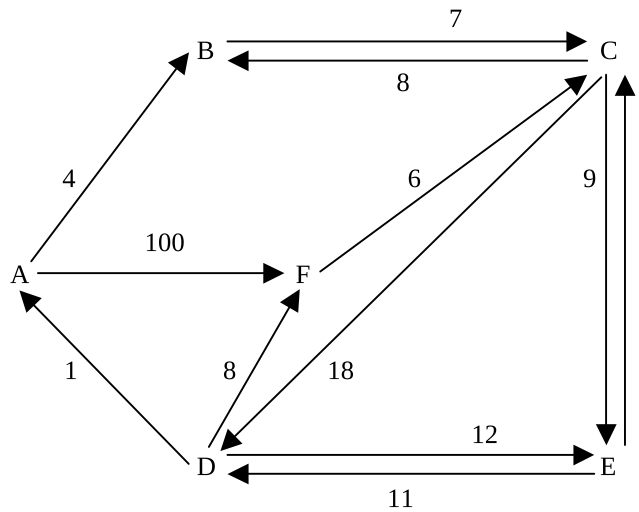


Figura 4.3.a

	k = 0	k = 1	k = 2	k = 3	k = 4
A	0 (A)	0 (A)	0 (A)	0 (A)	0 (A)
B	4 (A)	4 (A)	4 (A)	4 (A)	4 (A)
C	∞ (A)	11 (B)	11 (B)	11 (B)	11 (B)
D	∞ (A)	∞ (A)	29 (C)	29 (C)	29 (C)
E	∞ (A)	∞ (A)	20 (C)	20 (C)	20 (C)
F	100 (A)	100 (A)	100 (A)	37 (D)	37 (D)

Se han determinado los caminos mínimos desde A hasta el resto de vértices en un número de iteraciones menor que para el algoritmo de Dijkstra. Para la obtención de las etiquetas en la iteración *k*, se han utilizado las de la iteración *k-1*. Podemos acelerar algo el proceso si utilizamos los valores de las etiquetas ya obtenidas en la iteración.

	k = 0	k = 1	k = 2
A	0 (A)	0 (A)	0 (A)
B	4 (A)	4 (A)	4 (A)
C	∞ (A)	12 (B)	12 (B)
D	∞ (A)	29 (C)	29 (C)
E	∞ (A)	20 (C)	20 (C)
F	100 (A)	37 (D)	37 (D)

Esta última mejora del algoritmo puede formularse formalmente como sigue:

Paso 0 — Iniciación:

1. Asociar a cada arco a_{ij} un valor d_{ij}. Dicho valor será infinito si no hay conexión entre los vértices *i* y *j*.

2. Hacer $P_O = 0$, y $P_i = d_{Oi}$, para los i distintos de O.
3. Para todo vértice, hacer t_i = O.

Paso 1 Cuerpo del algoritmo:
1. Inicializar el detector de cambios: $c = 0$
2. Para todo i, y para todo j, hacer:
   - Si $P_i > P_j + d_{ji}$, hacer $P_i = P_j + d_{ji}$, $t_i = j$, $c = 1$
   - Si no, *continuar*.

Paso 2 Si $c = 1$, *ir a Paso 1*.
Si $c = 0$, *finalizar*.

7.5 Problemas de flujos

Es frecuente que nos encontremos con situaciones en las que debamos tratar con magnitudes de flujo: fluidos que circulan por un sistema de tuberías, cadenas de montaje en la que las máquinas tienen diferentes valores de productividad, etc. Estas situaciones pueden modelizarse mediante un grafo que represente una red de transporte (ver 3.2). Los problemas más frecuentes en estos casos consisten en:

a) Determinar el *flujo total máximo* que es capaz de vehicular el sistema.

b) Si conocemos el coste de vehicular flujo por cada uno de los vértices del grafo, puede interesarnos conocer cuánto flujo debe circular por cada arista para así obtener el *flujo a coste mínimo* para un determinado valor del flujo total.

Ambos problemas pueden resolverse por programación lineal, pero el de flujo total máximo puede resolverse con métodos más rápidos, obteniendo el corte de flujo mínimo o a través del algoritmo de Ford–Fulkerson.

7.5.1 El problema del flujo total máximo

Además de las condiciones genéricas para redes de transporte indicadas en la sección 2.3, suele imponerse a la red de transporte las siguientes condiciones:

1. *La red no debe tener bucles ni circuitos*. Los circuitos no añaden flujo total al sistema: si la red original tiene alguno, puede eliminarse del modelo alguno de los arcos del circuito a efectos de obtener el flujo máximo.

2. *La red debe tener un único origen y un único destino* (tal como se definieron en 2.3). Esta condición puede cumplirse en todo caso, añadiendo un origen ficticio (del que salen arcos de capacidad ilimitada, con destino en los diferentes orígenes) y un destino ficticio (al que llegan arcos de capacidad ilimitada, con origen en los diferentes destinos. Esta situación se ejemplifica en las figuras 5.1.a y 5.1.b.

