Cátedra Nissan

-PROTHIUS-

Métodos Cuantitativos de Organización Industrial: Teoría de Grafos.

Joaquín Bautista Valhondo

D-03/2011
(Rec. IO-JBV)

Departamento de Organización de Empresas

Universidad Politécnica de Cataluña

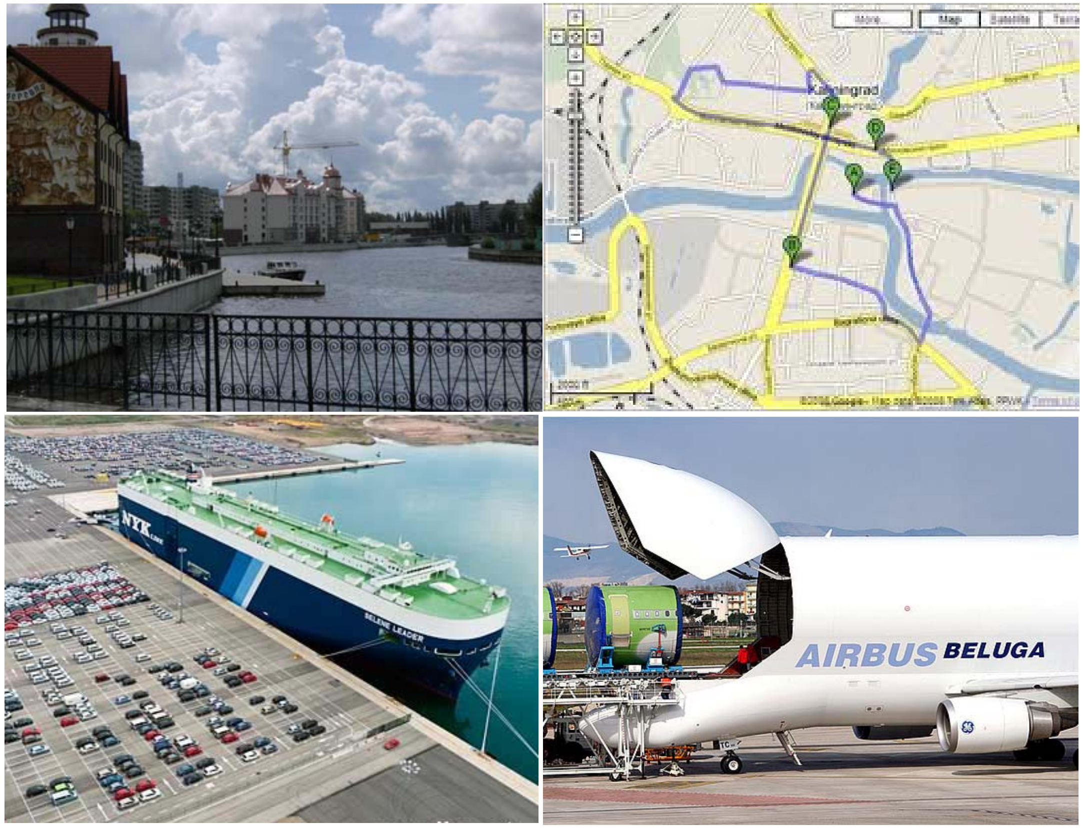
Publica:

Universitat Politècnica de Catalunya
www.upc.edu

Edita:

Cátedra Nissan
www.nissanchair.com
director@nissanchair.com

Teoría de grafos



NISSAN
CATEDRA NISSAN

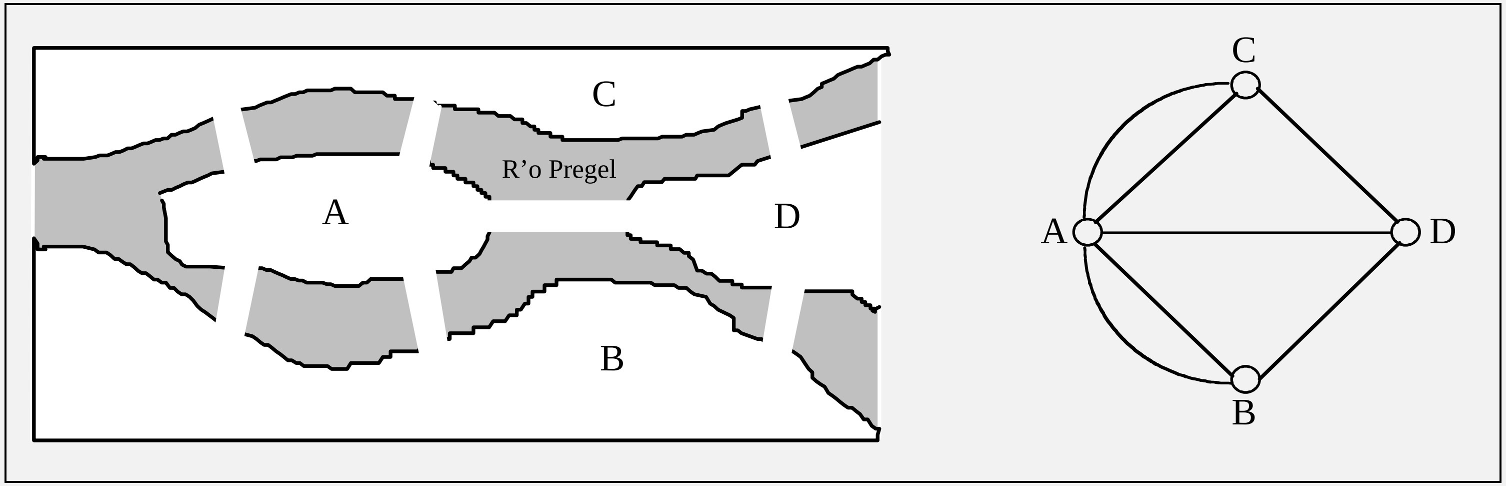
DOE
Departament
d'Organització
d'Empreses

Contenido

- *Preliminares*
- *Concepto de grafo*
- *Formas de representación*
- *Terminología*
- *Grafos como soporte de modelos*
- *Problemas de optimización en grafos*
- *Árbol parcial mínimo*
- *Caminos extremos entre vértices*
- *Flujos óptimos en redes*

Sus orígenes se remontan al menos a 1736: **Leonard Euler** (1707-1783) resolvió de manera ingeniosa el problema de **los puentes de Königsberg**, el cual se expone continuación:

En la ciudad de Königsberg había en el siglo XVIII siete puentes que cruzaban el río **Pregel**; uno de ellos conectaba dos islas (A y D) entre sí, mientras que los otros seis servían de conexión entre las islas y las dos orillas del río (C y B), tal como se muestra en la figura.



La cuestión era la siguiente: ¿es posible realizar un recorrido cruzando todos los puentes una sola vez? o lo que es lo mismo: ¿es posible dibujar de un solo trazo el esquema (grafo) presentado en la figura?

Respuesta: NO.

CONCEPTO DE GRAFO

Un grafo G consta de:

- Un conjunto de elementos, X, denominados nudos o vértices.
- Una lista de parejas de vértices que expresa las relaciones entre los elementos de X.

  - Aristas, grafo no orientado. Se designa por E.
  - Arcos , grafo orientado. Se designa por U.
  - El número de vértices distintos se denomina orden del grafo.
  - Si **p** es el número máximo de veces que aparece un par: **p-grafo**.

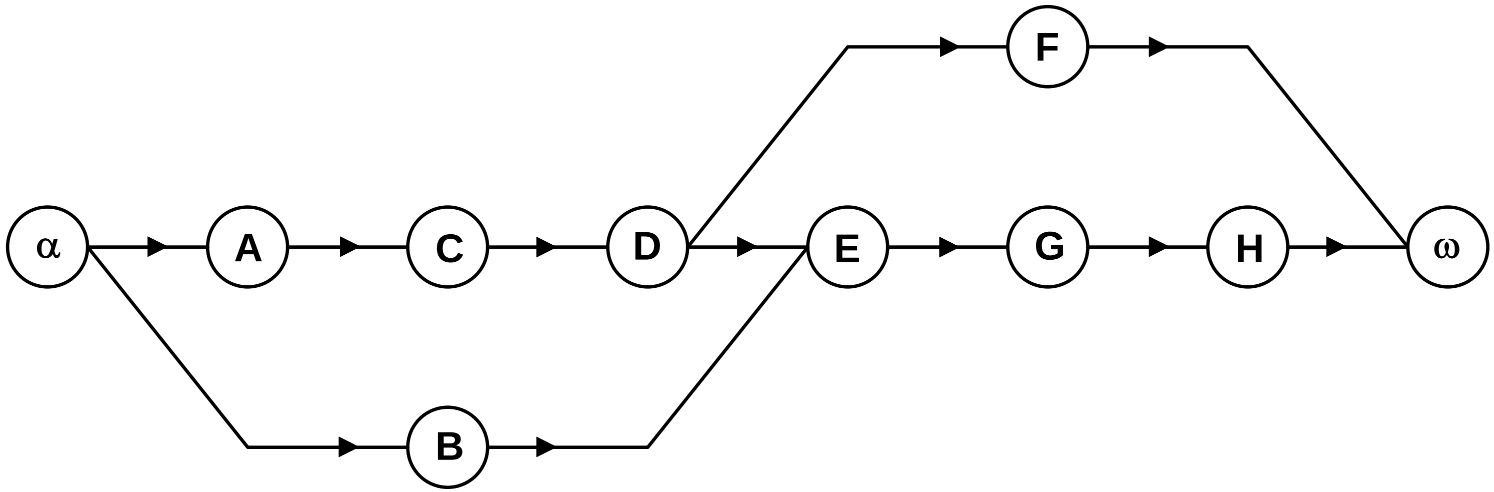
(p.e. el grafo de los puentes de Königsberg es un 2-grafo no orientado de orden 4).

Los grafos son un instrumento para modelizar sistemas:

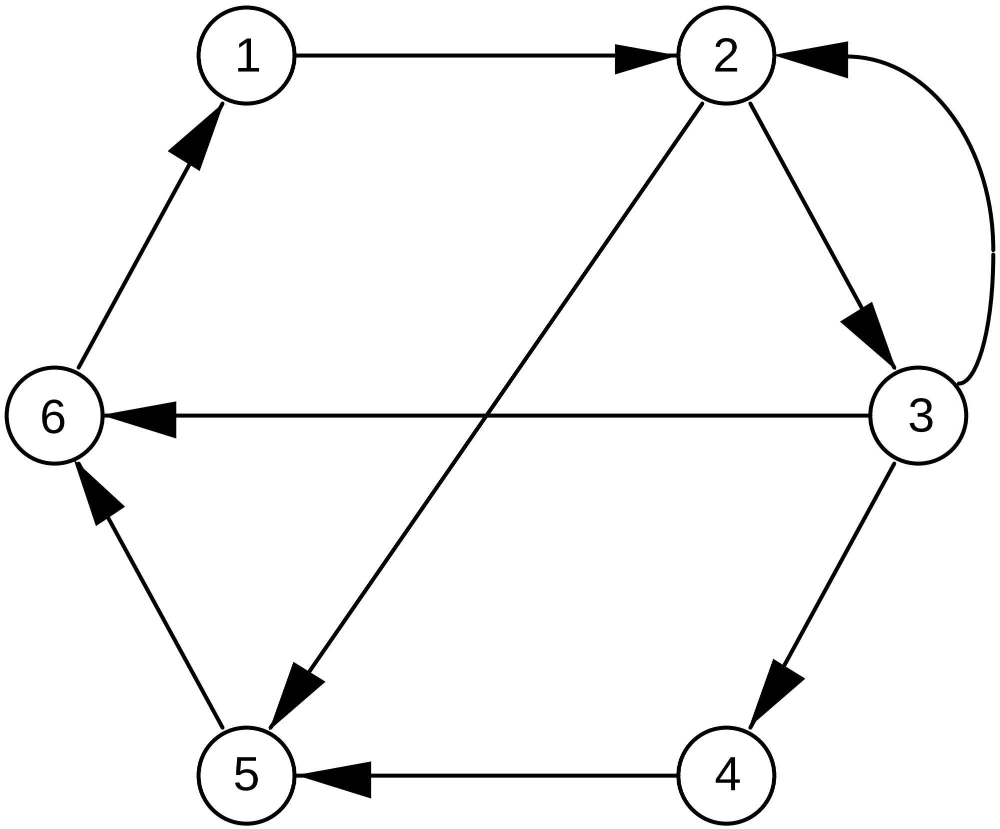
- Redes de comunicaciones: calles de una ciudad, carreteras, oleoductos, gaseoductos, redes eléctricas, redes de información, rutas de productos, procesos, pasillos en almacenes, etc.
- Relaciones de orden: jerarquías, estructura de un producto, organigramas, etc.
- Evolución de sistemas que cambian de estado: Sistemas con espera, envejecimiento y renovación de equipos, etc.

