U4: MODELOS DE RED

Muchos problemas de optimización se analizan mejor por representación grafica

BIBLIOGRAFÍA

- RENDER Capítulo 11 – p. 429
- ANDERSON Cap.10 - Página 418

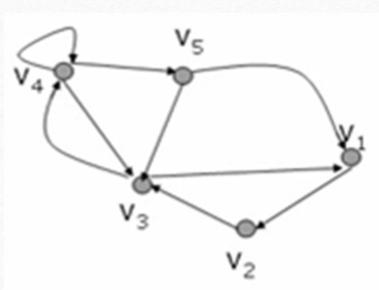
- Una gráfica o una red se compone de nodos y arcos
- Un arco es un par ordenado de puntos extremos y representa una posible dirección de movimiento que podría ocurrir entre puntos extremos o vértices
  - Los grafos pueden ser dirigidos o no dirigidos

	GRAFOS DIRIGIDOS	**GRAFOS NO DIRIGIDOS**
Los nodos se llaman		VERTICES O NODOS
Los pares ordenados son	ORDENADOS (*)	NO ORDENADOS
Los arcos se llaman	ARCOS	ARISTAS
Los arcos se representan	PUNTOS Y FLECHAS	POR PUNTOS Y LINEAS

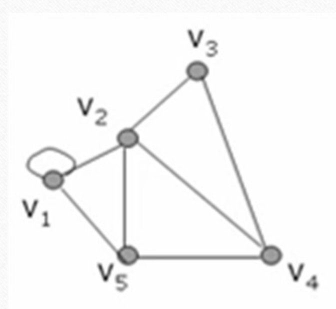
(*) Ordenado significa que tiene un sentido, que no es lo mismo (v1, v2) que (v2,v1)

EJEMPLOS

- GRAFOS DIRIGIDOS



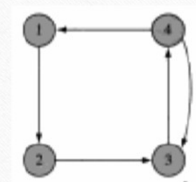
- GRAFOS NO DIRIGIDOS



CONCEPTOS BASICOS

Tener presente que de acuerdo a los textos consultados o la disciplina a que se los aplique estos tres conceptos pueden variar

- Cadena:
  - Es toda sucesión finita alterna de vértices y aristas.
  - Es una secuencia de arcos tal que cada arco tiene exactamente un vértice en común con el arco previo. Ejemplo: (1,2)-(2,3)-(4,3)
- Una trayectoria o camino
  - Es una cadena en la que el nodo terminal de cada arco es idéntico al nodo inicial del arco siguiente. Ejemplo: (1,2)-(2,3)-(3,4) (es una cadena y una trayectoria)
- Un ciclo es cuando una cadena comienza y termina en el mismo vértice
- Un árbol es una serie de nodos que no contiene ciclos
- Un árbol expandido es un árbol que conecta todos los nodos de la red (contiene n-1 arcos)



- Modelos Algoritmos
  - Árbol de Expansión Mínima
    - Algoritmo de Prim
    - Algoritmo de Kruskal
  - Problema de la Trayectoria más corta
    - Algoritmo de Dijkstra
    - Problema camino mas corto como PL Entera y Transporte
  - Problemas de Flujo Máximo
    - Método de Ford-Fulkerson
    - Como Problema Lineal
  - Pert y CPM
    - Camino Crítico, Holguras, Compresión Costo/Tiempo
    - Utilización de PL para compresión de red

Árbol de Expansión Mínima

Se utiliza cuando se necesita conectar todos los nodos de una red sin formar loops

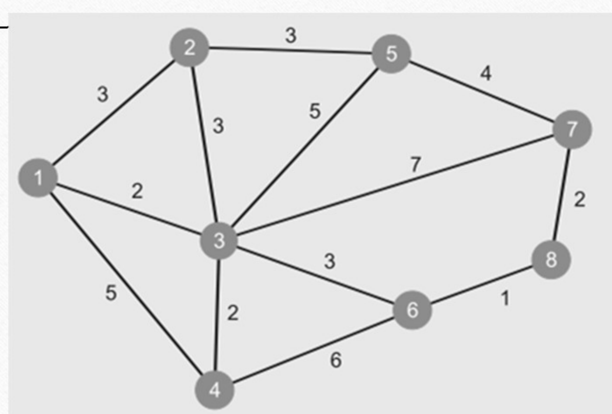
Con el objetivo de minimizar la longitud total

Usos: Redes con cableados, redes de electricidad, ferroviarias, etc

ALGORITMO DE PRIM

- Comenzar arbitrariamente en cualquier nodo y conectarlo con el más próximo (menos costoso)
- Identificar dentro del conjunto de los nodos no conectados, el nodo más próximo o menos costoso de cualquiera de los nodos ya conectados.
- Deshacer los empates arbitrariamente y agregar dicho nodo a los ya conectados
- Repetir hasta finalizar con todos los nodos

Ejemplo (Render, p. 431)



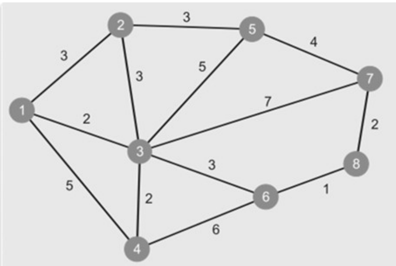
Solución

PASO	NODOS CONECTADOS	NODOS NO CONECTADOS	NODO NO CONECTADO MÁS CERCANO	ARCO SELECCIONADO	LONGITUD DEL ARCO	DISTANCIA TOTAL
1	1	2, 3, 4, 5, 6, 7, 8	3	1–3	2	2
2	1, 3	2, 4, 5, 6, 7, 8	4	3–4	2	4
3	1, 3, 4	2, 5, 6, 7, 8	2 o 6	2–3	3	7
4	1, 2, 3, 4	5, 6, 7, 8	5 o 6	2–5	3	10
5	1, 2, 3, 4, 5	6, 7, 8	6	3–6	3	13
6	1, 2, 3, 4, 5, 6	7, 8	8	6–8	1	14
7	1, 2, 3, 4, 5, 6, 8	7	7	7–8	2	16

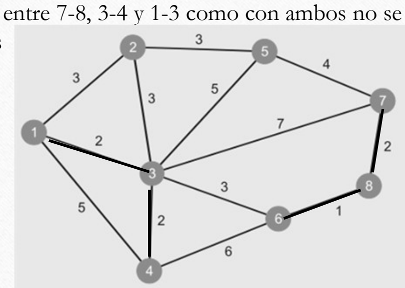
Algoritmo de Kruskal

Sólo cambia el punto de inicio. Aquí buscamos comenzar con la arista de menor valor y unimos.