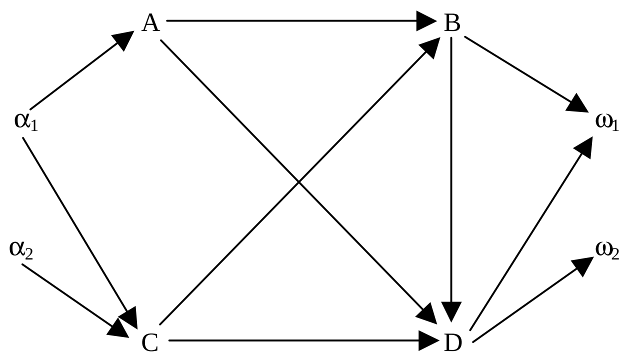


Figura 5.1.a

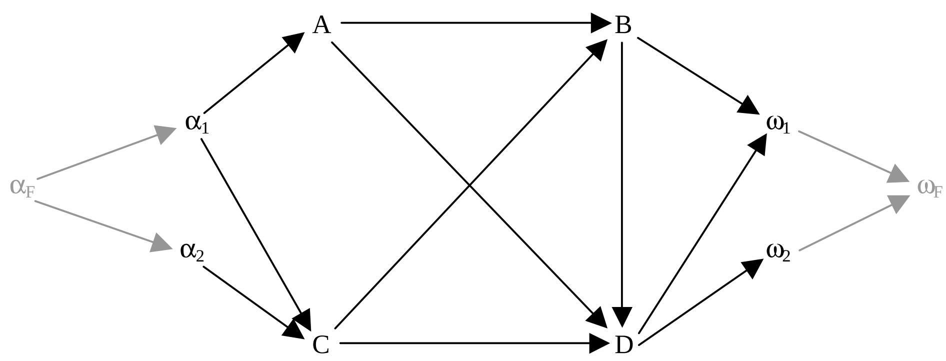


Figura 5.1.b

El problema consiste en determinar el flujo máximo que circulará a través de la red de transporte, que será igual al flujo entrante al sistema por el nodo origen α, y al flujo saliente por el nodo destino ω, sabiendo que cada arco a_{ij} es capaz de transportar un flujo máximo igual a FM_{ij}. Además de dicho valor, nos puede interesar hallar los flujos f_{ij} que circularán por cada uno de los arcos para transportar el flujo total máximo. El conjunto de flujos f_{ij} deberá cumplir en todo momento (no sólo cuando el flujo total sea máximo) las condiciones propias de la red de transporte:

a) La suma de los flujos asociados a los arcos con origen en el nodo origen es igual a la suma de los flujos asociados a los arcos con destino en el nodo destino. Dicha magnitud es igual al flujo transportado por la red.

b) Para el resto de vértices (vértices de transbordo), la suma de flujos asociados a los arcos que inciden en el vértice será igual a la suma de flujos asociados a los arcos que emergen de él.

7.5.2 Corte de flujo mínimo

Un concepto importante asociado a las redes de transporte es el de *corte*. Un corte no es más que cualquier conjunto de arcos cuya supresión deja incomunicados el nodo origen y el destino. El *flujo asociado a un corte* no es más que la suma de los FM_{ij} de los arcos que componen el corte. Estos conceptos nos pueden ayudar a hallar el flujo máximo que circula por una red de transporte, ya que *el flujo total máximo transportado por una red de transporte es igual al mínimo de los flujos asociados a los cortes del grafo*.

Puede observarse este hecho con la red de transporte de la figura 5.2.a. En los arcos se ha indicado el valor de los diferentes FM:

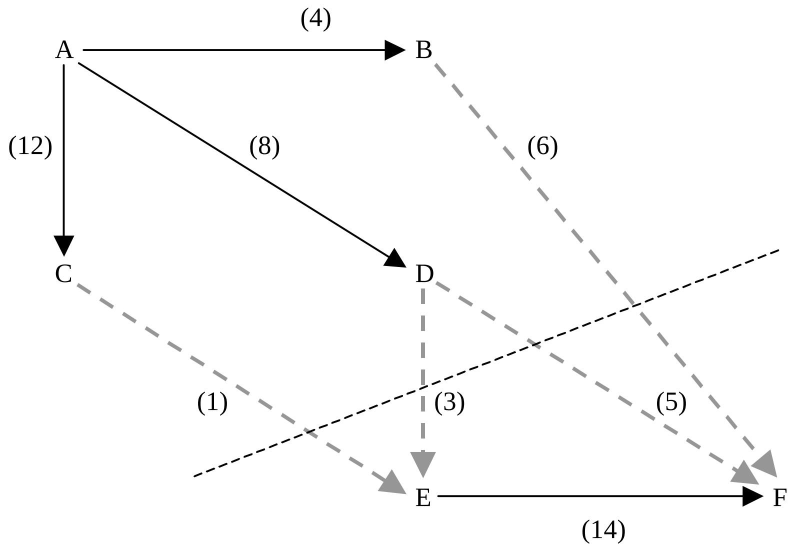


Figura 5.2.a

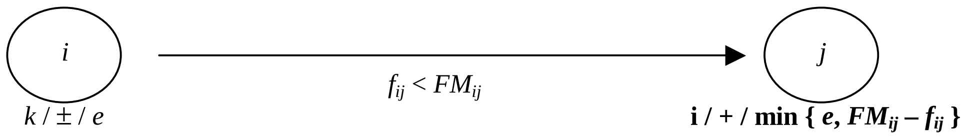
El flujo máximo que puede transportar esta red viene determinado por el corte indicado en la figura. De la suma de los valores asociados al corte, obtenemos que dicho flujo es de 1 + 3 + 5 + 6 = 15. Compruebe el lector que el resto de cortes de la red tienen asociado un valor de flujo mayor.

Para redes de transporte grandes, en las que el corte de flujo asociado mínimo sea difícil de obtener, podemos usar el algoritmo de Ford–Fulkerson. Su estrategia es la de ir vehiculando cantidades adicionales de flujo, a partir de un flujo compatible con la red.

7.5.3 Algoritmo de Ford-Fulkerson

Como se ha indicado, el algoritmo de Ford–Fulkerson permite encontrar el flujo máximo a través de una red de transporte. Para ello, tienen lugar varias iteraciones, en las que se intenta transportar el máximo flujo posible cada vez, hasta que no es posible transportar más flujo. Para ello, se utiliza la marcación de vértices. Existen dos reglas de marcación: la marcación en sentido directo y la marcación en sentido inverso.

Marcación en sentido directo: permite aumentar el flujo del arco, cuando el flujo ya asignado sea inferior a su capacidad. Sólo podemos marcar vértices a los que incidan arcos cuyo vértice origen esté marcado:



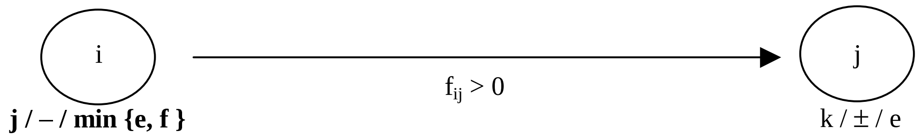
El vértice j ha sido marcado de forma directa. La marca tiene tres componentes:

1. El primero marca el origen del arco por el que se transporta el flujo: en este caso, indicamos que la marcación proviene del vértice i.