Cuando el grafo es orientado, los arcos se representan mediante flechas que emergen del primer elemento del par, denominado extremo inicial u origen, e inciden en el segundo elemento del par que se denomina extremo final.

Ejemplo-00: Esquema de las operaciones que hacen falta para cambiar la rueda de un coche:



1	α	Inicio	6	E	Poner rueda
2	A	Coger gato	7	F	Guardar rueda
3	B	Preparar rueda de recambio	8	G	Bajar coche
4	C	Levantar coche	9	H	Guardar gato
5	D	Quitar rueda averiada	10	ω	Fin



Ejemplo-01: Una compañia de autobuses realiza regularmente transporte entre 6 ciudades, que numeraremos de 1 a 6. Las líneas directas son las siguientes: L1 desde la ciudad 1 hasta la 2, L2 de 2 a 3, L3 de 2 a 5, L4 de 3 a 2, L5 de 3 a 4, L6 de 3 a 6, L7 de 4 a 5, L8 de 5 a 6 y L9 de 6 a 1.

El grafo asociado al ejemplo-01, G=(X,U), está formado por los conjuntos:

X=(1,2,3,4,5,6) de vértices.
U= [(1,2), (2,3), (2,5), (3,2), (3,4), (3,6), (4,5), (5,6), (6,1)] de arcos.

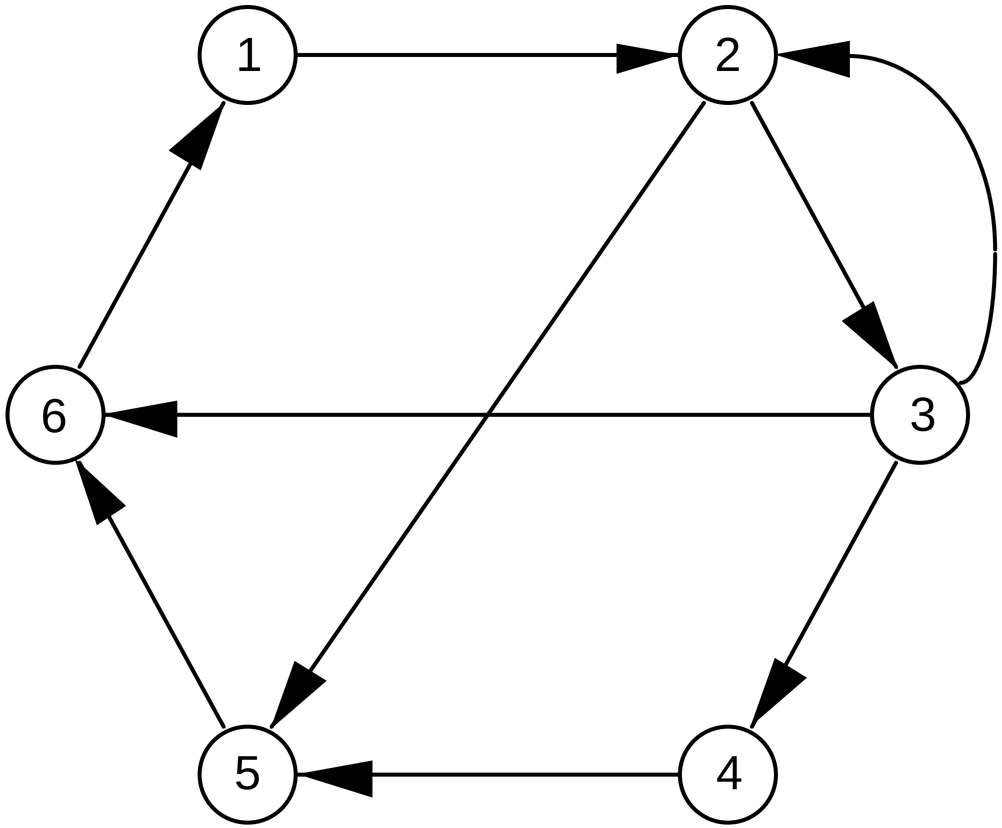
OTRAS FORMAS DE REPRESENTACIÓN

Representaciones numéricas:

- Matriz de incidencias.
- Diccionario del grafo.
- Diccionario inverso.
- Lista de arcos.
- Lista condensada de arcos.

Ilustramos dichas representaciones mediante el ejemplo-01.

MATRIZ DE INCIDENCIAS

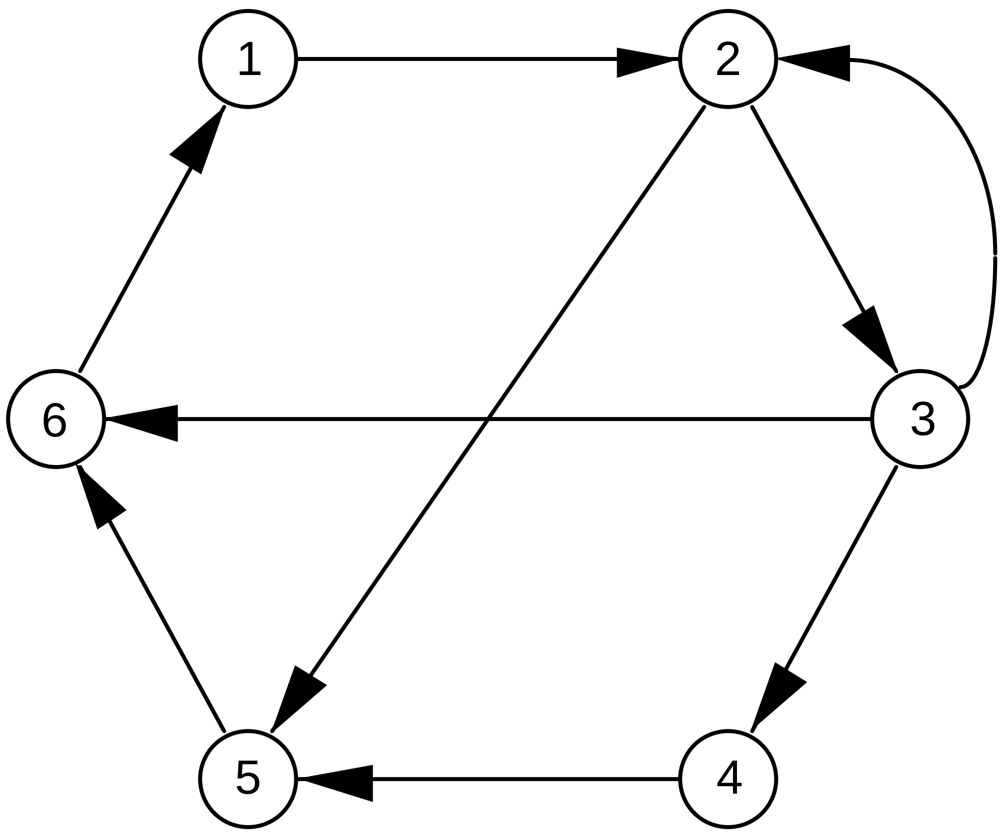


Indica el número de arcos que emergen de los vértices localizados por filas incidiendo en los vértices localizados en las columnas.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	1	0	1	0	1
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	1	0	0	0	0	0

Esta forma de representación require una capacidad para almacenar la información igual al cuadrado del número de vértices (nxn).

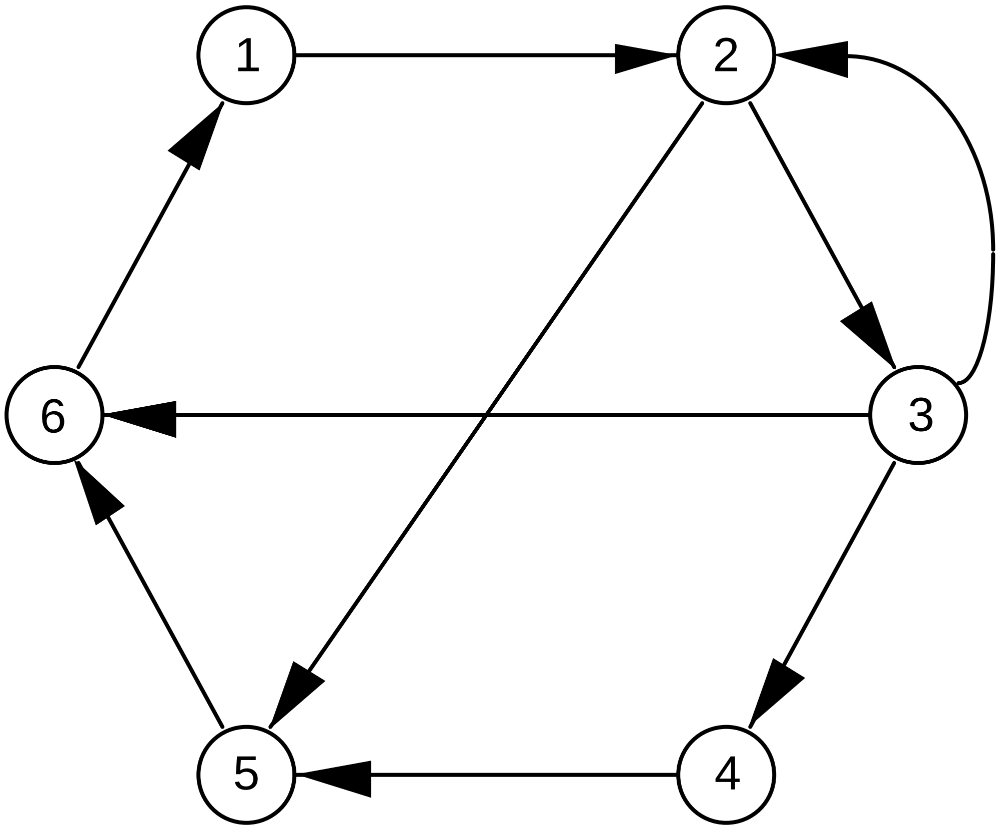
DICCIONARIO DEL GRAFO



A cada vértice extremo origen se asocian todos los vértices extremos destino. Para el ejemplo-01, se tiene:

	1º	**2º**	**3º**
1	2		
2	3	5	
3	2	4	6
4	5		
5	6		
6	1		

Esta forma de representación requiere una capacidad para almacenar la información igual producto del número de vértices por el número máximo de arcos que emergen de cualquiera de ellos.