Continuamos siempre que no se formen ciclos

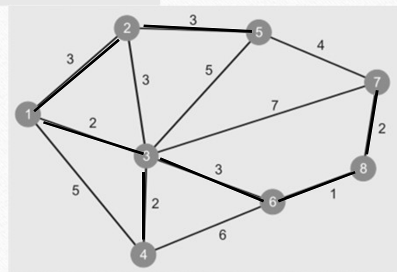


Comenzamos por 1 es decir la unión entre 6-8, luego elegimos el 2 entonces hay empate entre 7-8, 3-4 y 1-3 como con ambos no se hace ciclo conectamos



Seleccionamos los 3 tenemos 2-5, 1-2, 2-3, 3-6. Añadimos las dos primeras uniones y luego ya no podemos hacer 2-3 porque generaría ciclo, pero si 3-6

Seleccionamos los 4 tenemos 5-7 no podemos porque se forma un ciclo
Seleccionamos los 5 tenemos 1-4 y 3-5 no podemos porque se forman ciclos
Seleccionamos los 6 tenemos 4-6 no podemos porque se forma un ciclo
Seleccionamos los 7 tenemos 3-7 no podemos porque se forma un ciclo



Por lo tanto ese es el AEM

ALGORITMO DE DIJKSTRA (1959)

- Se parte de un nodo inicial y se calcula el camino mas corto a los restantes nodos/vértices del grafo accesibles desde él.
- Este proceso se realiza iterativamente, por lo que agregaremos una etiqueta temporaria junto a cada uno de los nodos restantes, anotaremos cual es su distancia mínima al inicial y cual es el nodo anterior en el camino mas corto del inicial a él mismo.

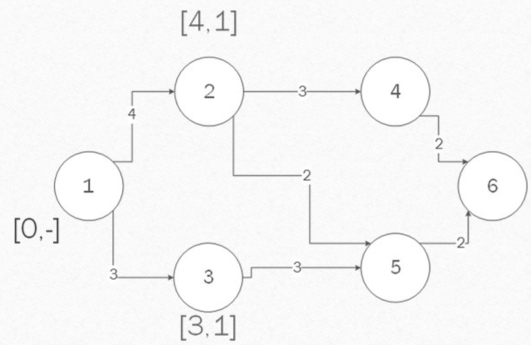
Algoritmo de Dijkstra

Etiqueta= [u_i+dij , i] donde: ui = es la distancia más corta desde el nodo 1 hasta el nodo i
dij= es la longitud del arco entre nodos ij

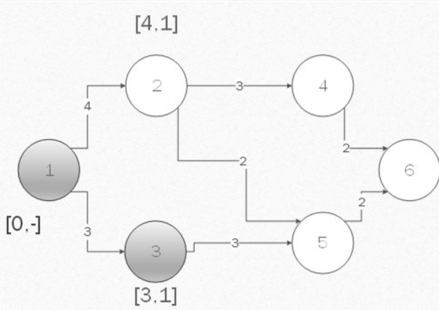
Utilizaremos el rojo para las etiquetas y nodos permanentes y el azul para las transitorias.

- Marcamos el nodo 1 con una etiqueta permanente de 0
- Etiquetamos en forma temporal todos los nodos conectados al nodo 1 con la longitud del arco.
- Elegimos la menor, es decir

Min {etiqueta temporales por nodo}



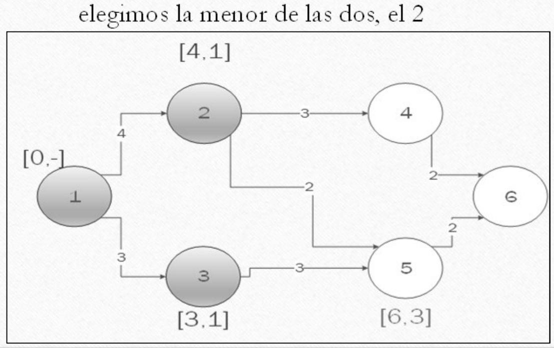
Etiqueta=[Distancia acumulada, Nodo anterior]



Ahora sabemos que el nodo más cercano al nodo 1 es el 3

Entonces continuamos con el nodo 3 y repetimos el proceso, es decir etiquetamos los nodos adyacentes o vecinos al 3, en este caso solo el 5.
Tenemos dos etiquetas temporales la del 2 y la del 5 elegimos la menor de las dos, el 2

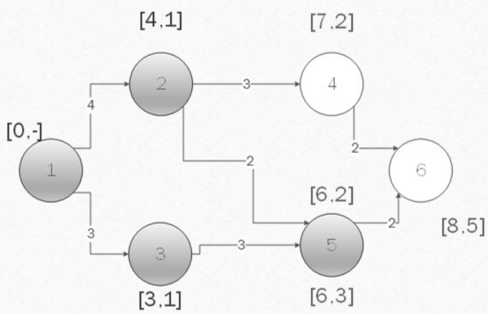
Si tenemos más de una etiqueta elegimos la menor y la transformamos en permanente

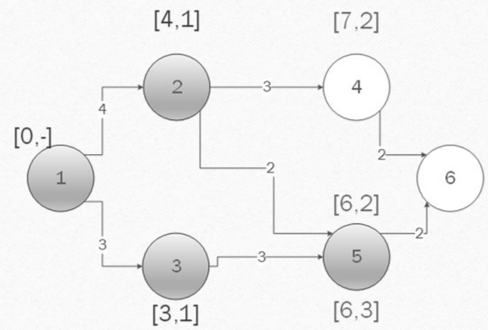


Ahora continuamos trabajando desde el nodo 2 y etiquetamos a los adyacentes

Como vemos la menor distancia esta en el nodo 5 y procedemos a etiquetarlo en forma permanente