2. El segundo muestra que si marca es de sentido directo (como en este caso, con el signo +) o de sentido inverso (tendríamos entonces un signo –).

3. El tercero indica el valor del flujo transportable, determinado por en el caso de la marcación directa por el mínimo de estas dos magnitudes:
   El flujo procedente del vértice *i* (en este caso, un flujo igual a *e*).
   La capacidad del arco a_{ij} de transportar flujo adicional, que será igual a $FM_{ij} - f_{ij}$.

Marcación en sentido inverso: Permite reducir el flujo cuando el flujo asignado al arco es menor que 0. Esta reducción es posible porque el flujo anteriormente asignado a dicho arco se derivará por otra rama de la red. Esta marcación sólo es posible cuando ha sido marcado con anterioridad el vértice destino del arco:



El vértice *i* ha sido marcado de forma inversa. La marca tiene tres componentes:

1. El primero marca el destino del arco por el que se transporta el flujo, en este caso, el vértice *j*.

2. El signo – del segundo componente indica que la marcación es en sentido inverso.

3. El tercer componente indica el valor del flujo que puede desviarse por otro camino, que es igual al mínimo de estos dos componentes:
   El flujo procedente del vértice *j* (en este caso, un flujo igual a *e*).
   El flujo que puede decrementarse del arco a_{ij}, que será precisamente el que circula por él en este momento: f_{ij}.

Una vez determinadas las reglas de marcación, podemos describir el algoritmo de Ford–Fulkerson.

Paso 0	Iniciación: se establece un flujo compatible con la red (por ejemplo, se asigna a todos los arcos un flujo igual a 0, o bien unos valores posibles en una primera aproximación).
Paso 1	Se marca el nodo origen con + / ∞.Todos los vértices pasan a estado no explorado. Mientras haya vértices marcados, hacer: • Tomar un vértice marcado, sea *i* dicho vértice. • A partir de *i*, se intentan marcar todos los vértices adyacentes no explorados siguiendo las reglas de marcación directa e inversa. Los vértices adyacentes que no se puedan marcar pasan a estado bloqueado.[2] • *i* pasa a estado explorado.
Paso 2	Si el nodo destino está marcado, incrementar el flujo en la red a través de la cadena que marcan las etiquetas en una cantidad igual a la asignada a dicho nodo, en el sentido asignado por la marcación. *Ir a paso 1*. Si el vértice destino ω está limpio (no marcado ni explorado) o bloqueado, el flujo es óptimo: finalizar.

[2] La exploración de otro vértice marcado puede desbloquear el vértice. Sólo será vértice bloqueado de forma definitiva si no quedan vértices marcados).

A modo de ejemplo, determinaremos el flujo total máximo que puede circular por la red de transporte siguiente (entre paréntesis, el flujo máximo que puede circular por cada uno de los arcos:

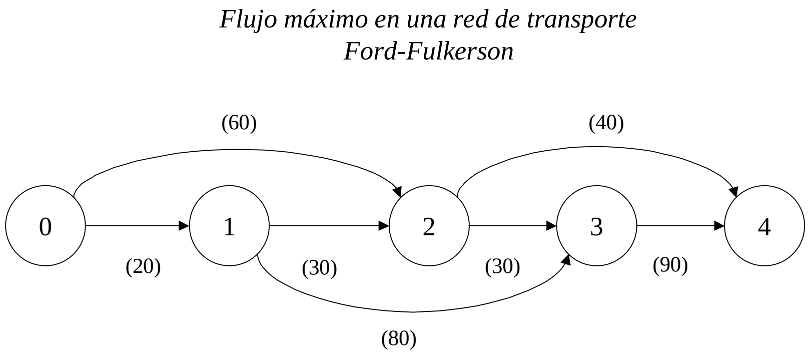


Figura 5.3.a: situación inicial

En primer lugar, se establece un flujo compatible con la red. Dicho flujo puede ser de cero en todas las ramas, o bien unos valores iniciales cercanos al óptimo, como en este caso. Dicho flujo se corresponde con los números que no están entre paréntesis.

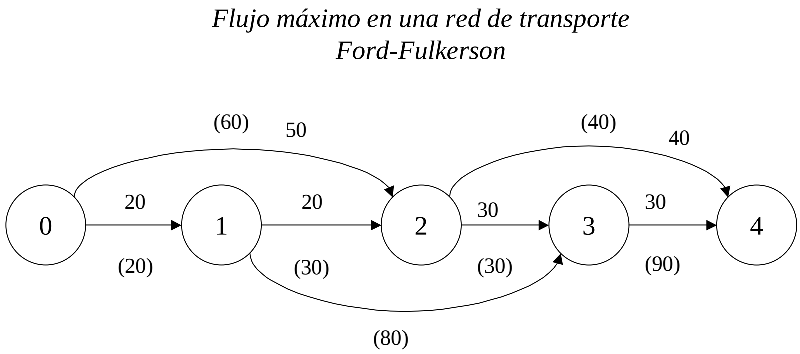


Figura 5.3.b: flujo compatible con la red

A partir de este flujo inicial, se procede a explorar los vértices, y a marcarlos según las reglas establecidas anteriormente. La tabla siguiente muestra la sucesión de exploración de vértices empleada en este caso, indicando si el vértice está *limpio* (l), *bloqueado* (b) o *marcado* en sentido directo (m+) o inverso (m–). Una vez el vértice ha sido *explorado* (e), no vuelve a examinarse de nuevo.

exploración	0	1	2	3	4
0	e	b	m+	l	l
2	e	m–	e	b	l
1	e	e	e	m+	l
3	e	e	e	e	m+

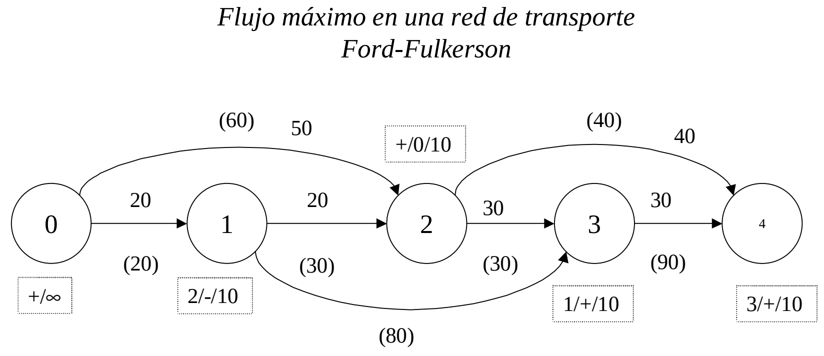


Figura 5.3.c: marcación de los vértices

Como puede observarse en la figura, el vértice destino 4 está marcado con un flujo de 10. Esto significa que podemos incrementar el flujo total en diez, respecto del flujo compatible con la red establecido inicialmente. Para ello, debe seguirse el sentido de las marcaciones, tal como se muestra en 5.3.d:

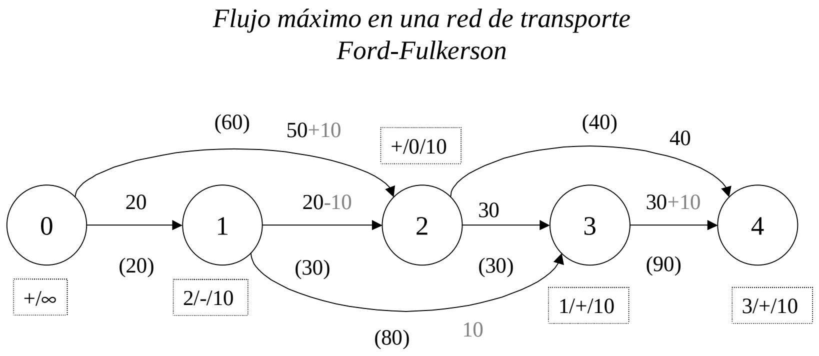


Figura 5.3.d: flujo adicional obtenido

Una vez establecido el nuevo flujo por la red, que ha hecho aumentar el flujo total máximo de 70 a 80, se procede nuevamente a explorar el vértice origen. En esta ocasión, observamos que los vértices 1 y 2 están bloqueados, y 3 y 4 limpios. Esto nos indica que hemos obtenido el flujo total máximo, dado que no es posible explorar 4.

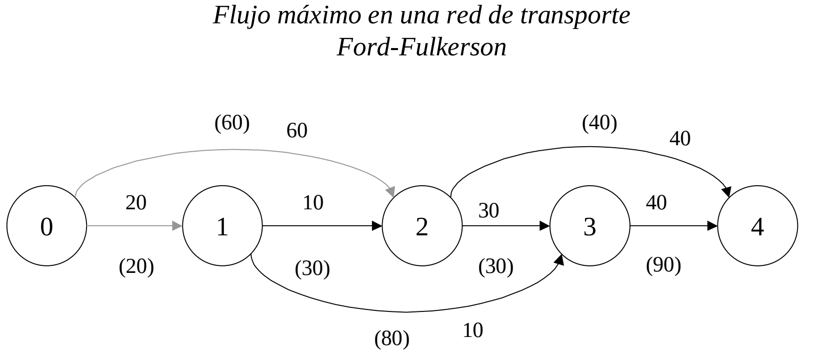


Figura 5.3.e: corte de flujo mínimo

El flujo total máximo resulta ser de 80. El corte de flujo mínimo es el formado por los arcos a_{01} y a_{02}, que precisamente determinan ese flujo máximo. Si aumentamos el flujo máximo en estos arcos, aumentaremos el flujo total. Por el contrario, aumentar el flujo máximo en el resto de arcos no afectará al valor del flujo total máximo.

7.6 Ejercicios

Problema 1

Se le ha encargado diseñar el cableado del sistema de domótica de un edificio inteligente. El edificio cuenta con cinco terminales con aparatos controlables a través de una central. No es necesario que la conexión de un terminal con la central sea directa: puede realizarse indirectamente a través de conexiones con otras terminales. El coste de conectar las diferentes terminales entre sí es el que se indica en la tabla.

	Matriz	*T1*	*T2*	*T3*	*T4*	*T5*
Matriz	-	210	160	120	250	190
T1	210	-	110	230	220	55
T2	160	110	-	180	130	210
T3	120	230	180	-	185	280
T4	250	220	130	185	-	330
T5	190	55	210	280	330	-

Se trata de determinar qué terminales deben conectarse entre sí, de manera que todas ellas estén conectadas (directa o indirectamente) con la central a coste mínimo.

Problema 2

Se le ha encomendado la tarea de planificar para los próximos cuatro años (X1, X2, X3 y X4) la explotación de una finca maderera. En esencia, la explotación consiste en lo siguiente:

A final de año, debe decidirse si se desea talar el bosque y vender la madera. La cantidad de madera será función del tiempo que hace que se plantó el bosque, según la tabla siguiente:

Años desde que se plantó	Cantidad de madera (m^3)
1 año	100
2 años	150
3 años	200
4 años	220

El coste de talar el bosque (fundamentalmente mano de obra) aumenta con el tiempo, y también lo hace el precio del m3 de madera, según la siguiente tabla:

Fecha en que se tala	Coste tala (um)	Precio madera (um/m^3)
31/12/X1	50	1,5
31/12/X2	70	2
31/12/X3	90	2
31/12/X4	110	2,5

Si el bosque se ha talado, debe replantarse el bosque, a principios del año siguiente. Por las condiciones contractuales establecidas con el propietario, a finales del año X4 el bosque debe estar talado, porque dicho propietario desea replantarlo a principios de X5. Los costes de plantar el bosque se indican en la tabla adjunta:

Fecha en que se planta el bosque	Coste de plantación (um)
1/1/X1	20
1/1/X2	22
1/1/X3	25
1/1/X4	27

Sabiendo que a principios de X1 debemos plantar la explotación, determinar en qué años hemos de talar el bosque para replantarlo después y obtener la máxima ganancia.

Problema 3

El grafo que se presenta a continuación representa las conexiones de agua existentes entre los diferentes almacenes de una determinada empresa. Sobre cada arco, se indica la capacidad máxima de transporte de agua (en m^3/hora) de la tubería o tuberías que conectan los almacenes.

