Optimización en Redes

Dpto. Ingeniería Industrial, Universidad de Chile

IN3701, Optimización

Contenidos

- Grafos
- Algoritmos específicos para grafos

Introducción

- Es natural definir problemas sobre redes
- Muchos problemas se pueden modelar de esta forma
- Si bien SIMPLEX ofrece un algoritmo para resolverlos, existen mejores algoritmos: mejores en teoría, en la practica y que implican menos requerimientos computacionales

- Un **grafo** es una estructura matemática compuesta de un conjunto de puntos denominados **nodos** o **vértices** y un conjunto de trazos que unen los nodos llamados **arcos** o **aristas**
- **Grafo no dirigido (no orientado)**:
  - Un grafo no dirigido G=(V,E) es un conjunto de nodos V y un conjunto de arcos E (que son pares ordenados de nodos)
  - El arco u={i,j} ϵ E se dice incidente a i y a j
- **Grafo dirigido (orientado)**:
  - El concepto es el mismo de antes, salvo que ahora el arco {i,j} tiene dirección, es decir, va estrictamente de i a j
- **Grado de un nodo i:** corresponde al número de arcos que son incidentes en i

- Para un grafo G=[V, E] usualmente denotamos $|V|=n$ y $|E|=m$. Y decimos que el **Tamaño** del grafo es (n.m)
- **Subgrafo:** Dado un grafo G=[V,E], un subgrafo de G es G´=[V´,E´] tal que $V' \subseteq V$ y $E \subseteq E'$
- **Camino** (en grafos no orientados): un camino del nodo i_1 a i_t es una secuencia finita de nodos $i_1, i_2,\ldots,i_t$ que nos llevan de i_1 a i_t tal que el arco $(i_k, i_{k+1}) \in E \ \forall\ k=1,\ldots,t-1$
- **Camino** (en grafos orientados): Es una secuencia de nodos $i_1, i_2,\ldots,i_t$ con una correspondiente secuencia de arcos $a_1, a_2,\ldots,a_{t-1}$ tal que para k= 1,2,..,t-1 esta presente ya sea el arco:
  - $a_k = (i_k, i_{k+1})$ en cuyo caso decimos que es un arco hacia adelante
  - $a_k = (i_{k+1}, i_k)$ en cuyo caso decimos que es un arco hacia atrás

  -> Un camino es un camino (o ruta) hacia adelante si todos sus arcos son hacia adelante

  -> Un camino es un camino (o ruta) hacia atrás si t sus arcos son hacia atras

- **Ciclo:** Es un camino donde el nodo final y el inicial coinciden
- **Conexo:** Un grafo es conexo si existe un camino entre cada par de nodos
- **Árbol:** Es un grafo conexo y sin ciclos

Propiedades de un árbol:

- Un árbol de n nodos tiene exactamente n-1 arcos
- Un árbol tiene al menos 2 nodos con grado igual a 1
- Cada par de nodos de un árbol esta conectado por un único camino

- Dado un Grafo G=[V,E] se denomina **Flujo** a la función que hace corresponder a cada arco (i,j) del grafo un valor real f_{ij}

- **Divergencia:** Es todo el flujo que sale de un nodo hacia otros nodos menos todo el flujo que entra a ese nodo desde otros nodos

$$y_i = \sum_{j \text{ tal que } (i,j) \in E} f_{ij} - \sum_{j \text{ tal que } (j,i) \in E} f_{ji}$$

Si es positiva al nodo se le dice fuente, si es negativa sumidero y si es cero nodo de transferencia

Por su parte un **flujo exógeno** es aquel que proviene o termina en el exterior del grafo

- **Red:** es un sistema formado por un grafo G=[V,E] y una función de flujos

Problema de Flujo en redes

- Este problema esta caracterizado por una serie de restricciones :

- Limite superior e inferior del flujo
- Conservación de flujo

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$l_{ij} \le x_{ij} \le u_{ij}$$

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = q_i \ \forall \mathrm{i}$$

- q_i es igual a cero cuando no hay flujo exógeno. Es igual al valor del flujo exógeno cuando este entra al nodo e igual al valor pero con signo negativo si sale del nodo

Problema de Distancia Mínima

- Dado un grafo dirigido G=[V,E] queremos encontrar la distancia mínima desde un nodo s a un nodo t