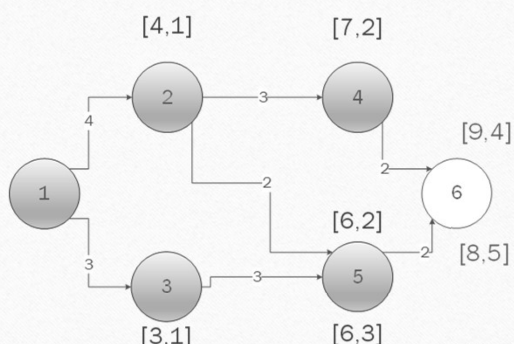
Luego etiquetamos el nodo 6



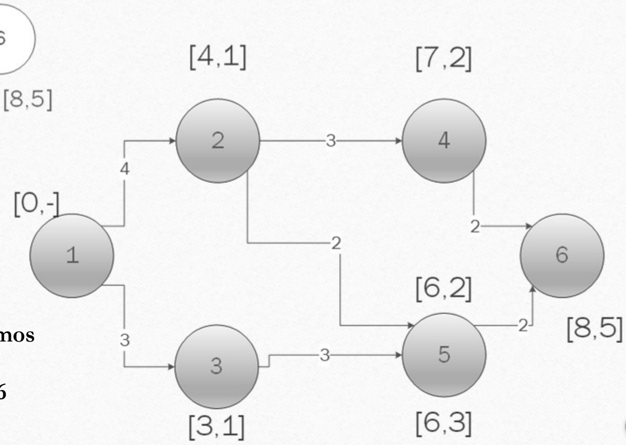


Tenemos dos etiquetas temporales la del 4 y la del 6 elegimos la menor de las dos, el 4. La hacemos permanente

Evaluamos desde 4 y etiquetamos nuevamente a 6



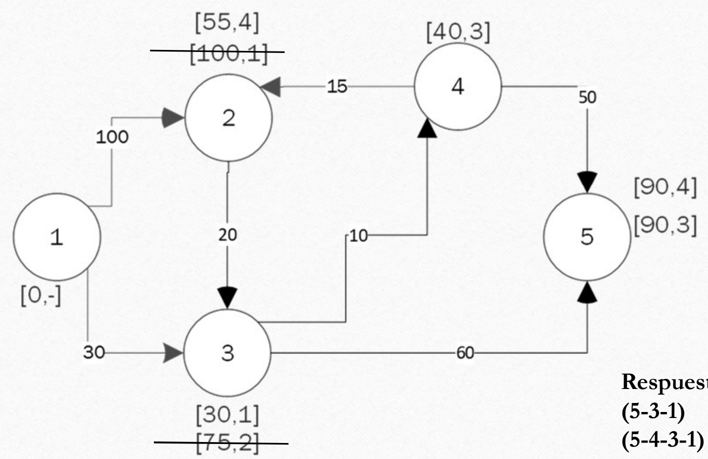
Elegimos la menor y hemos terminado con todos los nodos.



Solución: La duración mínima es 8 km, y tenemos dos rutas con igual distancia:

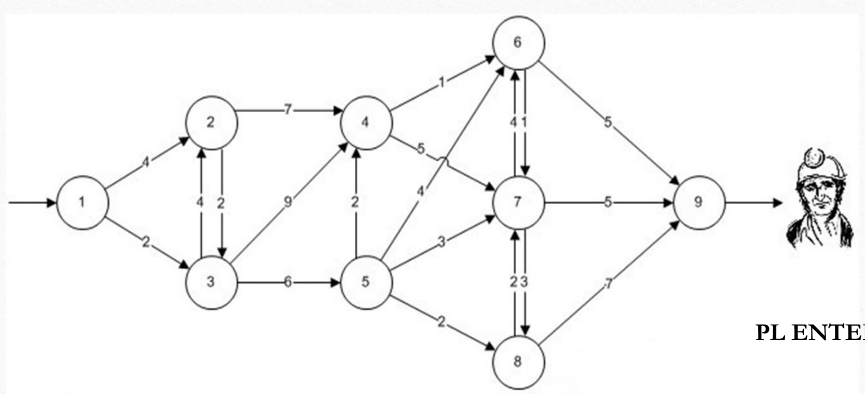
a) 1-3-5-6 y b) 1-2-5-6

OTRO EJEMPLO



Respuesta: Camino más corto 90
(5-3-1)
(5-4-3-1)

Un minero ha quedado atrapado en una mina, la entrada a la mina se encuentra ubicada en el nodo 1, se conoce de antemano que el minero permanece atrapado en el nodo 9, para llegar a dicho nodo hay que atravesar una red de túneles que van conectados entre sí. El tiempo de vida que le queda al minero sin recibir auxilio es cada vez menor y se hace indispensable hallar la ruta de acceso al nodo 9 más corta. Las distancias entre nodos de la mina se encuentran en la siguiente gráfica dadas en cientos de metros. Formule un modelo de transbordo y establecer la ruta más corta para poder así auxiliar al minero



PL ENTERA

VARIABLES DE DECISIÓN

El nombre de las variables en este caso se la puede asociar con el envío de unidades desde la entrada de la mina hacia el minero, por ende puede sugerirse este como nombre de las variables. "*Cantidad de unidades enviadas desde el nodo i hacia el nodo j*".