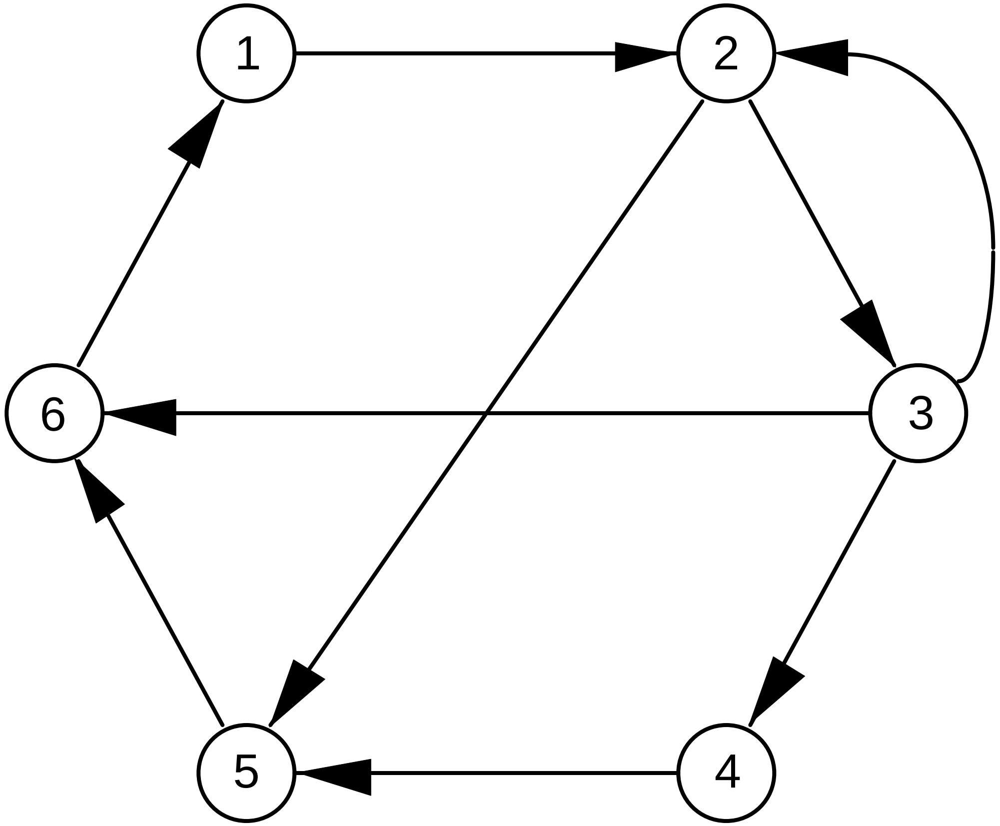


A cada vértice extremo destino se asocian todos los vértices extremos origen. Para el ejemplo-01, se tiene:

	1º	**2º**
1	6	
2	1	3
3	2	
4	3	
5	2	4
6	3	5

Esta forma de representación require una capacidad para almacenar la información igual producto del número de vértices por el número máximo de arcos que inciden en cualquiera de ellos.

LISTA DE ARCOS

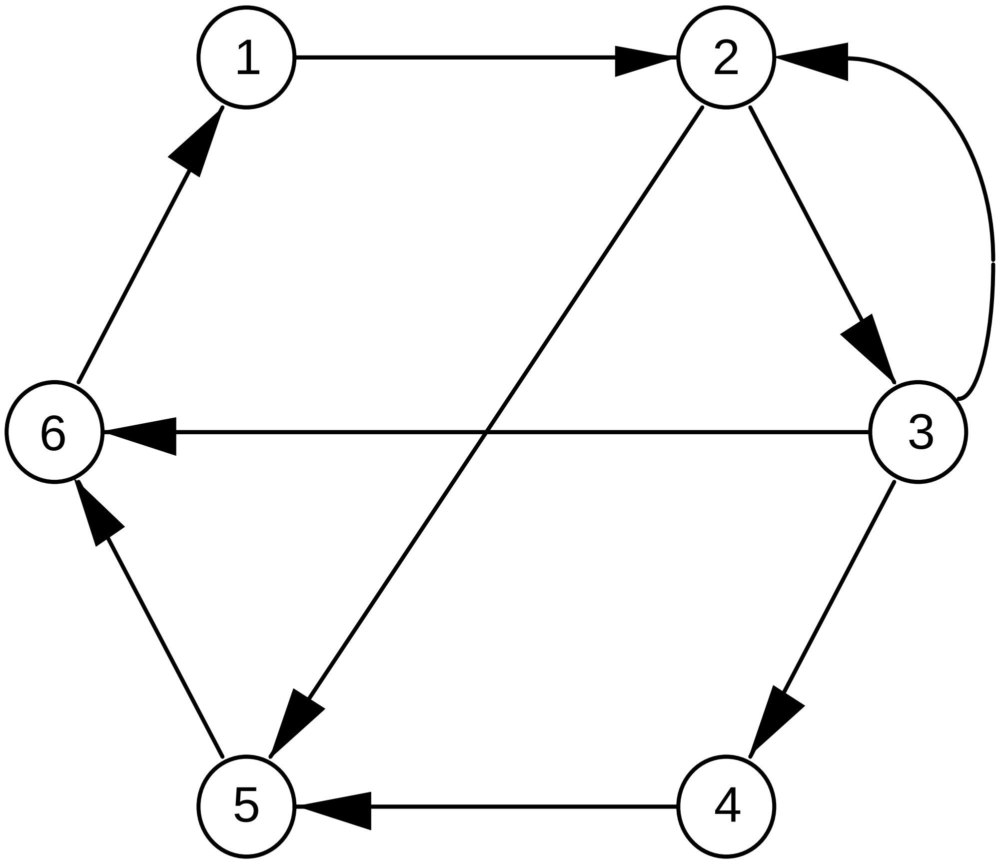


Se indican los vértices extremos origen y destino de cada arco. Para el ejemplo-01 se tiene:

Origen	1	2	2	3	3	3	4	5	6
Destino	2	3	5	2	4	6	5	6	1

Esta forma de representación require una capacidad para almacenar la información igual al doble del número de arcos (2xm).

LISTA CONDENSADA DE ARCOS



Forma compacta de la lista de arcos: se construye una lista de punteros. Para el ejemplo-01 se tiene:

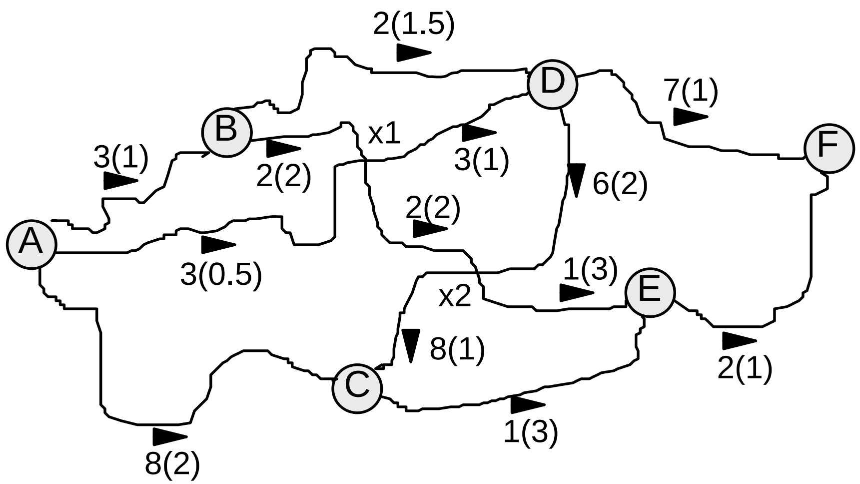
ver.	pun.		des.	nro.
1	1		2	1
2	2		3	2
3	4		5	3
4	7		2	4
5	8		4	5
6	9		6	6
			5	7
			6	8
			1	9

Se require una capacidad para almacenar la información igual al número de vértices más el número de arcos (n+m).

TERMINOLOGÍA

- *Semigrado interior de un vértice*: número de arcos que inciden en él.
- *Semigrado exterior de un vértice*: número de arcos que emergen de él.
- *Grado de un vértice*: número de arcos que inciden o emergen del vértice. Suma de semigrados.
- *Bucle*: arco o arista cuyos vértices extremos coinciden.
- *Camino*: sucesión de arcos tal que el vértice extremo final de cada uno (exceptuando el último) coincide con el vértice extremo origen de su siguiente en dicha sucesión.
- *Cadena*: brevemente, camino no orientado.
- *Circuito*: camino que se inicia y termina en el mismo vértice.
- *Ciclo*: Brevemente, circuito no orientado. En el grafo del ejemplo-01, 1-6-5-2-1 es un ciclo.
- *Euleriano o simple*: se dice de los caminos, cadenas, circuitos y ciclos que pasan una sola vez por todos los arcos o aristas del grafo.
- *Hamiltoniano o elemental*: se dice de los caminos, cadenas, circuitos y ciclos que pasan una sola vez por todos los vértices del grafo.
- *Conexo*: (o simplemente conexo) se dice del grafo en el que existe al menos una cadena entre cualquier pareja de vértices.
- *Fuertemente conexo*: se dice del grafo en el que existe al menos un camino entre cualquier pareja de vértices.
- *Árbol*: grafo conexo, sin bucles ni ciclos. Un árbol de decisión es un árbol.
- *Arborescencia*: grafo orientado, conexo, sin bucles ni ciclos, en el que todos los vértices tienen semigrado interior igual a 1, excepto uno, denominado raíz de la arborescencia, cuyo semigrado interior es igual a 0.

GRAFOS - SOPORTE DE MODELOS. Sistemas que necesitan expresar posibilidad de comunicación



Ejemplo-02: Una empresa dedicada al transporte de materias peligrosas desea establecer una ruta fija diaria con carga en la ciudad A y descarga en la ciudad F. La red de carreteras que permite conectar A y F es la mostrada en el croquis geográfico donde aparecen, además de una serie de núcleos urbanos y los cruces X1 y X2, los tiempos (en horas) precisos para cubrir los tramos y, entre paréntesis, el número medio de accidentes por cada 1000 viajes.

Problemas:

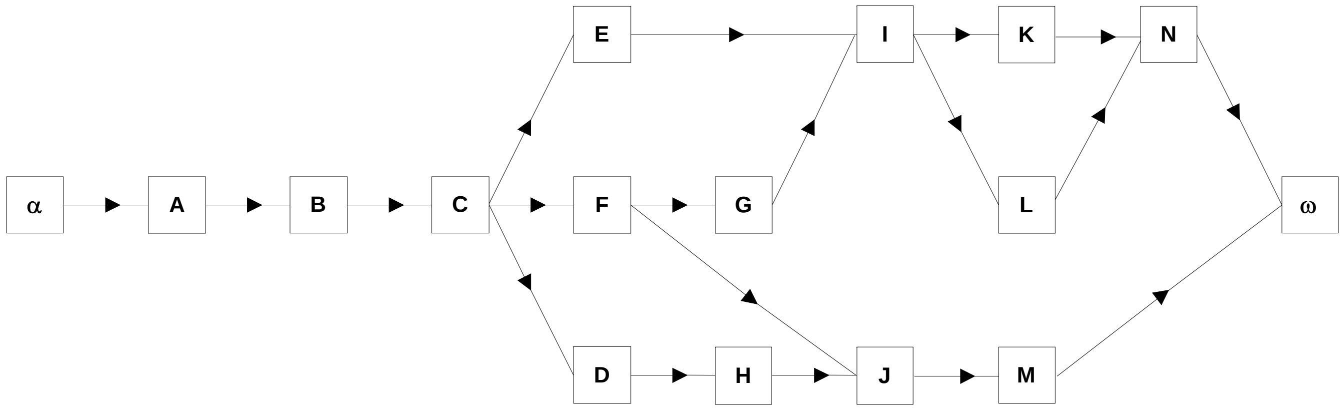
- ¿Qué ruta es la más rápida?
- ¿Qué ruta es la más segura?
- ¿Qué ruta es la más adecuada combinado ambos aspectos (rapidez y seguridad)?