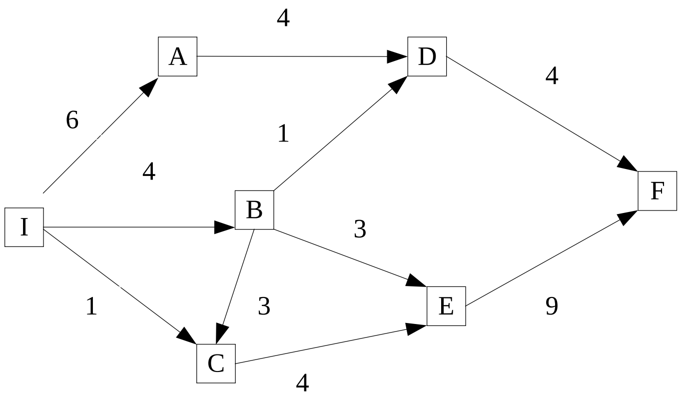
Se pide:

a) ¿ Cuál será la máxima cantidad de agua que puede recogerse en el almacén F?.

b) Determinar qué flujo debe circular por cada una de las tuberías para obtener el flujo máximo

c) Formular el problema lineal que permitiría resolver este problema. Asimismo, dar la expresión de manera que pueda ser resuelto por el método SÍMPLEX (variables de holgura y artificiales). Describir el significado de las variables de holgura.

d) ¿Cuáles serán las variables que formen parte de la solución básica en el óptimo de este programa lineal?



Problema 4

Cinco camiones de carga tienen que entregar siete tipos de paquetes. Hay tres paquetes de cada tipo, y las capacidades de los cinco camiones son 6, 4, 5, 4 y 3 paquetes respectivamente. Determinar si se pueden cargar los paquetes de tal manera que ningún camión cargue dos paquetes del mismo tipo.

Problema 5

Hay cuatro trabajadores disponibles para realizar los trabajos 1, 2, 3 y 4. Tres trabajadores pueden realizar solamente ciertos trabajos:

a) el trabajador 1 y el trabajador 2 solamente los trabajos 1 y 2;

b) el trabajador 3, solamente el trabajo 2;

c) el trabajador 4, cualquier trabajo.

Determinar si se pueden asignar todos los trabajos a un trabajador idóneo.

7.7 Glosario de términos

Árbol

Grafo conexo sin ciclos. Un subconjunto de arcos o aristas de un grafo que sea un árbol es un árbol parcial de dicho grafo.

Arco

Representación de la relación entre dos vértices de un grafo, cuando es relevante conocer el origen y el destino de la relación: de otro modo, además de la dirección de la relación, debemos representar el sentido. Los arcos se encuentran en los grafos orientados.

Arista

Representación de la relación entre dos vértices de un grafo, en la que no se especifica el origen y el destino de la relación. Las aristas aparecen en grafos no orientados.

Arborescencia

Grafo orientado, fuertemente conexo, sin ciclos ni bucles, en que todos los vértices tendrán semigrado interior igual a la unidad, excepto uno, raíz de la arborescencia, cuyo semigrado interior es 0. Es útil para representar procesos decisionales.

Bucle

Arco o arista cuyos vértices coinciden. Representa una conexión directa de un vértice consigo mismo.

Cadena

Camino no orientado, esto es, sucesión de aristas tal que el vértice extremo de cada una (exceptuando la última) coincide con el vértice extremo de la siguiente en la sucesión. Dos vértices de un grafo no orientado que no están conectados directamente con una arista pueden estar conectados indirectamente a través de una cadena.

Camino

Sucesión de arcos tal que el vértice extremo de cada uno (exceptuando el último) coincide con el vértice extremo del siguiente en la sucesión. Dos vértices de un grafo orientado que no estén conectados directamente con una arista pueden estarlo indirectamente por un camino.

Ciclo

Cadena que se inicia y termina en el mismo vértice. Por extensión, en un grafo no orientado se define como un conjunto de arcos que unen una serie de vértices, prescindiendo de su orientación.

Circuito

Camino que se inicia y termina en el mismo vértice. Representa una conexión indirecta de un vértice consigo mismo en un grafo orientado. Se trata de un concepto más fuerte que el de ciclo: todos los circuitos son ciclos, pero no todos los ciclos son circuitos.

Conexo (grafo)

Grafo en que existe al menos una cadena entre toda pareja de vértices. El concepto es aplicable tanto a grafos orientados como para no orientados.

Destino,m vértice

Ver *semigrado exterior*.

Distancia

Valor numérico asociado a un arco o una arista de un grafo que representa posibilidades de comunicación. La distancia de un camino es igual a la suma de las distancias de los arcos que componen dicho camino. Según la situación que queramos representar, los valores de distancia de los arcos pueden representar tiempo, coste u otros conceptos, en vez de distancia.

Flujo

Magnitud asociada a un arco, que representa la cantidad de determinada variable vehiculada a través de dicho arco por unidad de tiempo. Además del significado más obvio de caudal de fluido, puede representar en ciertas situaciones magnitudes como productividad (producción por unidad de tiempo). El valor total de los flujos que llegan a un vértice ha de ser igual al flujo neto establecido para dicho vértice.

Flujo total

Cantidad máxima de flujo que puede vehicular una red de transporte. Es igual al flujo que emerge de los vértices origen y al que incide en los vértices destino.

Fuertemente conexo (grafo)

Grafo en que existe al menos un camino entre toda pareja de vértices. Es una propiedad más fuerte que la de grafo conexo, dado que todo grafo fuertemente conexo es conexo.

Grado

Es el número total de arcos que tienen origen o destino en el vértice. Es igual a la suma del semigrado interior y el semigrado exterior.

Grafo

Representación de las relaciones existentes entre los elementos de un sistema. Los elementos se representan por los vértices del grafo, y las relaciones por arcos o aristas.

Grafo orientado

Representación de las relaciones existentes entre los elementos de un sistema, cuando la relación entre dos elementos *i* y *j* no tiene porqué ser la misma que la existente entre *j* e *i*. En los grafos orientados, debemos distinguir entre el origen y el destino de la relación que establezcamos.

Grafo no orientado

Representación de las relaciones existentes entre los elementos de un sistema, cuando la relación entre dos elementos *i* y *j* es siempre la misma que la existente entre *j* e *i*.