X_{12} = Cantidad de unidades enviadas desde el nodo 1, hacia el nodo 2
X_{13} = Cantidad de unidades enviadas desde el nodo 1, hacia el nodo 3
X_{23} = Cantidad de unidades enviadas desde el nodo 2, hacia el nodo 3
X_{24} = Cantidad de unidades enviadas desde el nodo 2, hacia el nodo 4
X_{32} = Cantidad de unidades enviadas desde el nodo 3, hacia el nodo 2
X_{34} = Cantidad de unidades enviadas desde el nodo 3, hacia el nodo 4
X_{35} = Cantidad de unidades enviadas desde el nodo 3, hacia el nodo 5
X_{46} = Cantidad de unidades enviadas desde el nodo 4, hacia el nodo 6
X_{47} = Cantidad de unidades enviadas desde el nodo 4, hacia el nodo 7
X_{54} = Cantidad de unidades enviadas desde el nodo 5, hacia el nodo 4
X_{56} = Cantidad de unidades enviadas desde el nodo 5, hacia el nodo 6
X_{57} = Cantidad de unidades enviadas desde el nodo 5, hacia el nodo 7
X_{58} = Cantidad de unidades enviadas desde el nodo 5, hacia el nodo 8
X_{67} = Cantidad de unidades enviadas desde el nodo 6, hacia el nodo 7
X_{69} = Cantidad de unidades enviadas desde el nodo 6, hacia el nodo 9
X_{76} = Cantidad de unidades enviadas desde el nodo 7, hacia el nodo 6
X_{78} = Cantidad de unidades enviadas desde el nodo 7, hacia el nodo 8
X_{79} = Cantidad de unidades enviadas desde el nodo 7, hacia el nodo 9
X_{87} = Cantidad de unidades enviadas desde el nodo 8, hacia el nodo 7
X_{89} = Cantidad de unidades enviadas desde el nodo 8, hacia el nodo 9

FUNCIÓN OBJETIVO

Min $4X_{12} + 2X_{13} + 2X_{23} + 7X_{24} + 4X_{32} + 9X_{34} + 6X_{35} + 1X_{46} + 5X_{47} + 2X_{54} + 4X_{56} + 3X_{57} + 2X_{58} + 1X_{67} + 5X_{69} + 4X_{76} + 3X_{78} + 5X_{79} + 2X_{87} + 7X_{89}$

Notación como tabla

i

	1	2	3	4	5	6	7	8	9
1	0	4	2						
2		0	2	7					
3		4	0	9	6				
4				0		1	5		
5				2	0	4	3	2	
6						0	1		5
7						4	0	3	5
8							2	0	7
9									0

La ***matriz de adyacencia*** de un grafo es simétrica. Se colocan los vértices(nodos) como filas y columnas, y colocamos 1 si hay unión o 0 si no la hay. Entonces la matriz de adyacencia contiene solo ceros y unos (matriz binaria) y la diagonal esta compuesta sólo por ceros. Nos brinda la estructura del grafo.
En este caso es nxn → donde n es la cantidad de vértices del grafo (nodos) →9x9.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	1	1	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0
4	0	0	0	0	0	1	1	0	0
5	0	0	0	1	0	1	1	1	0
6	0	0	0	0	0	0	1	0	1
7	0	0	0	0	0	1	0	1	1
8	0	0	0	0	0	0	1	0	1
9	0	0	0	0	0	0	0	0	0

El modelo lineal entonces será

Min 4X12 + 2X13 + 2X23 + 7X24 + 4X32 + 9X34 + 6X35 + 1X46 + 5X47 + 2X54 + 4X56 + 3X57 + 2X58 + 1X67 + 5X69 + 4X76 + 3X78 + 5X79 + 2X87 + 7X89
st
X12 + X13 = 1
X69 + X79 + X89 = 1

X12 + X32 - X23 - X24 = 0
X13 + X23 - X32 - X34 - X35 = 0
X24 + X34 + X54 - X46 - X47 = 0
X35 - X54 - X56 - X57 - X58 = 0
X46 + X56 + X57 - X67 - X69 = 0
X67 + X47 + X57 + X87 - X76 - X78 - X79 = 0
X78 + X58 - X89 –x87 = 0
Todas las variables enteras

En el Lindo la solución seria….

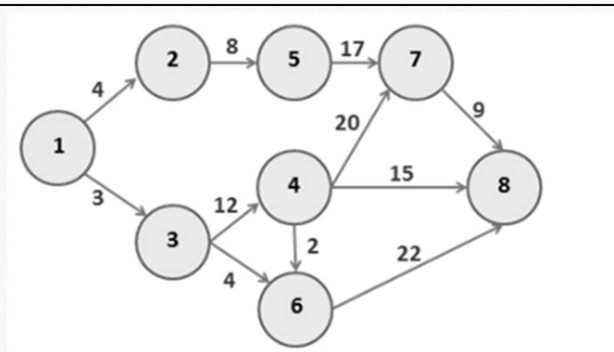
LP OPTIMUM FOUND AT STEP 9
OBJECTIVE FUNCTION VALUE

1) 16.00000

VARIABLE	VALUE	REDUCED COST
X12	0.000000	0.000000
X13	1.000000	0.000000
X23	0.000000	4.000000
X24	0.000000	1.000000
X32	0.000000	2.000000
X34	0.000000	1.000000
X35	1.000000	0.000000
X46	1.000000	0.000000
X47	0.000000	3.000000

VARIABLE	VALUE	REDUCED COST
X54	1.000000	0.000000
X56	0.000000	1.000000
X57	0.000000	4.000000
X58	0.000000	1.000000
X67	0.000000	0.000000
X69	1.000000	0.000000
X76	0.000000	0.000000
X78	0.000000	6.000000
X79	0.000000	1.000000
X87	0.000000	6.000000
X89	0.000000	0.000000

Ejemplo del problema del camino más corto con variables binarias



Variables de Decisión

$$X_{ij}: \begin{cases} 1 \ Si\ el\ nodo\ j\ es\ visitado\ inmediatamente\ después\ del\ nodo\ i \\ 0\ Si\ no \end{cases}$$

Función Objetivo: Minimizar la distancia total en [km] dada por la siguiente expresión:

$$4X_{12} + 3X_{13} + 8X_{25} + 12X_{34} + 4X_{36} + 17X_{57} + 20X_{47} + 2X_{46} + 15X_{48} + 22X_{68} + 9X_{78}$$

(1) $X_{12} + X_{13} = 1$

(2) $X_{12} - X_{25} = 0$

(3) $X_{13} - X_{34} - X_{36} = 0$

(4) $X_{25} - X_{57} = 0$

(5) $X_{34} - X_{47} - X_{48} - X_{46} = 0$

(6) $X_{36} + X_{46} - X_{68} = 0$

(7) $X_{57} + X_{47} - X_{78} = 0$

(8) $X_{78} + X_{48} + X_{68} = 1$

Modelo y resolución con LINDO

```
Min 4 x12 + 3 x13 + 8 x25 + 12 x34 + 4 x36 + 17 x57 + 20 x47+ 2 x46 + 15 x48 + 22 x68 + 9 x78
st
x12 + x13 = 1
x12 - x25 =0
x13 - x34 - x36=0
x25 - x57 =0
x34 - x47 - x48 - x46 =0
x36 + x46 - x68 =0
x57 + x47 - x78 =0
x78 + x48 +x68 =1
end
Int (11)
```