GRAFOS - SOPORTE DE MODELOS. Sistemas que necesitan traducir relaciones de orden

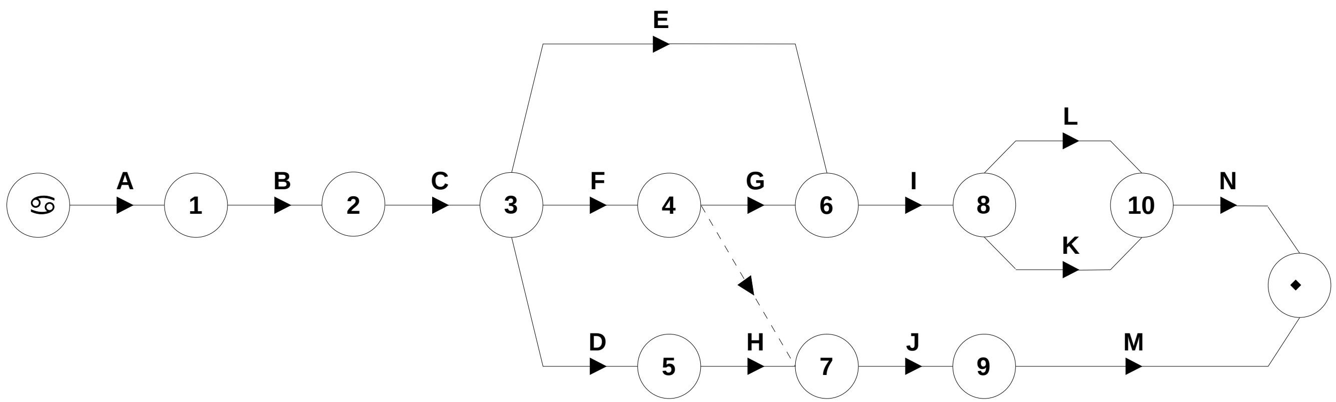
	PROYECTO: V-11		
Act.	**Denominación**	**Dur.**	**Prec.**
A	Excavación de terrenos	2	-
B	Cimentación	3	A
C	Obra estructural	10	B
D	Adecuar techos	6	C
E	Instalación eléctrica	7	C
F	Conducciones externas	4	C
G	Conducciones internas	5	F
H	Recubrimiento exterior	7	D
I	Recubrimiento interior	8	C, G
J	Pintura exterior	9	F, H
K	Pintura interior	6	I
L	Adecuar suelos	4	I
M	Acabados exteriores	2	J
N	Acabados interiores	7	K, L

Ejemplo-03: La promotora PH (PRO-HABITATGES Stat. Coop. Cat. Ltda.) encarga la edificación de un bloque de 11 viviendas unifamiliares en unos terrenos prácticamente horizontales en los que se ha realizado un estudio geotécnico del subsuelo con resultados satisfactorios. Después de una asignación inicial de recursos, el proyecto, al que se ha dado el distintivo V-11, se ha descompuesto en catorce actividades, cuyos datos básicos figuran en la tabla.

GRAFOS - SOPORTE DE MODELOS. Sistemas que necesitan traducir relaciones de orden



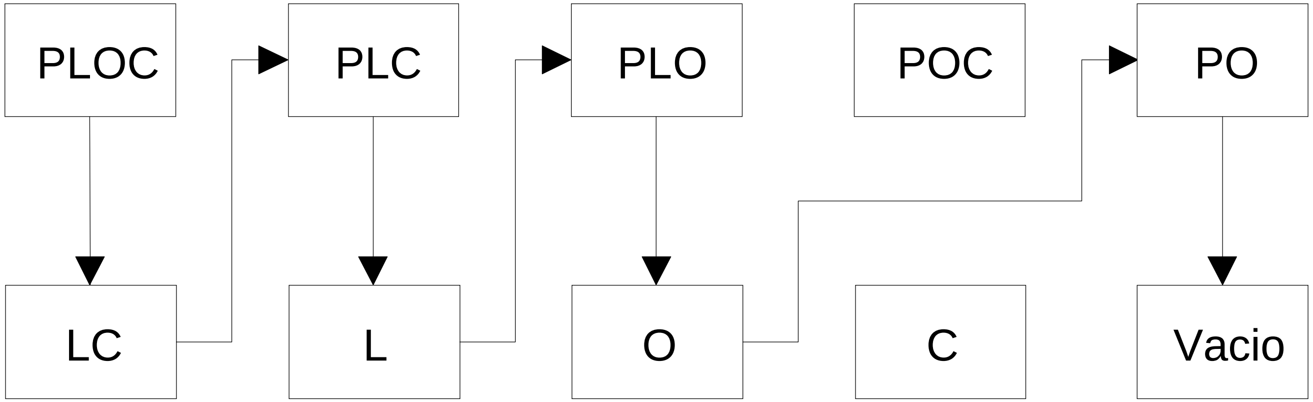
ROY



PERT

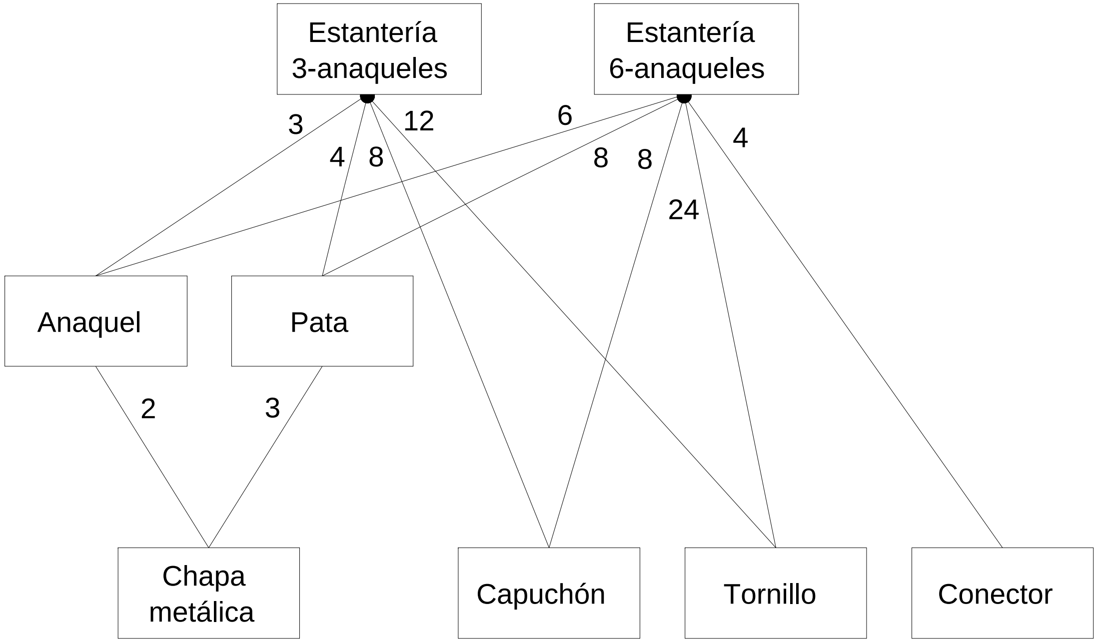
Act.	**Dur.**	**Prec.**
A	2	-
B	3	A
C	10	B
D	6	C
E	7	C
F	4	C
G	5	F
H	7	D
I	8	C, G
J	9	F, H
K	6	I
L	4	I
M	2	J
N	7	K, L

Ejemplo-04: Un paisano (P) debía transportar a la otra orilla de un río a un lobo (L), una oveja (O) y una col (C). La barca era tan pequeña que el paisano no podía llevar más de uno por vez; además, no podía dejar juntos, sin su presencia, ni a la oveja con el lobo ni a la col con la oveja ¿Cómo puede efectuarse el transporte?



Una forma de resolver el problema consiste en definir los estados posibles (los que tienen lugar en la primera orilla, por ejemplo) así como las transiciones permitidas entre éstos. Se debe buscar un camino (con número mínimo de arcos si es posible) que conecte el vértice inicial (PLOC) y el vértice final (Vacío).

GRAFOS - SOPORTE DE MODELOS. Sistemas que necesitan representar enlaces entre objetos



Ejemplo-05: Para la elaboración de una unidad de un producto que presenta dos variantes distintas: (01) estanterías de 3 anaqueles y (02) estanterías de 6 anaqueles, se precisa una serie de componentes: (03) anaquel, (04) pata, (05) capuchón, (06) tornillo, (07) conector y (08) chapa metálica, según la siguiente lista de materiales:

(01) = 3x(03)+4x(04)+8x(05)+12x(06)
(02) = 6x(03)+8x(04)+8x(05)+24x(06)+4x(07)
(03) = 2x(08)
(04) = 3x(08)

PROBLEMAS DE OPTIMIZACIÓN EN GRAFOS

Se estudiarán los siguientes:

- Árbol de expansión mínima o máxima (MST).
- Caminos mínimos o máximos entre vértices.
- Flujo máximo en una red de transporte.

Para resolver estos problemas se propondrán algoritmos.

El término *algoritmo* (derivado de ***Al-Jwarizmi***, sobrenombre de ***Mohámed ben Musa***) se emplea para designar un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Los algoritmos requieren y ofrecen:

- Una <u>iniciación</u> o datos de partida,
- <u>Dan un resultado</u> en un número <u>finito de iteraciones</u>.
- Las operaciones se agrupan normalmente en etapas que deben estar bien definidas.
- La <u>efectividad</u> de un algoritmo es función del tiempo que emplea en dar un resultado.

ÁRBOL PARCIAL MÍNIMO. Descripción del problema

El problema MST consiste en hallar un conjunto de aristas tal que permita la conexión entre toda pareja de vértices del grafo de referencia y que la suma de los valores asociados a dichas aristas sea mínima.

El grafo parcial hallado no puede tener ciclos (pues en caso contrario siempre se puede suprimir una arista reduciendo el valor total), por consiguiente es resultado debe ser por fuerza un árbol.

Un árbol parcial mínimo corresponde a una red de comunicaciones de coste mínimo que enlaza cualquier pareja de vértices de un conjunto dado. La determinación de dicho árbol es deseable en muchas situaciones:

- Conexión de un conjunto de poblaciones mediante una red de autopistas, ferrocarril o vuelos.
- Redes de suministro de agua, gas y electricidad.
- Redes de servicios como alcantarillado, líneas telefónicas, televisión por cable, etc.
- Conexión entre ordenadores.

ÁRBOL PARCIAL MÍNIMO. Resolución del problema

Algoritmos ávidos (greedy).

Algoritmo básico para el problema MST

Paso 0 Iniciación:

- Asociar a cada arista e_{ij} un valor l_{ij}, si no existe conexión entre los vértices i y j hacer l_{ij} igual a infinito (caso de mínimo).
- Sea S el conjunto de vértices conectados en la iteración en curso. Inicialmente, S está vacío.

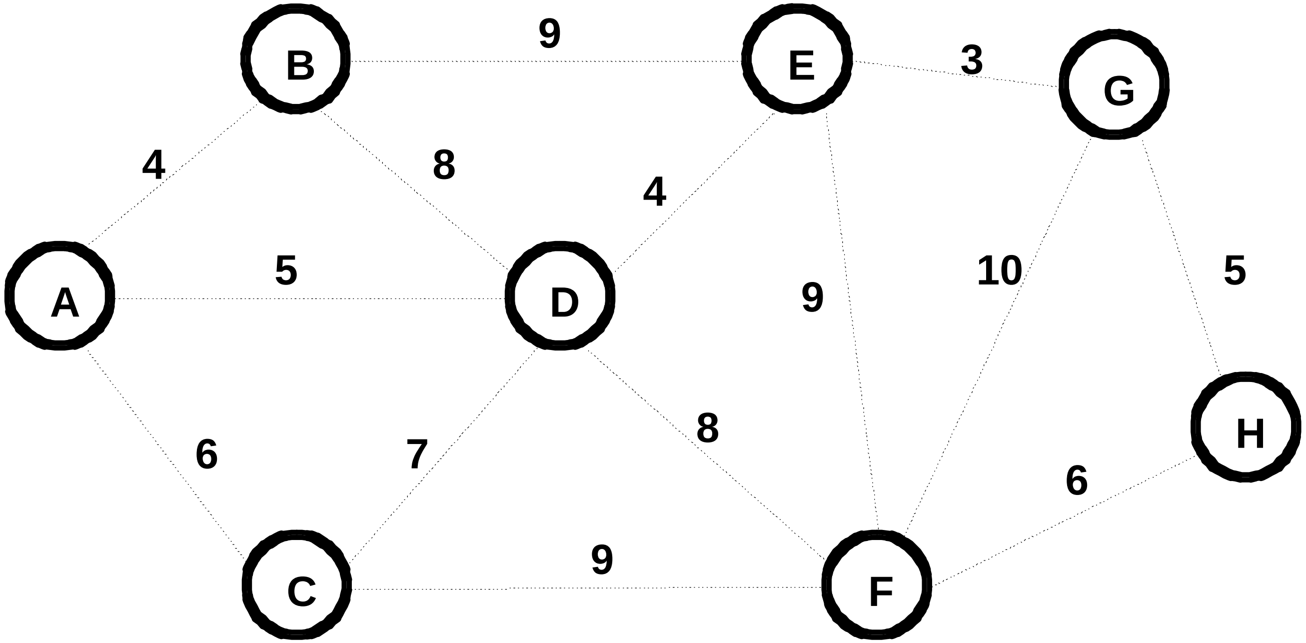
Paso 1 Tomar un vértice cualquiera e incorporarlo al conjunto S.