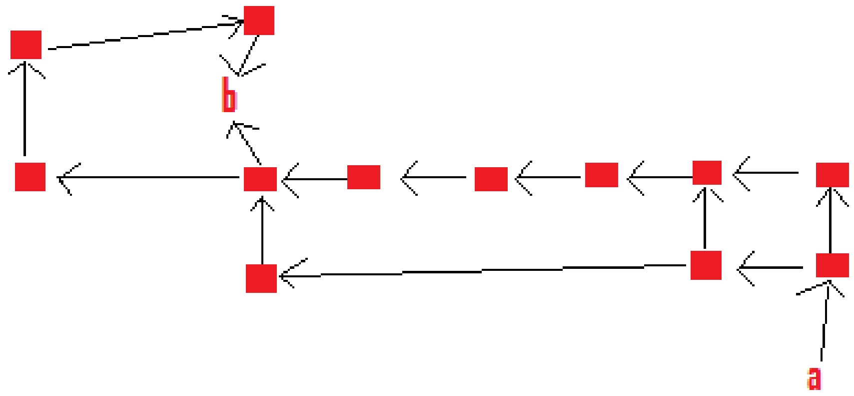
Opción 1: PPL

Opción 2: Otro algoritmo

Algoritmo Dijkstra

Queremos ir de Beauchef 850 a Av. Republica 701 caminando la menor distancia posible

- El algoritmo opera sobre un grafo en el cual cada arco tiene asociados costos, los cuales pueden representar tiempo, distancia o dinero.
- Para nuestro ejemplo debemos convertir el mapa en un grafo (en problemas más grandes esto puede ser difícil)



- Arriba se ve nuestro problema transformado en un grafo, nos faltaría calcular los costos de los arcos.

Es esto ultimo fácil de hacer?

Por ejemplo si estamos decidiendo el camino más rápido para una empresa de reparto, los tiempos de traslado son siempre los mismos?

- Pero volvamos a nuestro algoritmo…

- Sea el grafo orientado G=[V,E] y la función de costos $l:A \rightarrow R^+$ *(OJO: se requiere que el grafo sea orientado y la función de costos entregue valores positivos)*
- Vamos a definir un conjunto S que inicialmente va a ser vacio, el operador л(i) que nos va a decir la menor distancia para llegar a i y P(i) que va a simbolizar el nodo por el que pasamos antes de llegar a i
- Luego:

1. Se define el conjunto S= vacio, junto con л(1)=0, л(i)=+∞ ∀i ≠ 1 y P(1)=1
2. Repetir hasta S=N:

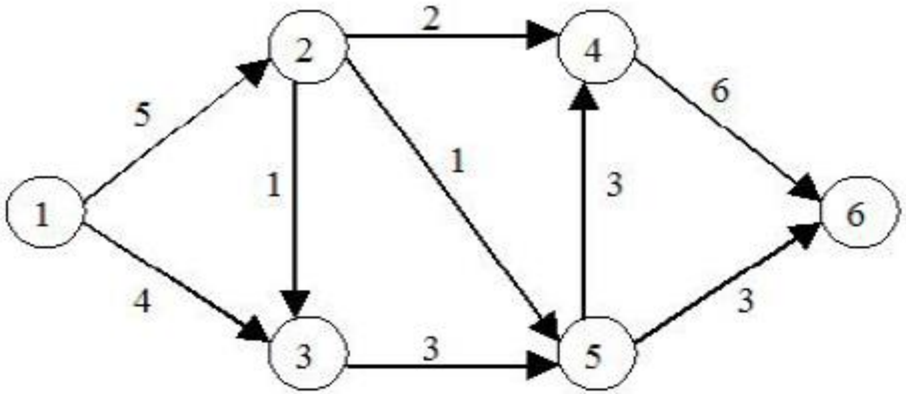
Encontrar j no perteneciente a S tal que л(j) sea minimo. Luego incorporar ese nodo a S, ie, S=S U {j}

Para cada i tal que existe un arco de j a i, con i NO perteneciente a S, hacer:

Si л(i) > л(j) + l(i,j) entonces л(i) = л(j) + l(i,j) y P(i) se actualiza a P(i) = j

- **EJEMPLO:** Radio Pizza, promete que si no llega en 27 minutos la pizza es gratis. Para cumplir con esta promesa, la empresa cuenta con un sofisticado sistema de optimización el cual se basa en el algoritmo Dijkstra. Lamentablemente el software de la compañía esta caído y es por ello que se le solicita a usted que resuelva el siguiente problema:

Encontrar la ruta mínima desde el local (nodo 1) a los distintos lugares de pedido (todos los otros nodos)



Problema de Flujo Máximo

- Deseamos transportar la mayor cantidad de flujo desde un nodo inicial a un nodo final. Cada arco por el que transportamos los flujos tienen una capacidad máxima y una mínima
- Nuevamente Podríamos plantear el PPL, pero puede haber una mejor opción

- **Corte:** Un corte Q en un grafo dirigido G=[V,E] es una partición del conjunto de nodos en 2 subconjuntos no vacios → Q=[S, V-S].