LP OPTIMUM FOUND AT STEP 1

OBJECTIVE FUNCTION VALUE

1) 29.00000

VARIABLE	VALUE	REDUCED COST
X12	0.000000	0.000000
X13	1.000000	0.000000
X25	0.000000	0.000000
X34	0.000000	0.000000
X36	1.000000	0.000000
X57	0.000000	9.000000
X47	0.000000	15.000000
X46	0.000000	10.000000
X48	0.000000	1.000000
X68	1.000000	0.000000
X78	0.000000	0.000000

Ejemplo

El siguiente grafo corresponde a una red de ordenadores en la que los nodos O1 y O2 representan los servidores de correo electrónico de los ordenadores de varias ciudades europeas, T1 y T2 son nodos de distribución intermedios y D1, D2 y D3 representan los servidores de correo que dan servicio a los ordenadores de varias ciudades americanas. Se estima que a O1 llegan 1000 mensajes al día, y 1200 a O2. Sin embargo, D1 solamente puede dar salida a 800 mensajes diarios, D2 a 900, y D3 a 500. Se supone que no hay problemas de capacidad en las líneas que conectan estos nodos, por lo que por cada tramo se pueden transmitir tantos mensajes como sean necesarios. Los números indicados en la red representan el tiempo que tarda en milésimas de segundo la transmisión del mensaje por ese tramo de la red. Modelizar y resolver el problema que permite transmitir los mensajes en el menor tiempo posible.

Min $3\,x13 + 4\,x14 + 2\,x23 + 5\,x24 + 8\,x35 + 6\,x36 + 7\,x34 + 4\,x46 + 9\,x47 + 5\,x56 + 3\,x67$

st

$x13 + x14 \le 1000$

$x23 + x24 \le 1200$

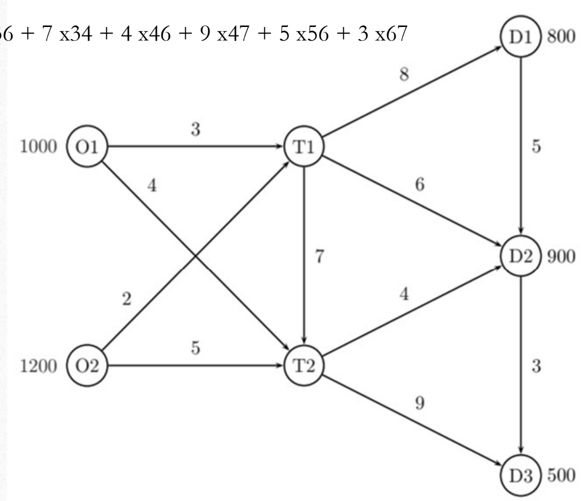
$x35 - x56 \ge 800$

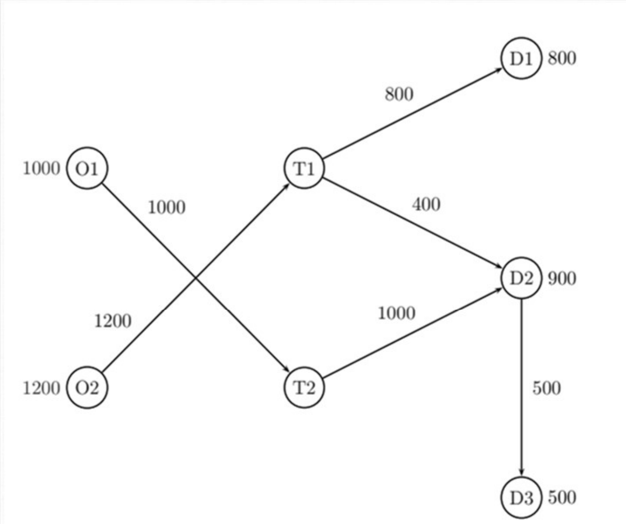
$x56 + x36 + x46 - x67 \ge 900$

$x47 + x67 \ge 500$

$x13 + x23 - x35 - x36 - x34 = 0$

$x14 + x24 + x34 - x46 - x47 = 0$





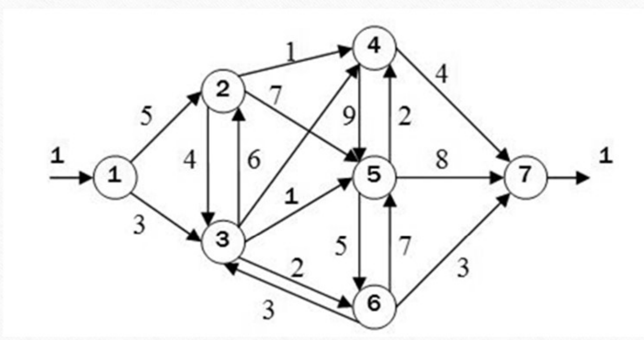
LP OPTIMUM FOUND AT STEP 1

OBJECTIVE FUNCTION VALUE

1) 20700.00

VARIABLE	VALUE	REDUCED COST
X13	0.000000	1.000000
X14	1000.000000	0.000000
X23	1200.000000	0.000000
X24	0.000000	1.000000
X35	800.000000	0.000000
X36	400.000000	0.000000
X34	0.000000	5.000000
X46	1000.000000	0.000000
X47	0.000000	2.000000
X56	0.000000	7.000000
X67	500.000000	0.000000

Practicamos…



Actividades:
1) Generar la tabla de transporte
2) Escribir el modelo lineal sin matriz de transporte, balancear los nodos
3) Aplicar Dijkstra

"

PROBLEMAS DE FLUJO MAXIMO

"

ALGORITMO DE FORD FULKERSON (1962)

El algoritmo de Ford-Fulkerson propone buscar caminos en los que se pueda aumentar el flujo, hasta que se alcance el flujo máximo. La idea es encontrar una ruta de penetración con un flujo positivo neto que una los nodos origen y destino.