Paso 2 Buscar la arista e_{ij} de menor valor (coste, longitud, etc.) l_{ij} , tal que: el vértice i pertenece a S y el vértice j no pertenece a S. Sean i^* y j^* los vértices de la arista hallada (en caso de que haya empate entre arista, seleccionar cualquiera de ellas).

Paso 3 Incorporar el vértice j^* al conjunto S: $S \leftarrow S \cup \{j^*\}$

Paso 4 Si todos los vértices están conectados (S=X), Finalizar.
Si no, Ir a *Paso* 2.

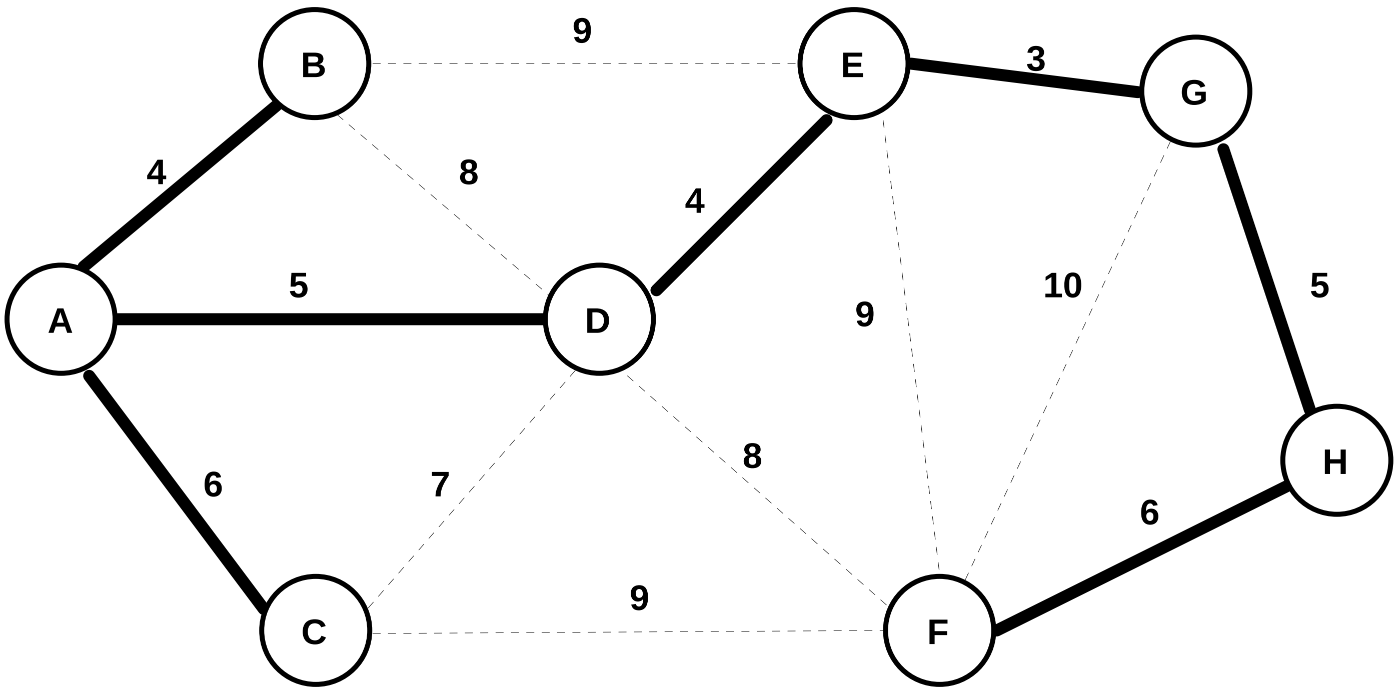
ÁRBOL PARCIAL MÍNIMO. Un ejemplo de resolución (I)



Ejemplo-06:

La administración del parque de atracciones Bellavista ha decidido la localización de 8 estaciones de control y servicio: A, B, C, D, E, F, G y H, y desea determinar los recorridos bajo los cuales se debe tender las líneas telefónicas para conectar todas las estaciones con una longitud de cable mínima. Después de analizar el terreno, se han determinado los trayectos sobre los que puede realizarse el tendido sin problemas y las distancias de los mismos medidas en Km; esta información se esquematiza en el grafo de la figura.

ÁRBOL PARCIAL MÍNIMO. Un ejemplo de resolución (II)



El valor total de la conexión es 33 Km.

CAMINOS EXTREMOS ENTRE VÉRTICES. Descripción del problema

Dado un grafo conexo, a cuyos arcos se asocian unos valores (distancias), se trata de encontar un camino entre dos vértices cuyo valor sea mínimo o máximo. Este problema es básico en el diseño de itinerarios.

La medida de la calidad de la solución:

- La distancia recorrida.
- El tiempo empleado.
- La fiabilidad.
- Los costes.
- El flujo de material.
- La capacidad.

Función a optimizar:

- Suma de valores. (tiempos)
- Producto de valores. (fiabilidad)
- El mínimo entre los valores de los arcos que constituyen el camino (medición de la capacidad)
- Combinación de sumas y de productos (cálculo de necesidades de materiales).

Variantes a estudiar:

- Camino de valor mínimo entre una pareja de vértices.
- Caminos de valor mínimo entre un vértice y todos los demás.
- Caminos de valor mínimo entre todas las parejas de vértices.

Dado un grafo y dos vértices: el origen del camino (α), y el final del camino (ω), se trata de hallar un camino (si existe) de forma que la suma de los valores asociados a los arcos que lo componen sea mínima.

Para resolver el problema se propone un algoritmo básico que emplea los elementos siguientes:

- l_{ij} Valor asociado al arco u_{ij}, si no existe arco este valor se iguala a infinito. Los valores deben ser no negativos
- P_i Etiqueta asociada al vértice i que contiene en la iteración en curso el valor del mejor camino desde α hasta i hallado hasta el momento.
- S Conjunto de vértices seleccionados en la construcción del camino hasta la iteración en curso. Inicialmente S contiene el vértice α.

En cada iteración se añade un arco al segmento ya construido hasta el momento, incorporando al conjunto S un nuevo vértice; evidentemente, el algoritmo termina cuando el vértice ω se añade al conjunto S.

En la k-ésima iteración se recalculan las etiquetas de los vértices no contenidos en S mediante la siguiente expresión:

$$P_z^k = \min \left\{ P_z^{k-1},\ P_y^{k-1} + l_{y,z} \right\}$$

donde y es el vértice que se incorporó a S en la iteración k-1.

Algoritmo básico:

Paso 0 Iniciación:

- Asociar a cada arco u_{ij} un valor l_{ij}, si no existe conexión entre los vértices i y j hacer l_{ij} igual a infinito (caso de mínimo).
- Sea S el conjunto de vértices conectados en la iteración en curso. Incorporar a S el vértice α.
- Hacer $P_\alpha=0$ y $P_i=\infty$ si i es distinto de α.

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

$$P_z = \min\{P_z, P_y + l_{y,z}\}$$

Paso 2 Entre todos los vértices (z) no pertenecientes a S, determinar z^* tal que:

$$P_{z*} = \min_{z \notin S}\{P_z\}$$

Paso 3 Si $P_{z*} = \infty$, Finalizar (no hay caminos de longitud finita entre α y los vértices no pertenecientes a S.

Si no, continuar.

Paso 4 Incorporar el vértice z^* al conjunto S: $S \leftarrow S \cup \{z^*\}$

Paso 5 Si $\omega=z^*$ (ω pertenece a S), Finalizar.

Si no, Ir a *Paso* 1.

El valor del camino de valor mínimo entre α y ω se halla en la etiqueta P_ω.

Ilustración *Algoritmo básico (I)*:

Ejemplo-07: La ciudad de Alfaville se considera dividida en seis barrios: B1, B2, B3, B4, B5 y B6, unidos entre sí por vías de tráfico tal como se muestra en la figura-8, en la que la cifra asociada a cada arco indica el tiempo medio que se necesita para ir de centro a centro de los barrios conectados en un momento determinado. ¿Cúal es el trayecto más corto para ir desde el centro **B1** hasta el de **B4**?.

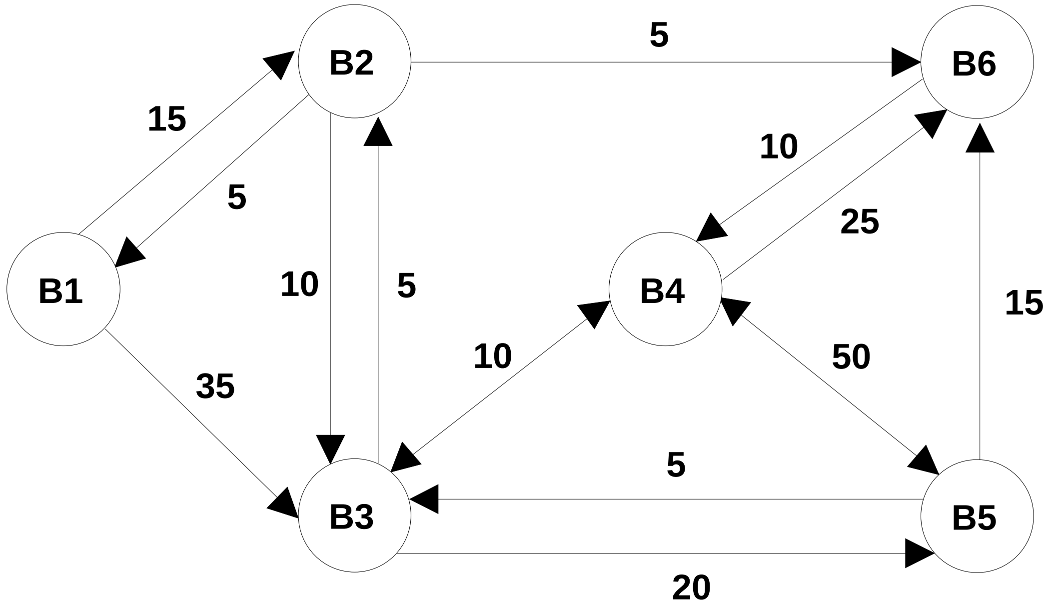


Figura-8: Grafo de tiempos inter-barrios de Alfaville.

Ilustración *Algoritmo básico (II)*:

Matriz distancias origen-destino							*Iteraciones*				
	B1	**B2**	**B3**	**B4**	**B5**	**B6**	K=0	K=1	K=2	K=3	K=4
B1	-	15	35	-	-	-	0	*	*	*	*
B2	5	-	10	-	-	5	-	**15 (B1)**	*	*	*
B3	-	5	-	10	20	-	-	35 (B1)	25 (B2)	**25 (B2)**	*
B4	-	-	10	-	50	25	-	-	-	30 (B6)	**30 (B6)**
B5	-	-	5	50	-	15	-	-	-	-	45 (B3)
B6	-	-	-	10	-	-	-	-	**20 (B2)**	*	*
								Y=B1	Y=B2	Y=B6	Y=B3
								Z*=B2	Z*=B6	Z*=B3	Z*=B4

Aplicando el algoritmo básico al ejemplo-07, se obtiene, en cuatro iteraciones, que el camino más rápido entre B1 y B4 tiene una duración de 30 minutos.

Ilustración *Algoritmo básico (III)*:

Si a lo largo del proceso de cálculo anotamos, tal como hemos hecho, el vértice anterior a partir del cual se obtiene la mejor etiqueta de los vértices, se puede reconstruir el camino, arco a arco, siguiendo esta traza en sentido inverso. En nuestro ejemplo, el camino hallado es: B1-B2-B6-B4 (ver figura-9).