Origen, vértice

Ver *semigrado interior*.

Red de transporte

Grafo que representa las posibilidades de vehicular flujos a través de un conjunto de vértices. Cuando es posible vehicular flujo entre dos vértices, tendremos un arco al que usualmente asociaremos un valor máximo de flujo.

Semigrado exterior de un vértice

Número de arcos que emergen de un vértice. Un vértice de una red de transporte con semigrado exterior igual a cero es un destino de los flujos de dicha red.

Semigrado interior de un vértice

Número de arcos que inciden en un vértice. Un vértice de una red de transporte con semigrado interior igual a cero es un origen de los flujos de dicha red.

Vértice

Representación de los elementos del sistema cuyas relaciones vienen representadas en el grafo.

Modelo de evaluaciones

Problema 1

Una empresa de productos químicos elabora dos compuestos, denominados P1 y P2, a partir de dos principios activos M y N. Además de los principios activos, es necesario utilizar un determinado tiempo de reactor para producirlos. La cantidad de recursos necesaria para elaborar un Kg de P1 y P2 es la que se indica en la tabla adjunta:

	M (Kg)	N (Kg)	Tpo. reactor (h.)
P1	4	3	3
P2	6	9	2

Los costes de los recursos, así como la cantidad disponible, son:

	Coste	*Cantidad*
Principio M	4 euros/Kg	10.000 Kg.
Principio N	3 euros/Kg	14.000 Kg.
Reactor	1 euro/hora	7.000 horas

La empresa puede vender tanto P1 como P2, como los principios M y N, directamente. Los precios de venta son:

Precio de P1: 69 euros/Kg
Precio de P2: 57 euros/Kg
Precio de M: 4,5 euros/Kg
Precio de N: 3,2 euros/Kg

Con estos datos, se pide:

1. Definir el modelo lineal para obtener el máximo beneficio, indicando:
   - Definición de las variables.
   - Función objetivo.
   - Restricciones.

2. Restricciones a añadir al problema original si debemos vender uno y sólo de los dos productos (P1 o P2), y toda la cantidad que deseemos de M y N.

3. Restricciones a añadir al problema original si sólo podemos vender tres de los cuatro productos.

4. Modificaciones del problema original para considerar el efecto de un coste fijo, de 6.000 para P1 y de 4.000 para P2, en el que se incurre sólo con que vendamos una unidad de producto. No incurriremos en él, en consecuencia, si decidimos no vender ese producto.

Solución problema 1

1. *Definir el modelo lineal para obtener el máximo beneficio, indicando:*
   - *Definición de las variables.*
   - *Función objetivo.*
   - *Restricciones.* **(4 puntos)**

El modelo puede obtenerse a partir de las variables siguientes:

P1: cantidad a vender de P1
P2: cantidad de vender de P2
M: cantidad a vender de producto M
N: cantidad a vender de producto N

El margen obtenido con cada producto es:

Precio producto – coste de M – coste de N – horas de trabajo

P1 = 69 – 4· 4 – 3· 3 – 3· 1 = 41
P2 = 57 – 6· 4 – 9· 3 – 2· 1 = 4
M = 4,5 – 4 = 0,5
N = 3,2 – 3 = 0,2

De modo que la función objetivo resulta ser:

$$[MAX]\ z = 41P1 + 4P2 + 0{,}5M + 0{,}2N$$

En cuanto a las restricciones, expresarán las limitaciones en los recursos:

Cantidad de M)	$M + 4P1 + 6P2 \leq 10.000$
Cantidad de N)	$N + 3P1 + 9P2 \leq 14.000$
Cantidad de horas)	$3P1 + 2P2 \leq 7.000$
No – negatividad)	$P1, P2, M, N \geq 0$

Pueden usarse más variables (por ejemplo, cantidad de M y N utilizadas para producir P1 y P2) que dan lugar a restricciones de igualdad, del tipo siguiente:

	$M_{P1} = 4P1$
	$M_{P2} = 6P2$
	$N_{P1} = 3P1$
	$N_{P2} = 9P2$
Cantidad de M)	$M + M_{P1} + M_{P2} \leq 10.000$
Cantidad de N)	$N + N_{P1} + N_{P2} \leq 14.000$
Cantidad de horas)	$3P1 + 2P2 \leq 7.000$
No – negatividad)	$P1, P2, M, N \geq 0$

2. *Restricciones a añadir al problema original si debemos vender uno y sólo de los dos productos (P1 o P2), y toda la cantidad que deseemos de M y N.* **(2 puntos)**

Debemos añadir una variable binaria B0, tal que:

$B_0 = 1$ se vende P1
$B_0 = 0$ se vende P2

Entonces deben añadirse las restricciones:

$$P1 \leq K\cdot B_0$$
$$P2 \leq K\cdot (1- B_0)$$

3. *Restricciones a añadir al problema original si sólo podemos vender tres de los cuatro productos.* **(2 puntos)**

Ahora deben introducirse cuatro variables binarias, una para cada producto, del tipo:

$B_i = 1$ se vende el producto i
$B_i = 0$ no se vende el producto i
Para *i* = P1, P2, M, N

Entonces las restricciones son:

$$P1 \leq K\cdot B_{P1}$$
$$P2 \leq K\cdot B_{P2}$$
$$M \leq K\cdot B_M$$
$$N \leq K\cdot B_N$$
$$B_{P1} + B_{P2} + B_M + B_N = 3$$

4. *Modificaciones del problema original para considerar el efecto de un coste fijo, de 6.000 para P1 y de 4.000 para P2, en el que se incurre al vender una unidad de producto. No incurriremos en él, en consecuencia, si decidimos no vender ese producto.* **(2 puntos)**

Ahora emplearemos BP1 y BP2 del mismo modo que se emplearon en el caso anterior.