Los algoritmos de flujos resuelven el problema de encontrar el flujo máximo de una fuente a un sumidero respetando una serie de restricciones.

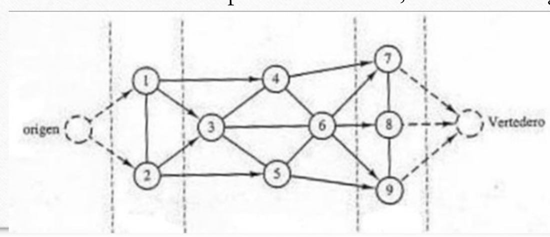
- Nodo inicial = Fuente
- Nodo Final = Sumidero
- Los flujos se miden como el flujo que sale de un nodo, si así ocurriera el flujo se considera positivo, en caso contrario tenemos un flujo negativo.
- La fuente tiene un flujo neto positivo, el sumidero tiene un flujo neto negativo, y los nodos intermedios en los caminos que van de la fuente al sumidero tienen un flujo neto igual a cero.

propiedad de conservación del flujo $\sum \boldsymbol{f_i} = \sum \boldsymbol{f_o}$ (por nodo).

En algunas redes circula por los arcos un flujo (envío o circulación de unidades homogéneas de algún producto: automóviles en una red de rutas o bits por un cable de fibra óptica) desde el origen o fuente al destino, también denominado sumidero o vertedero. Los arcos tienen una capacidad máxima de flujo, y se trata de enviar desde la fuente al sumidero la mayor cantidad posible de flujo, de tal manera que:

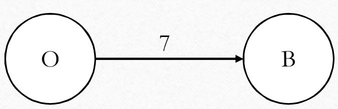
1. El flujo es siempre positivo y con unidades enteras.
2. El flujo a través de un arco es menor o igual que la capacidad.
3. El flujo que entra en un nodo es igual al que sale de él.

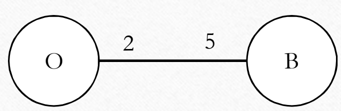
En el caso de que el origen o el destino no existan en el problema, se añaden ficticiamente utilizando arcos unidireccionales de capacidad infinita, como en grafo mostrado a continuación:



Este algoritmo se basa en dos conceptos básicos:
- Red Residual:
- Trayectoria aumentada

La red residual muestra las capacidades restantes (llamadas capacidades residuales) para asignar flujos adicionales . Ejemplo

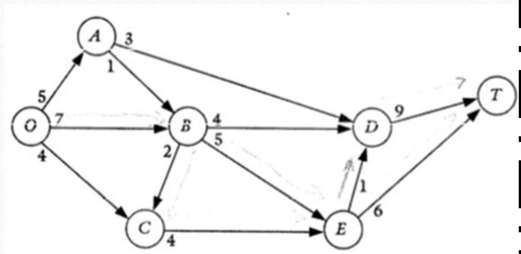


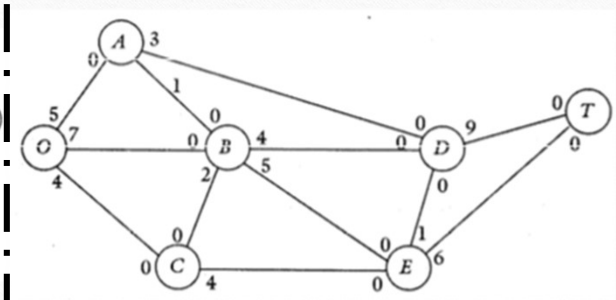


Si asignamos un flujo de 5 → Capacidad Residual es 7-5 = 2

El número sobre el arco junto a un nodo da la capacidad residual para el flujo *desde* ese nodo *hasta* el otro. Para el ejemplo sería la capacidad residual de 2 para el flujo O *a* B, el 5 de la derecha indica una capacidad residual de 5 para asignar un flujo de B hasta O

Si partimos entonces de una red dirigida podemos transformarla en una **red residual,** con arcos no dirigidos





Siempre que se asigna una cantidad de flujo a un arco, esa cantidad se ***resta*** de la capacidad residual en la misma dirección y se ***suma*** a la capacidad residual en la dirección opuesta.

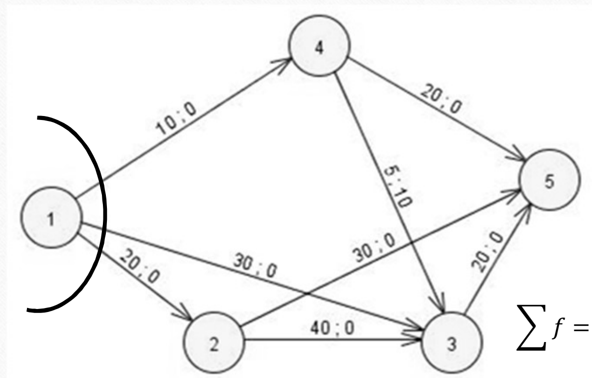
Una ***trayectoria de aumento*** es una trayectoria dirigida del nodo fuente al nodo destino en la red residual, tal que ***todos*** los arcos en esta trayectoria tienen capacidad residual *estrictamente positiva.* El *mínimo* de estas capacidades residuales se llama *capacidad residual de la trayectoria de aumento* porque representa la cantidad de flujo que es factible agregar en toda la trayectoria.