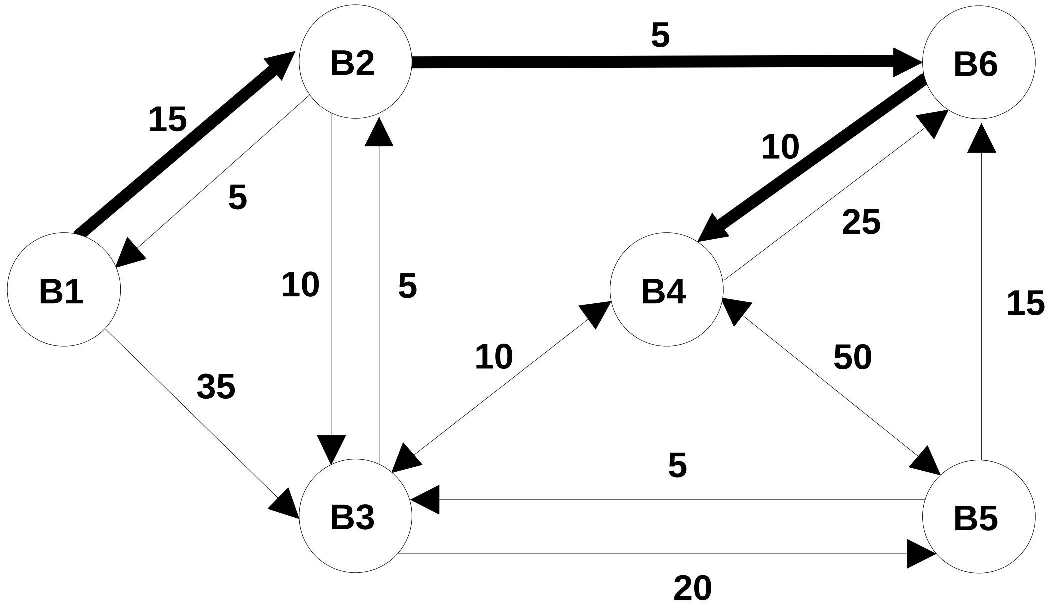


Figura-9

Caminos extremos entre un vértice y todos los demás.

Dado un grafo y un vértice origen (α) se trata de hallar los caminos (si existen) desde α hasta el resto de vértices de forma que la suma de los valores asociados a los arcos que los componen sea mínima o máxima.

Para resolver el problema describiremos un procedimiento general debido a Ford (familia de algoritmos) y un algoritmo básico.

Las características de dichos métodos son las siguientes:

- En cada iteración se etiquetan los vértices del grafo de forma que ninguna etiqueta se considera permanente hasta que el procedimiento termina. Al finalizar, las etiquetas contienen los valores óptimos de los caminos (mínimos o máximos) desde el vértice origen hasta el resto de vértices del grafo.

- En cada iteración se intenta mejorar los valores de las etiquetas de cada vértice a partir de sus arcos incidentes y las etiquetas asociadas a los vértices iniciales de éstos. El etiquetaje (en caso de mínimo) se basa en las siguientes recurrencias:

$$P_\alpha^0 = 0 \;\; ; \;\; P_i^0 = l_{\alpha,i} \quad \text{para todo } i \text{ distinto de } \alpha. \qquad [1]$$

$$P_i^k = \min\left\{P_i^{k-1}, \min_j\left(P_j^{k-1} + l_{j,i}\right)\right\} \qquad [2]$$

- El procedimiento finaliza cuando en la iteración en curso no es posible mejorar ninguna etiqueta,

Procedimiento de Ford:

Fase 0 Asociar una etiqueta a cada vértice, por ejemplo a partir de las igualdades [1].

Fase 1 Repetir las siguientes operaciones:

- Buscar un arco $u_{i,j}$ tal que: $P_i > P_j + l_{ji}$
- Hacer: $P_i = P_j + l_{ji}$

Finalizar si no existe ningún arco que cumpla la condición.

Evidentemente la lista de arcos se puede explorar de muchas formas distintas en la matriz de distancias origen destino: por filas, por columnas, incluso al azar siempre que la exploración sea exhaustiva; por consiguiente, cada método de exploración concreto dará lugar a una variante, un algoritmo, del procedimiento de Ford.

Algoritmo básico:

Paso 0 Iniciación:

- Asociar a cada arco u_{ij} un valor l_{ij}, si no existe conexión entre los vértices i y j hacer l_{ij} igual a infinito (caso de mínimo).
- Hacer $P_{\alpha}=0$ y $P_i=l_{\alpha i}$ si i es distinto de α.
- Para todo vértice, hacer t_i (vértice a partir del cual se marca i) igual a α.

Paso 1 Iteraciones o cuerpo del algoritmo:

- Iniciar el detector de cambios: $sw=0$.

- Para todo i ($1 \leq i \leq n$), Hacer:
    - Para todo j ($1 \leq i \leq n$) , Determinar: $S_{ji} = P_j + l_{ji}$
    - Entre todo j ($1 \leq j \leq n$), Determinar: j^* tal que $S_{j^* i} = min\{S_{ji}\}$
    - Si $S_{j^* i} < P_i$, Hacer: $P_i = S_{j^* i}$, $t_i = j^*$, $sw = 1$.

Paso 2 Si hay cambios ($sw=1$), Ir a *Paso* 1.
Si no ($sw=0$), Finalizar.

Ilustración *Algoritmo básico (I)*

Ejemplo-08: En el barrio B4 de Alfaville (ver ejemplo-07) está localizada la central de la policía municipal de la ciudad. Interesa conocer los caminos más rápidos desde este barrio al resto, considerando una distribución de tiempos tal como la mostrada en la figura-8.

Matriz distancias origen-destino							*Iteraciones*			
	B1	**B2**	**B3**	**B4**	**B5**	**B6**	K=0	K=1	K=2	K=3
B1	-	15	35	-	-	-	-	-	20 (B2)	20 (B2)
B2	5	-	10	-	-	5	-	15 (B3)	15 (B3)	15 (B3)
B3	-	5	-	10	20	-	10 (B4)	10 (B4)	10 (B4)	10 (B4)
B4	-	-	10	-	50	25	0 (B4)	0 (B4)	0 (B4)	0 (B4)
B5	-	-	5	50	-	15	50 (B4)	30 (B3)	30 (B3)	30 (B3)
B6	-	-	-	10	-	-	25 (B4)	20 (B2)	20 (B2)	20 (B2)

El algoritmo converge en tres iteraciones (la última sirve de comprobación) obteniendo los valores para los caminos que figuran en la columna k=3 de la tabla adjunta.

Ilustración *Algoritmo básico (II)*

Para reconstruir un camino, por ejemplo de B4 a B6, basta leer la traza en sentido inverso: B6 se ha etiquetado con B2, B2 se ha etiquetado con B3 que a su vez se ha etiquetado con B4; por tanto, el camino más rápido de B4 a B6 es en detalle: B4-B3-B2-B6 con un valor igual a 20 minutos (ver figura-10).

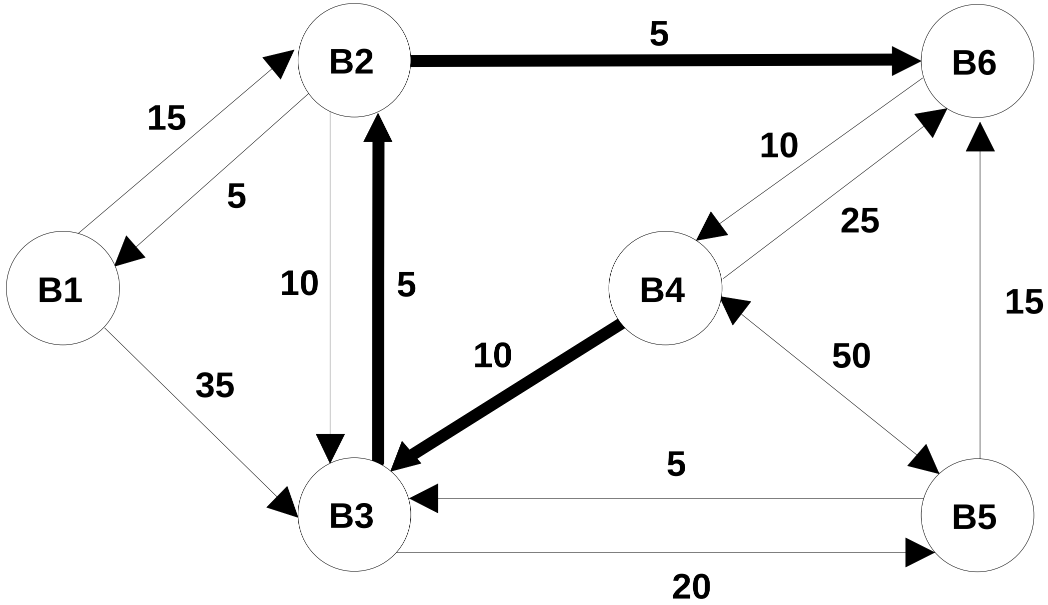


Figura-10.

Ilustración *Algoritmo básico (III). Resumen*

	Desde	*Hasta*	*Camino*	*Valor*	*Desglose*
	B4	B1	B4-B3-B2-B1	20	(10+5+5)
	B4	B2	B4-B3-B2	15	(10+5)
	B4	B3	B4-B3	10	(10 directo)
	B4	B4	-	0	(0 directo)
	B4	B5	B4-B3-B5	30	(10+20)
	B4	B6	B4-B3-B2-B6	20	(10+5+5)

En el algoritmo propuesto, observamos que las etiquetas obtenidas en una iteración se conservan inalteradas hasta que se han hallado las de la iteración siguiente. Esto es poco eficaz por dos motivos: (1) se precisa más memoria y (2) se emplean unos valores de las etiquetas, los hallados en la iteración anterior a la que está en curso, que no son los mejores obtenidos hasta el momento.

Una mejora inmediata del algoritmo consiste en modificar las igualdades [2] por las que siguen:

$$P_i^k = \min\left\{P_i^{k-1}, \min_{j<i}\left(P_j^k + l_{j,i}\right)\right\} \qquad P_i^k = \min\left\{P_i^{k-1}, \min_{j>i}\left(P_j^{k-1} + l_{j,i}\right)\right\} \qquad [3]$$

Caminos extremos entre toda pareja de vértices

Se pueden emplear los algoritmos anteriores (información básica para el diseño de rutas). Otra alternativa es emplear un algoritmo de caminos entre toda pareja de vértices.

Elementos

l_{ij} Valor asociado al arco u_{ij}, si no existe arco este valor se iguala a infinito.

P_{ij} Etiqueta asociada a la pareja de vértices (i,j) que contiene en la iteración en curso el valor del mejor camino desde i hasta j hallado hasta el momento. Inicialmente se hace $P_{ij} = l_{ij}$.

m_{ij} Traza asociada a la pareja de vértices (i,j) que contiene en la iteración en curso el vértice siguiente a i en el camino (de mejor valor hasta el momento) de i a j. Al inicio del procedimiento se hace:

- Si i es iguala a j, $m_{ij} = j$.
- Si i es distinto de j y existe el arco u_{ij} , $m_{ij} = j$.
- Si i es distinto de j y no existe el arco u_{ij} , m_{ij} se iguala a infinito.

En cada iteración se intenta mejorar las etiquetas de cada par de vértices, empleando (en caso de mínimo) las siguientes recurrencias:

$$P_{ij}^{k} = \min\left\{P_{ij}^{k-1}, \min_{z<j}\left(l_{iz} + P_{zj}^{k}\right)\right\} \qquad P_{ij}^{k} = \min\left\{P_{ij}^{k-1}, \min_{z>j}\left(l_{iz} + P_{zj}^{k-1}\right)\right\} \qquad [4]$$

El procedimiento finaliza cuando en la iteración en curso no es posible mejorar ninguna etiqueta.

Algoritmo de caminos extremos entre toda pareja de vértices:

Paso 0 Iniciación:

- Asociar a cada arco u_{ij} un valor l_{ij}, si no existe conexión entre los vértices i y j hacer l_{ij} igual a infinito (caso de mínimo).
- Hacer $P_{ij}=l_{ij}$.
- Hacer $m_{ij}=j$, si $i=j$ o existe u_{ij} ; si no, hacer m_{ij} igual a infinito.