La función objetivo tiene ahora la expresión:

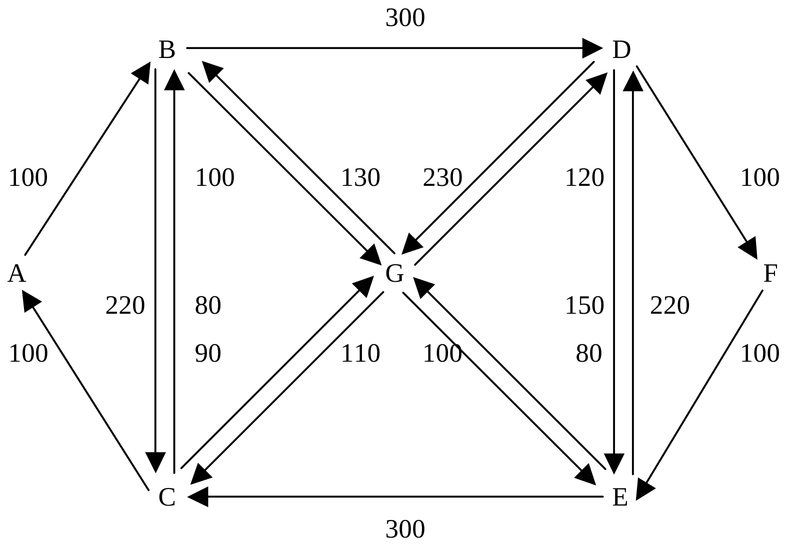
$$[MAX]\ z = 41P1 + 4P2 + 0{,}5M + 0{,}2N – 6.000B_{P1} – 4.000B_{P2}$$

Y deben incluirse las restricciones:

$$P1 \leq K\cdot B_{P1}$$
$$P2 \leq K\cdot B_{P2}$$

Problema 2

El esquema siguiente representa el flujo de circulación viaria de una ciudad, que cuenta con una ronda de circunvalación (nodos A hasta F), desde la cual puede irse al centro (G). Los valores asociados a los arcos representan el tiempo que se tarda en ir de un nodo a otro en hora punta.



Tenemos un cuartel de bomberos en el nodo A e interesa determinar la ruta más rápida desde el cuartel a cada uno de los nodos del grafo en hora punta.

1. Determinar, mediante el algoritmo de Bellman – Kalaba (mejorado), la ruta más rápida desde A al resto de nodos, así como el tiempo que se tarda en llegar a cada uno de ellos.

2. Supongamos ahora que tenemos un coche de bomberos en G. Determinar, mediante el algoritmo de Dijkstra la ruta más rápida desde ese punto G hasta el nodo F.

3. Definir la naturaleza de las variables a considerar, la expresión de la función objetivo y la naturaleza de las restricciones para encontrar el camino más corto desde A hasta F mediante programación lineal.

Solución problema 2

1. *Determinar, mediante el algoritmo de Bellman – Kalaba (mejorado), la ruta más rápida desde A al resto de nodos, así como el tiempo que se tarda en llegar a cada uno de ellos.* **(4 puntos)**

Los resultados de la aplicación del algoritmo son:

	K = 0	K = 1	K = 2	K = 3
A	0 (A)	0 (A)	0 (A)	0 (A)
B	100 (B)	100 (B)	100 (B)	100 (B)
C		320 (B)	310 (G)	310 (G)
D		400 (B)	320(G)	320 (G)
E		550 (D)	300 (G)	300 (G)
F		500 (D)	420 (D)	420 (D)
G		200 (B)	200 (B)	200 (B)

Los caminos más cortos desde A al resto de nodos son:

Nodo	*Camino*	*Distancia*
A	A	0
B	A – B	100
C	A – B – G – C	310
D	A – B – G – D	320
E	A – B – G – E	300
F	A – B – G – D – F	420
G	A – B – G	200

2. *Supongamos ahora que tenemos un coche de bomberos en G. Determinar, mediante el algoritmo de Dijkstra, la ruta más rápida desde ese punto G hasta el nodo F.* **(4 puntos)**

La exploración por Dijkstra es, en este caso, menos eficiente que el examen ocular. Sin embargo, se pide encontrar la distancia *siguiendo este algoritmo*, por lo que tenemos:

A				210 (C)	210 (C)	210 (C)
B		130 (G)	130 (G)	130 (G)	**130 (G)**	130 (G)
C		110 (G)	**110 (G)**	110 (G)	110 (G)	110 (G)
D		120 (G)	120 (G)	**120 (G)**	120 (G)	120 (G)
E		**100 (G)**	100 (G)	100 (G)	100 (G)	100 (G)
F					220 (D)	220 (D)
G	**0 (G)**	0 (G)	0 (G)	0 (G)	0 (G)	0 (G)
Z*	G	E	C	D	B	A

Y en el siguiente paso entrará F, sin verse modificado el camino. El camino más corto vale 220, y pasa por G – D – F.

3. *Definir la naturaleza de las variables a considerar, la expresión de la función objetivo y la naturaleza de las restricciones para encontrar el camino más corto desde A hasta F mediante programación lineal.* **(2 puntos)**

Las variables son de tipo binario:

Xij = 1 el arco pertenece al camino más corto
Xij = 0 el arco no pertenece al camino más corto

En cuanto a los parámetros, son de la forma:

Cij = distancia entre *i* y *j* (infinito si no hay conexión directa).

Con estos datos, el problema es de flujo de coste mínimo, con un flujo entrante y saliente igual a 1.

Problema 3

Este ejercicio consiste en un análisis de sensibilidad del problema PP1 del módulo I de entregas. A continuación se detalla:

a) el enunciado del problema original

b) la solución (incluyendo análisis de sensibilidad) con el programa LINDO *c*) un conjunto de preguntas a contestar acerca del análisis de sensibilidad.

Problema PP1

Una compañía produce cuatro tipos diferentes de piezas metálicas que deben maquinarse, pulirse y ensamblarse. Las necesidades específicas de tiempo (en horas) para cada producto son las siguientes:

	Maquinado	Pulido	Ensamblado
Producto I	3	1	2
Producto II	2	1	1
Producto III	2	2	2
Producto IV	4	3	1

La compañía dispone semanalmente de 480 horas para el maquinado, 400 para el pulido y 400 para ensamble. Las ganancias unitarias por producto son de 6$, 4$, 6$ y 8$ respectivamente. La compañía se compromete a entregar semanalmente 50 unidades del producto I y 100 unidades de cualquiera de los productos I, II y III, según sea la producción, pero sólo un máximo de 25 unidades del producto IV. ¿Cuántas unidades de cada producto deberá fabricar semanalmente la compañía a fin de cumplir con todas las condiciones del contrato y maximizar la ganancia total? Considérese que las piezas incompletas puede terminarse la siguiente semana.