EL ALGORITMO PASO A PASO

Determinar el flujo máximo entre los nodos origen y destino de la siguiente red

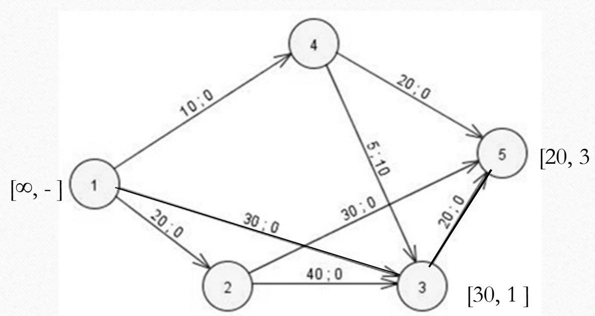
C= Capacidad
i,j= índices de los nodos
k=flujo mínimo del camino seleccionado
Cij, ji = (ci – k, Cj + k)

$\sum f = 60$

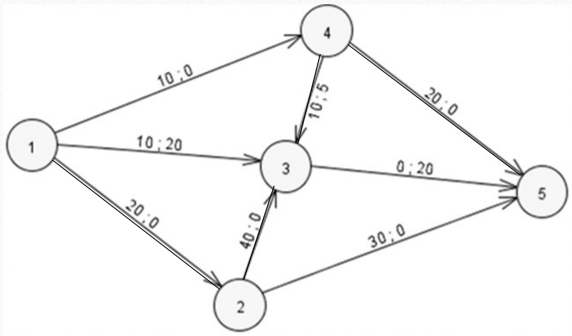


$\sum f = 0$

Pasos
1) Identificar los nodos origen y destino
2) Identificar la capacidad mas alta que sale del nodo de origen
3) Identificar el nodo intermediario con [af, i]
4) Repetir como si el nodo intermediario fuera el nodo origen

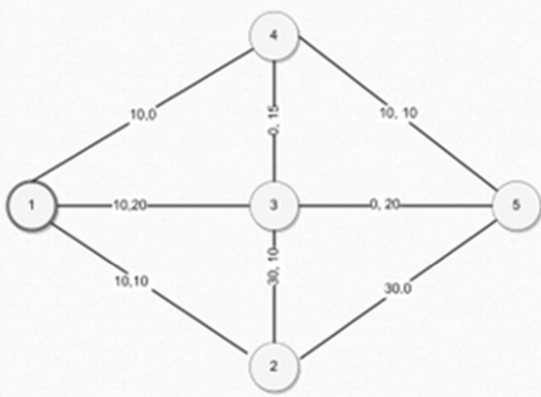


1º Iteración
- Etiquetamos el nodo origen.[af= flujo máx que puede recibir, nodo desde el que viene]
- Seleccionamos el nodo origen y elegimos el flujo máximo es decir 30, y etiquetamos
- Luego para llegar al destino volvemos a seleccionar el flujo máximo que es 20, y etiquetamos
- Escribimos y calculamos las nuevas capacidades, de acuerdo a la nomenclatura Cij, ji = (ci – k, Cj + k)
    - Calculamos k que es el mínimo de los flujos involucrados (∞, 30, 20) → k= 20
    - C13, 31 = (30 – 20 , 0 + 20) = (10 , 20)
    - C35, 53 = (20 – 20 , 0 + 20) = (0 , 20)
- Trasladamos las nuevas capacidades a la red y el resto reproducimos igual



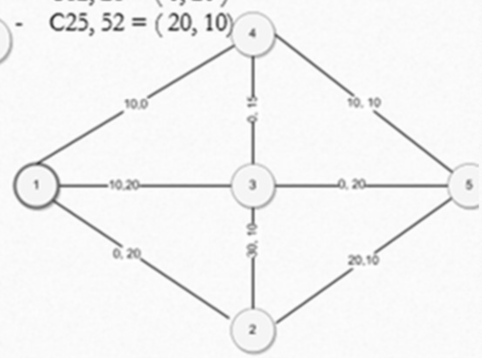
2º Iteración

- Seleccionamos el nodo origen y elegimos el flujo máximo es decir 20 en el nodo 2, luego 40 del nodo 3, después no se puede elegir 1 porque seria volver al origen entonces elegimos 10 del nodo 4 y por último nodo 5 con 20.
- Escribimos y calculamos las nuevas capacidades:
  - Calculamos k que es el mínimo de los flujos involucrados (20, 40, 10, 20) → k= 10
  - C12, 21 = (20 – 10 , 0 + 10) = (10 , 10)
  - C23, 32 = (40 – 10 , 0 + 10) = (30 , 10)
  - C34, 43 = (10 – 10 , 5 + 10) = (0 , 15)
  - C45, 54 = (20 – 10 , 0 + 10) = (10 , 10)
- Trasladamos las nuevas capacidades a la red y el resto reproducimos igual



3º Iteración

- Seleccionamos el camino 1-2-5
- Min (10,30) k= 10
- Escribimos y calculamos las nuevas capacidades:
  - C12, 21 = (0, 20)
  - C25, 52 = (20, 10)

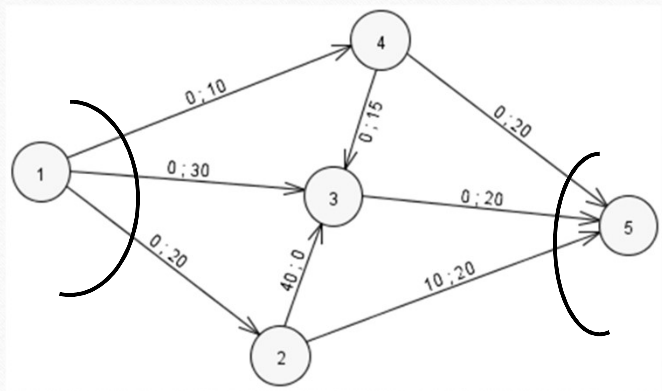


<u>4º Iteración</u>
- Seleccionamos el camino 1-3-2-5
- Min (10,10,20) k= 10

5° Iteración
- Seleccionamos 1-4-5
- Min (10,10) k=10

<u>La red después de la última Iteración</u>

$\sum f = 0$



$\sum f = 60$

Flujo máximo con programación lineal

Como ejemplo del problema de flujo máximo, considere el sistema de carreteras interestatales de norte a sur que pasan por Cincinnati, Ohio. El flujo vehicular de norte a sur alcanza un nivel de 15,000 vehículos por hora en horas críticas. Debido a un programa de mantenimiento de carreteras de verano, el cual exige el cierre temporal de carriles y límites de velocidad específicos, el comité de planeación de transporte ha propuesto una red de rutas alterna a través de Cincinnati. Las rutas alternas incluyen otras carreteras. Debido a las diferencias en los límites de velocidad y los patrones de tránsito, las capacidades de flujo varían en función de las calles y carreteras usadas en particular. La red propuesta con las capacidades de flujo de arcos aparece en la fi gura 10.16. En esta figura se indica la dirección del flujo para cada arco y se muestra la capacidad al lado de cada arco. Observe que la mayoría de las calles es de un sentido; no obstante, se puede encontrar una calle de dos sentidos entre los nodos 2 y 3 y 5 y 6. En ambos casos, la capacidad es la misma en cada dirección.