Paso 1 Iteraciones o cuerpo del algoritmo:

Iniciar el detector de cambios: sw=0.

Para todo i , Para todo j y Para todo z ($1\leq i,j,z \leq$n), Hacer:

- Si $P_{ij} > l_{iz}+P_{zj}$, Hacer: $P_{ij} = l_{iz}+P_{zj}$, $m_{ij} = z$, $sw = 1$.
- Si no, Continuar.

Paso 2 Si hay cambios (sw=1), Ir a *Paso* 1.

Si no (sw=0), Finalizar.

Ilustración *Algoritmo de caminos extremos entre toda pareja de vértices (I)*

Ejemplo-09: Interesa conocer los caminos más rápidos entre todos los pares de barrios de la ciudad de Alfaville. considerando una distribución de tiempos tal como la mostrada en la figura-8.

Iteración 0:

P	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	0	15	35	-	-	-
B2	5	0	10	-	-	5
B3	-	5	0	10	20	-
B4	-	-	10	0	50	25
B5	-	-	5	50	0	15
B6	-	-	-	10	-	0

m	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	B1	B2	B3	-	-	-
B2	B1	B2	B3	-	-	B6
B3	-	B2	B3	B4	B5	-
B4	-	-	B3	B4	B5	B6
B5	-	-	B3	B4	B5	B6
B6	-	-	-	B4	-	B6

Ilustración *Algoritmo de caminos extremos entre toda pareja de vértices (II)*

Iteración 1:

P	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	0	15	25	45	55	20
B2	5	0	10	15	30	5
B3	10	5	0	10	20	10
B4	20	15	10	0	30	20
B5	15	10	5	15	0	15
B6	30	25	20	10	40	0

m	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	B1	B2	B2	B3	B3	B2
B2	B1	B2	B3	B6	B3	B6
B3	B2	B2	B3	B4	B5	B2
B4	B3	B3	B3	B4	B3	B3
B5	B3	B3	B3	B3	B5	B6
B6	B4	B4	B4	B4	B4	B6

Iteración 2: (idem a 3)

P	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	0	15	25	30	45	20
B2	5	0	10	15	30	5
B3	10	5	0	10	20	10
B4	20	15	10	0	30	20
B5	15	10	5	15	0	15
B6	30	25	20	10	40	0

m	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**
B1	B1	B2	B2	B2	B2	B2
B2	B1	B2	B3	B6	B3	B6
B3	B2	B2	B3	B4	B5	B2
B4	B3	B3	B3	B4	B3	B3
B5	B3	B3	B3	B3	B5	B6
B6	B4	B4	B4	B4	B4	B6

Ilustración *Algoritmo de caminos extremos entre toda pareja de vértices (III)*

Para obtener el valor de un camino basta consultar la matriz de etiquetas. Por ejemplo, para ir de B6 a B1 (fila B6 columna B1) se emplean 30 minutos, siguiendo la secuencia de barrios: B6-B4-B3-B2-B1 (ver figura-11). Esta secuencia se obtiene consultando la matriz de traza (m):

Primero, de B6 vamos a B4, pues $m_{B6,B1}$ = B4; segundo, de B4 vamos a B3 ($m_{B4,B1}$ = B3); tercero, de B3 vamos a B2 ($m_{B3,B1}$ = B2); finalmente, de B2 vamos a B1, ya que $m_{B2,B1}$ = B1.

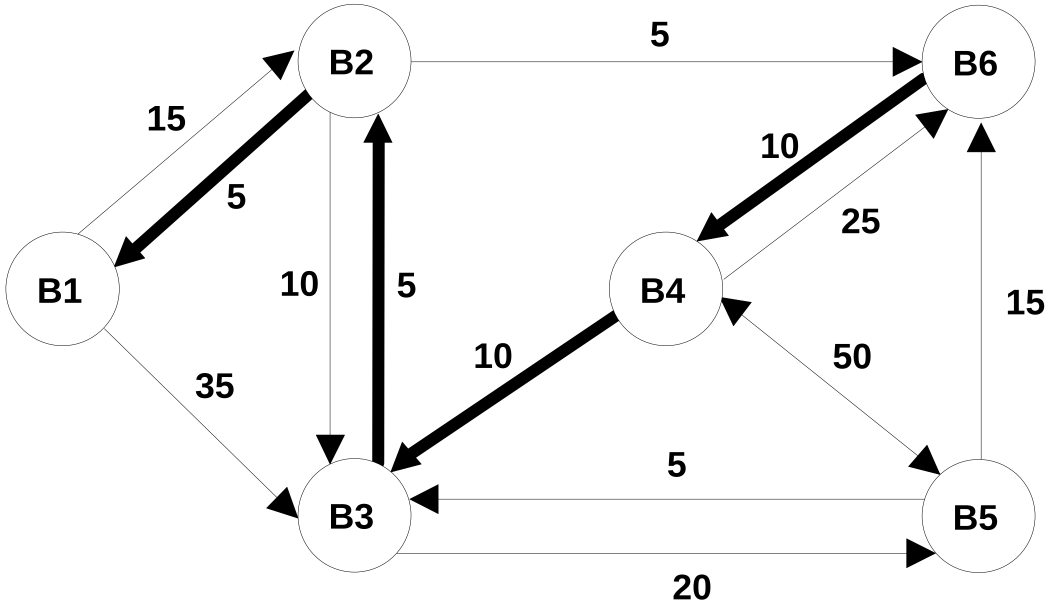


Figura-11.

FLUJOS ÓPTIMOS EN REDES. Descripción del problema

Transportar unidades (productos, materias primas, información, energía, etc.) desde un punto origen hasta un punto destino a través de una red con limitaciones de capacidad.

Ejemplos de capacidad:

- tasas de producción de las máquinas
- carga máxima de un semiremolque
- número máximo de vehículos que admite una vía
- caudales máximos a través de una red de tuberías
- intensidad de la corriente en un tendido eléctrico

La capacidad global del sistema queda limitada por las capacidades de sus elementos.

Problema: Hallar el flujo máximo en una red de transporte.

Grafo o red de transporte

La definición formal de una Red de Transporte es la siguiente:

- Es un grafo orientado, conexo y finito.
- El grafo no presenta ni bucles ni circuitos.
- Cada arco u tiene asociado un valor $c(u) \geq 0$ denominado capacidad del arco.
- Existe un sólo vértice en el grafo con semigrado interior igual a 0, al que se denomina origen de la red y notaremos por α.
- Existe un sólo vértice en el grafo con semigrado exterior igual a 0, al que se denomina salida de la red y notaremos por ω.

UN EJEMPLO PROTOTIPO (I)

Ejemplo-10: Un taller fabrica cinco variantes de cable de alta tensión: X1 a X5; para ello, dispone de 8 máquinas cuyas tasas máximas de producción, medidas en metros por hora, son: 50, 70, 60, 80, 70, 40, 60 y 50, con independencia del cable tratado, siendo, además, los tiempos de preparación despreciables. Las rutas de estos productos en el taller son:

- Variante X1: M1-M2-M3-M4-M5.
- Variante X2: M1-M6-M5.
- Variante X3: M7-M3-M8.
- Variante X4: M1-M2-M3-M8.
- Variante X5: M7-M3-M4-M5.

En tales condiciones, se desea conocer la producción horaria máxima que puede ofrecer el taller.

UN EJEMPLO PROTOTIPO (II)

El problema propuesto se puede asimilar al cálculo del flujo máximo a través de una red de transporte. La capacidad aociada a un arco (números entre paréntesis en la figura) se corresponde con la tasa de producción máxima de la máquina que dicho arco representa.

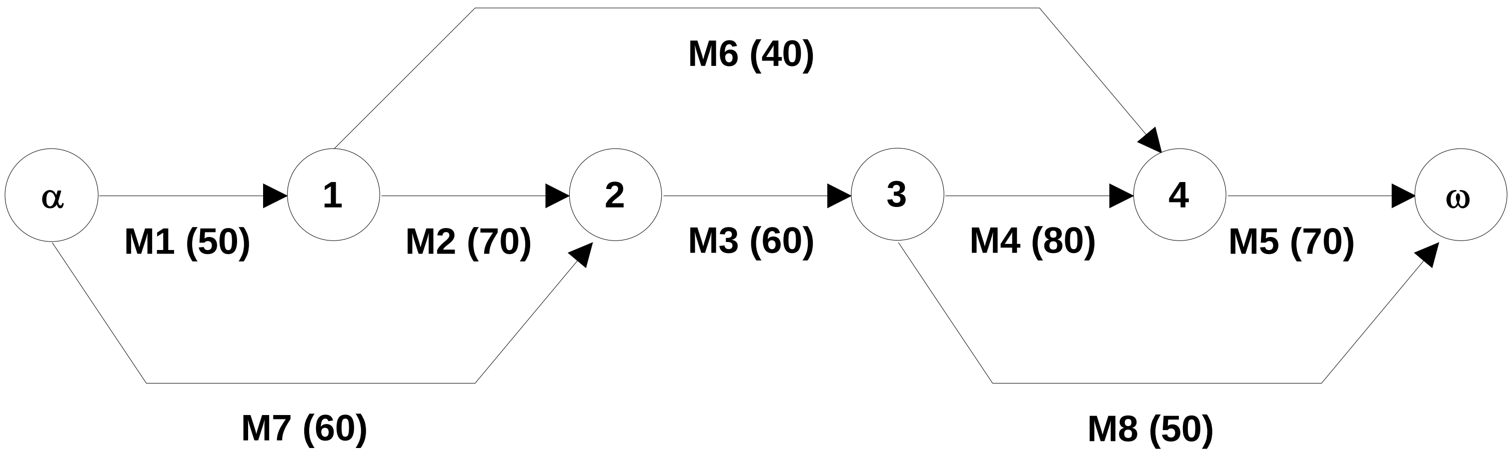


Figura-12.

- Variante X1: M1-M2-M3-M4-M5.
- Variante X2: M1-M6-M5.
- Variante X3: M7-M3-M8.
- Variante X4: M1-M2-M3-M8.
- Variante X5: M7-M3-M4-M5.

Flujo óptimo en una red de transporte

El *concepto de flujo* se corresponde con un conjunto de valores φ(u) (a determinar), asociados a los arcos, tales que:

- Son mayores o iguales que 0. φ (u)≥ 0.
- Respetan las capacidades de los arcos: φ (u)≤ c(u).
- En todo vértice, a excepción de α y ω, el flujo total que llega al vértice es igual al flujo total que sale del vértice.
- El flujo que atraviesa la red es igual al que sale de α e igual al que llega a ω.

$$[MAX]Z = \varphi_{M1} + \varphi_{M7} = \varphi_{M5} + \varphi_{M8}$$

(1) *Compatibilidad* :

$$0 \le \varphi_{M1} \le 50,\ \ 0 \le \varphi_{M2} \le 70,\ \ 0 \le \varphi_{M3} \le 60,\ \ 0 \le \varphi_{M4} \le 80$$

$$0 \le \varphi_{M5} \le 70,\ \ 0 \le \varphi_{M6} \le 40,\ \ 0 \le \varphi_{M7} \le 60,\ \ 0 \le \varphi_{M8} \le 50$$

(2) *Continuidad* :

$$\varphi_{M1} = \varphi_{M2} + \varphi_{M6};\ \varphi_{M2} + \varphi_{M7} = \varphi_{M3}; \varphi_{M3} = \varphi_{M4} + \varphi_{M8}; \varphi_{M4} + \varphi_{M6} = \varphi_{M5}$$

Concepto de Corte

Es un conjunto de arcos cuya supresión de la red elimina las conexiones (caminos) entre α y ω.
En el ejemplo-10, las máquinas M1 y M7 constituyen un corte; M6, M4 y M8 constituyen otro corte.