Solución (programa LINDO)

Denominando P1, P2, P3 y P4 a la cantidad necesaria de cada producto, tenemos el siguiente modelo. Las tres primeras restricciones representan las limitaciones de recursos (horas de maquinado, pulido y ensamblado) y las otras tres las condiciones contractuales asociadas a la demanda.

```
MAX 6P1 + 4P2 + 6P3 + 8P4
ST
MAQ) 3P1 + 2P2 + 2P3 + 4P4 < 480
PUL) P1 + P2 + 2P3 + 3P4 < 400
ENS) 2P1 + P2 + 2P3 + P4 < 400
D1) P1 > 50
D2) P1 + P2 + P3 > 100
D3) P4 < 25
END
```

La solución del problema, obtenida por LINDO, es la indicada a continuación:

```
OPTIMO ENCONTRADO EN PASO      4

        VALOR DE LA FUNCIÓN OBJETIVO

        1)      1250.000

  VARIABLE        VALOR        COSTE REDUCIDO
        P1        50.000000          0.000000
        P2         0.000000          0.666667
        P3       145.000000          0.000000
        P4        10.000000          0.000000
```

```
      FILA   HOLGURA O EXCESO  PRECIOS DUALES
      MAQ)         0.000000        1.666667
      PUL)        30.000000        0.000000
      ENS)         0.000000        1.333333
       D1)         0.000000       -1.666667
       D2)        95.000000        0.000000
       D3)        15.000000        0.000000

 NO. ITERACIONES=       4


 RANGOS DE VALORES PARA LOS QUE NO CAMBIA LA BASE:

                          RANGOS COEF. FUNC. OBJETIVO
 VARIABLE         ACTUAL         AUMENTO        DECREMENTO
                   COEF          PERMITID0       PERMITIDO
       P1        6.000000         1.666667         INFINITO
       P2        4.000000         0.666667         INFINITO
       P3        6.000000         9.999999         2.000000
       P4        8.000000         4.000000         2.000000

                          RANGOS TÉRMINOS INDEPENDIENTES
      FILA        ACTUAL         AUMENTO        DECREMENTO
                    TIN          PERMITID0       PERMITIDO
      MAQ      480.000000        45.000000        30.000000
      PUL      400.000000         INFINIT0        30.000000
      ENS      400.000000        30.000000        45.000000
       D1       50.000000        30.000000        18.000000
       D2      100.000000        95.000000         INFINITO
       D3       25.000000         INFINITO        15.000000
```

Preguntas (análisis sensibilidad)

Suponga que es el jefe de planificación de la compañía, y que su jefe le hace las preguntas siguientes. Contéstelas, razonando brevemente su respuesta, en *un máximo de dos páginas*.

1. ¿Qué sucederá con el valor de la función objetivo, y con la cantidad vendida de cada uno de los productos, si aumentamos en 1$ el precio de venta de P1?

2. Supongamos que podemos aumentar, sin coste alguno, las horas disponibles de uno solo de los procesos (maquinado, pulido o ensamblado). ¿Cuál de los tres nos interesará más aumentar?

3. Actualmente nos vemos obligados a cumplir una serie de condiciones contractuales respecto de la demanda. ¿Cuál de ellas puede interesarnos suprimir, si se mantienen las condiciones actuales?

4. Si ahora, en vez de 480 horas de maquinado, contamos sólo con 460, ¿me interesará producir P2? ¿Variará la cantidad vendida de los otros productos en el óptimo?

5. Si nos ofrecen aumentar las horas de ensamblado, pero esta vez pagando por ello. ¿Cuál será el precio máximo que estaríamos dispuestos a pagar por una hora adicional de ensamblado?

Solución

1. *¿Qué sucederá con el valor de la función objetivo, y con la cantidad vendida de cada uno de los productos, si aumentamos en 1$ el precio de venta de P1?* **(2 puntos)**

Como el incremento está por debajo de máximo a partir del cual cambia la base, se mantiene la base óptima. El valor de las variables en el óptimo no cambia, pero la función objetivo aumenta en 50, porque ha aumentado en 1 un coeficiente de una variable básica cuyo valor es 50.

2. *Supongamos que podemos aumentar, sin coste alguno, las horas disponibles de uno solo de los procesos (maquinado, pulido o ensamblado). ¿Cuál de los tres nos interesará más aumentar?* **(2 puntos)**

No conviene aumentar el pulido, puesto que nos sobran horas de ese recurso (holgura de 30). Puede resultar ventajoso aumentar las horas de maquinado y ensamblado. Puestos a escoger una, es mejor escoger el *maquinado*, porque tiene un mayor precio dual (un aumento en una unidad de ese recurso asegura un mayor valor de la función objetivo).

3. *Actualmente nos vemos obligados a cumplir una serie de condiciones contractuales respecto de la demanda. ¿Cuál de ellas puede interesarnos suprimir, si se mantienen las condiciones actuales?* **(2 puntos)**

La condición D1: el precio sombra negativo nos indica que si reducimos la cantidad mínima de P1, aumentará el valor de la función objetivo.

4. *Si ahora, en vez de 480 horas de maquinado, contamos sólo con 460, ¿me interesará producir P2? ¿Variará la cantidad vendida de los otros productos en el óptimo?* **(2 puntos)**

La variación del término independiente está dentro del rango de valores en los que se mantiene la base óptima. En esa base P2 es no básica, por lo que seguiremos sin producir P2.

Por otra parte, aunque la base se mantenga, cambian los valores de las variables básicas (recuérdese que son iguales a $B^{-1}b$, y que b ha cambiado).

5. *Si nos ofrecen aumentar las horas de ensamblado, pero esta vez pagando por ello. ¿Cuál será el precio máximo que estaríamos dispuestos a pagar por una hora adicional de ensamblado?* **(2 puntos)**

Del examen del precio dual de la restricción de ensamblado, observamos que el aumento en una hora de ese recurso hace que la función objetivo aumente en 1,33$. En consecuencia, ése será el precio máximo a pagar por el recurso.