FIGURA 10.16 RED DEL SISTEMA DE CARRETERAS Y CAPACIDADES DE FLUJO (1000/HORA) PARA CINCINNATI

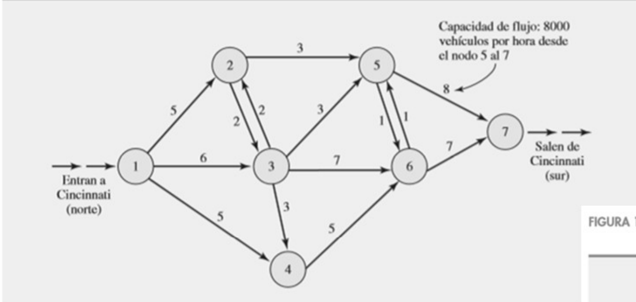
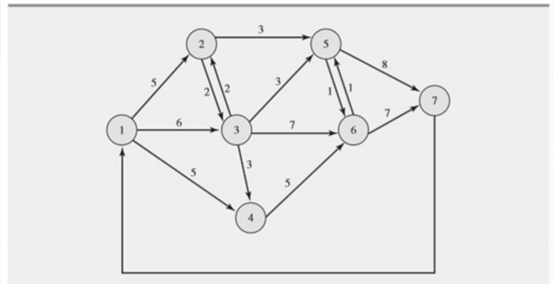


FIGURA 10.17 FLUJO POR EL ARCO DEL NODO 7 AL 1, PARA REPRESENTAR EL FLUJO TOTAL POR EL SISTEMA DE CARRETERAS DE CINCINNATI



VARIABLES DE DECISION

$$X_{IJ} = cantidad\ de\ flujo\ de\ tráfico\ desde\ el\ nodo\ i\ al\ j$$

FUNCION OBJETIVO

$$MAX\ x_{71}$$

RESTRICCIONES

Como los problemas de transbordo y la condición de conservación de los flujos Tendremos una restricción por cada nodo

$$x_{12} + x_{13} + x_{14} = x_{71}$$
$$x_{23} + x_{25} = x_{32} + x_{12}$$

....

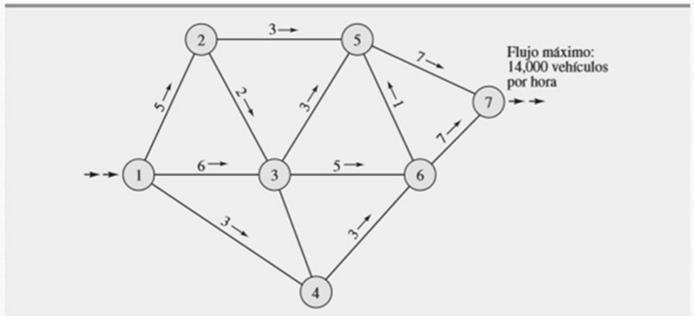
Agregamos restricciones de las capacidades de los arcos

$$x_{12} \leq 5$$
$$x_{13} \leq 6$$
$$x_{14} \leq 5$$
$$x_{23} \leq 2$$

…

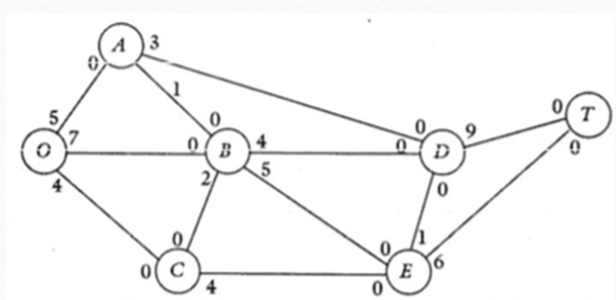
Solución….

FIGURA 10.19 RED DE CARRETERAS PARA EL PROBLEMA DE LA RUTA MÁS CORTA DE GORMAN CONSTRUCTION



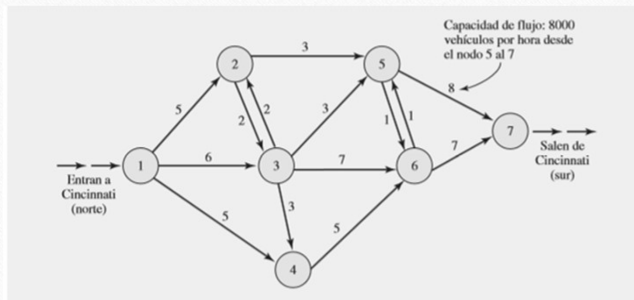
ACTIVIDAD 1

- Aplicar el algoritmo de flujo máximo a la red del ejemplo
- Escribir el modelo lineal
- Contrastar los resultados



ACTIVIDAD 2

- Tomar el problema de las carreteras, ya resuelto por PL, y generar la red residual
- Aplicar el algoritmo de Flujo Máximo
- Contrastar resultados



SOLUCION

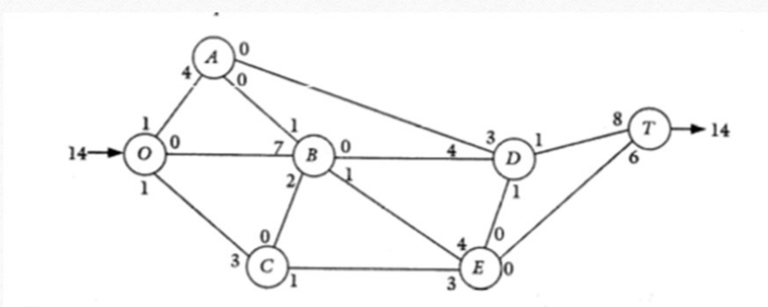


FIGURA 10.19 RED DE CARRETERAS PARA EL PROBLEMA DE LA RUTA MÁS CORTA DE GORMAN CONSTRUCTION