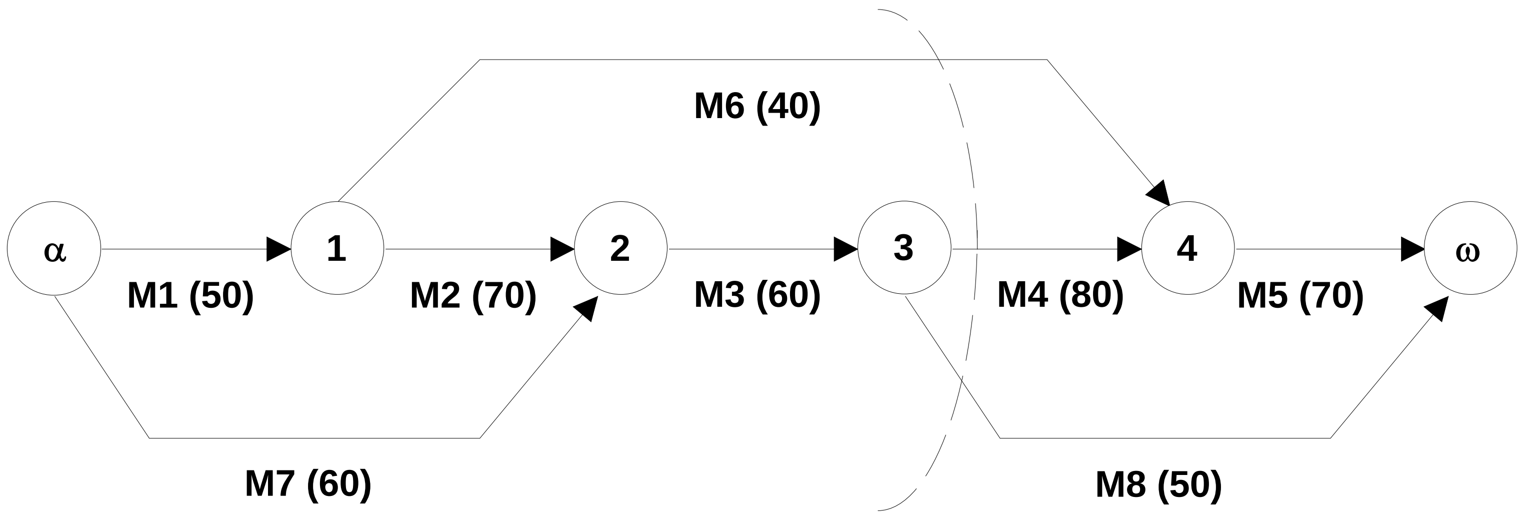


Figura-13.

Se llama *capacidad de un corte* a la suma de las capacidades de los arcos que componen dicho corte; por ejemplo, la capacidad del corte {M4,M6,M8} es 170 (80+40+50).

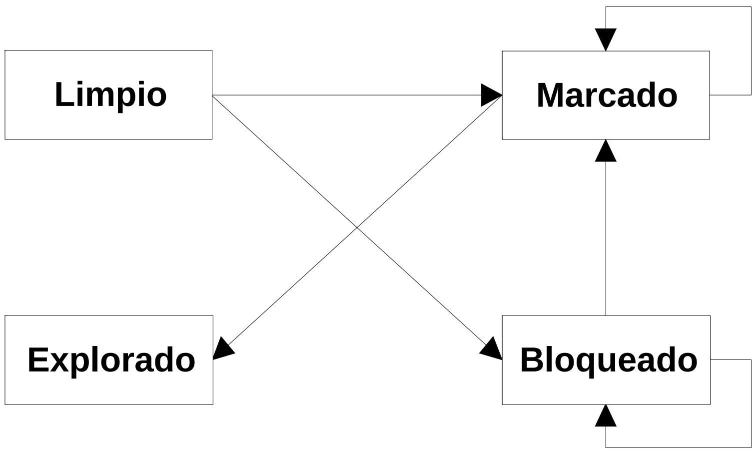
El flujo que pasa por la red debe ser menor o igual que la capacidad del *corte de mínima capacidad* (*teorema de Ford-Fulkerson*).

Resolución del problema: Algoritmo básico

El algoritmo básico propuesto consiste en una serie de iteraciones en cada una de las cuales se incrementa el flujo a través de un camino o una cadena que une α con ω; cuando ya no posible seguir iterando, el flujo es máximo.

La cadena que permite incrementar el flujo en una iteracion se determina mediante el etiquetaje de vértices, los cuales pueden presentar los estados siguientes:

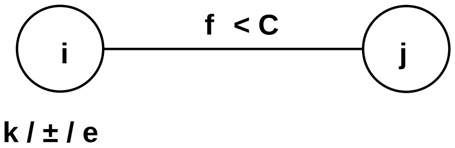
- ***Limpio***: vértice sin visitar al que no se ha asignado etiqueta.
- ***Marcado***: vértice al que se ha asignado un flujo de paso y su procedencia.
- ***Bloqueado***: vértice al que no puede asignarse un flujo de paso.
- ***Explorado***: vértice a partir del cual se han marcado o bloqueado todos sus adyacentes.



Reglas de etiquetaje

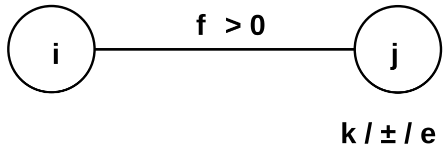
El etiquetaje de vértices no-explorados se realiza a partir de un vértice adyacente marcado (en fase de exploración) aplicando una de las dos reglas siguientes:

Regla 1: Sentido directo, permite incrementar el flujo cuando el flujo asignado al arco es inferior a su capacidad.



Se marca j con: i / + / min { e, C-f }

Regla 2: Sentido inverso, permite reducir el flujo cuando el flujo asignado al arco es mayor que 0.



Se marca i con: j / - / min { e, f }

Procedimiento básico:

Paso 0 Iniciación: Se establece un flujo compatible en la red, por ejemplo asignando a todos los arcos un flujo igual a 0.

Paso 1 Iteraciones o cuerpo del algoritmo:

- Se *marca* el vértice α con ./+/∞. El resto de vértices pasa al estado *limpio*.

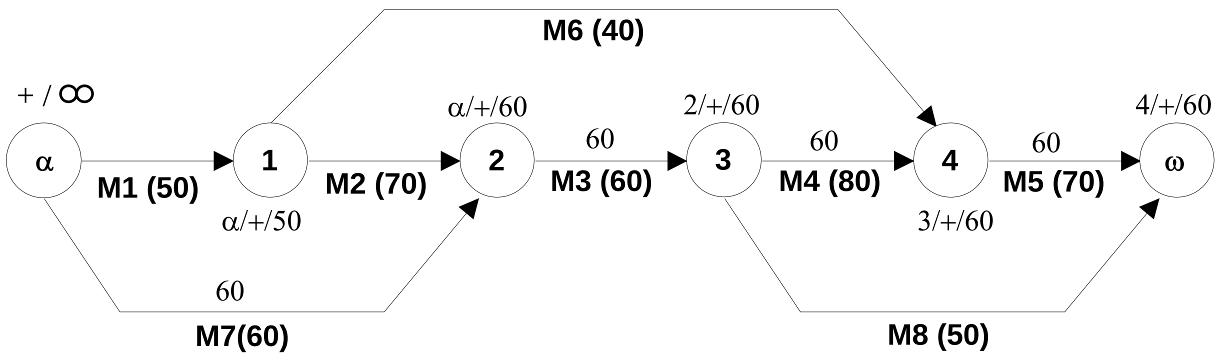
- Mientras haya vértices *marcados*, Hacer:
    - Tomar un vértice *marcado*; sea *v* dicho vértice.
    - A partir de *v*, se intenta marcar todos sus vértices adyacentes *no*-*explorados* siguiendo las reglas 1 y 2. Los vértices adyacentes que no se pueden marcar pasan al estado *bloqueado*.
    - El vértice *v* pasa al estado *explorado*.

Paso 2

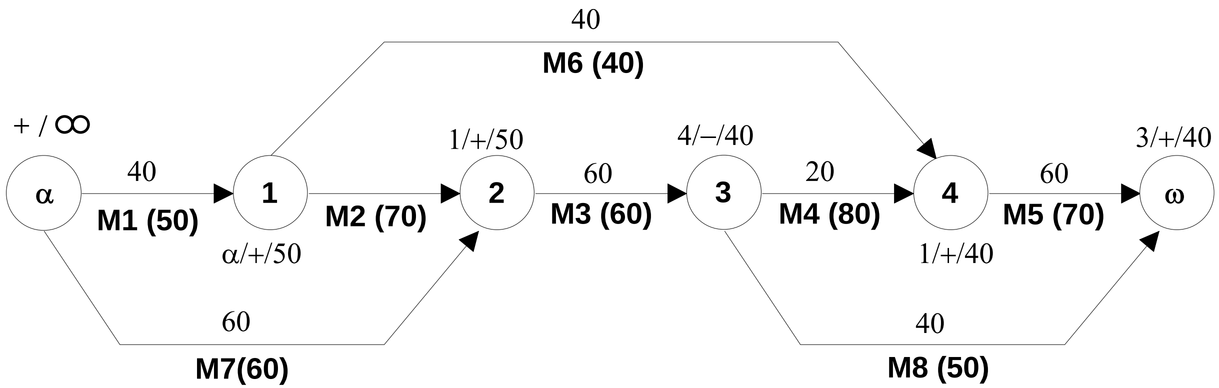
- Si ω está *explorado*, Incremetar el flujo en la red a través de la cadena que indican las etiquetas en una cantidad igual a la asignada ω. Ir a *Paso* 1.
- Si ω está *limpio* o *bloqueado*, Finalizar: el flujo es óptimo.

Iteraciones (I)

Iteración 1: El flujo se incrementa en 60 unidades a través de la cadena M7-M3-M4-M5. Flujo acumulado igual a 60 unidades.

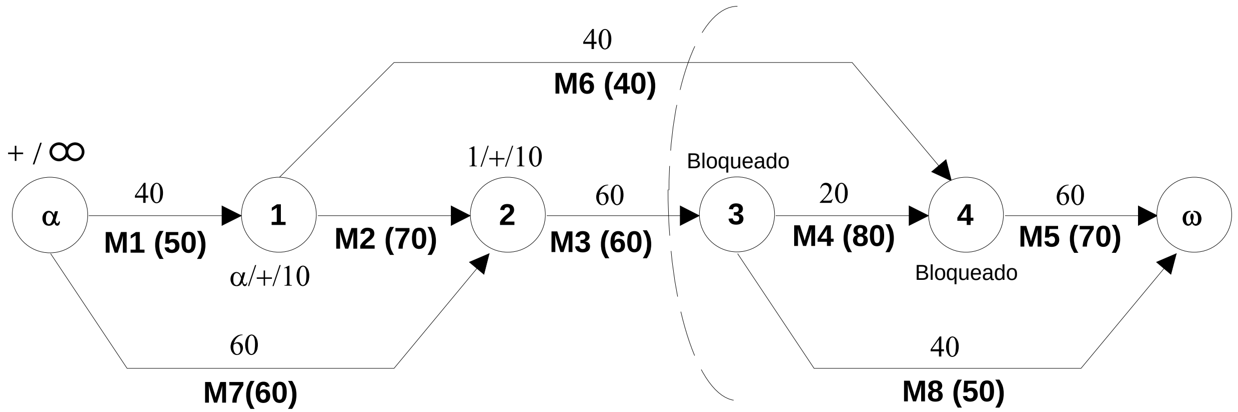


Iteración 2: El flujo se incrementa en 40 unidades a través de la cadena M1-M6-M4-M8. Flujo acumulado igual a 100 unidades.



Iteraciones (II)

Iteración 3: El flujo no se puede incrementar, ω está *límpio*; el flujo es óptimo e igual a 100 unidades. El corte de mínima capacidad lo constituyen las máquinas M3 y M6: son el cuello de botella del taller.



Aquí podemos observar que el flujo máximo que atraviesa la red (100) coincide con la capacidad de un corte (M3,M6) cuyos arcos si se suprimen del grafo imposibilitan el flujo desde el origen hasta la salida. Este corte es además el de capacidad mínima como indica el teorema de Ford-Fulkerson.