El corte nos genera 2 conjuntos de arcos:

$Q^+=\{(i,j) \in E : i \in S, j \in V\text{-}S\}$

$Q^-=\{(i,j) \in E : i \in V\text{-}S, j \in S\}$

Un corte Q separa al nodo s del nodo t si: s ∈ S y t ∈ V-S

- Dado un flujo x para la red, se define el **FLUX** a través del corte Q como el flujo total saliendo de S:

$$F(Q) = \sum_{(i,j)\in Q^+} x_{ij} - \sum_{(i,j)\in Q^-} x_{ij}$$

- Se puede demostrar que $F(Q) = \sum_{i\in S} y_i$

- Luego dado l_{ij} y u_{ij} capacidades mínimas y máximas de un arco. La **Capacidad** de un corte Q esta dada por:

$$C(Q) = \sum_{(i,j)\in Q^+} u_{ij} - \sum_{(i,j)\in Q^-} l_{ij}$$

$$F(Q) = \sum_{(i,j)\in Q^+} x_{ij} - \sum_{(i,j)\in Q^-} x_{ij} \qquad C(Q) = \sum_{(i,j)\in Q^+} u_{ij} - \sum_{(i,j)\in Q^-} l_{ij}$$

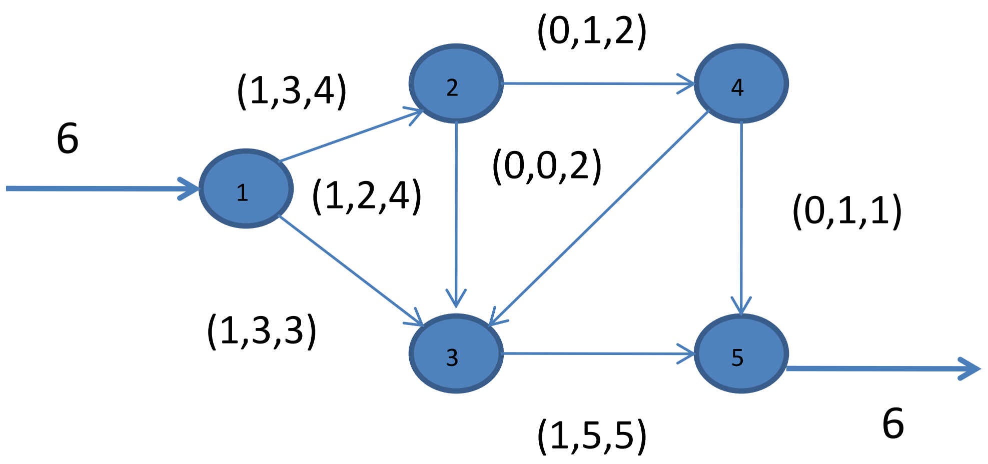
- Claramente F(Q) ≤ C(Q)
- Si F(Q) = C(Q) diremos que Q esta saturado con respecto al flujo x (el flujo de cada arco perteneciente a Q^+ debe estar en su cota superior y el perteneciente a Q^- en su cota inferior)

- Si X* son los flujos óptimos que pasan por los distintos arcos, permitiendo generar un flujo máximo igual a F* y sea Q un corte que separa s de t, se tendrá que:

F(Q)= F* ≤ C(Q) ∀ Q que separa el nodo inicial s del nodo final t

En particular se alcanzará la igualdad para el caso del corte de capacidad mínima que separa s de t

- **Ejemplo:** (l_{ij}, f_{ij}, u_{ij})



- Si Q=[(1,2,4),(3,5)] F(Q)=3+2+0+1-0=6 C(Q)=3+4+2+1=10
- Si Q=[(1,2,3,4),(5)] F(Q)=5+1-0=& C(Q)=5+1=6

EL ALGORITMO DE FORD Y FULKERSON (F&F)

0. Inicialización: determinar flujo factible f (si $l_{ij}=0\ \forall(i,j) \in E$ $f_{ij}=0\ \forall(i,j) \in E$ es flujo inicial factible admisible)

1. Construir Grafo auxiliar $G'=[V', E']$ donde $V=V'$ Para definir E' se hace lo siguiente para cada arco $(i,j) \in E$:

- si $f_{ij} < u_{ij}$ entonces el arco (i,j) se incorpora a E' con flujo $f'_{ij}= u_{ij}-f_{ij}$ Además (i,j) se agrega también al conjunto E'^{+}
- si $f_{ij} > l_{ij}$ entonces un arco (j,i) es incorporado a E' con flujo $f'_{ji}= f_{ij}-l_{ij}$ Además (j,i) se agrega también al conjunto E'^{-}

2. Determinar un camino hacia adelante C' en G' desde el nodo inicial al final. Si no existe camino hacia adelante el algoritmo termina, de lo contrario determinar el mínimo flujo ε que atraviesa alguno de los arcos del camino

$$\varepsilon = \min\{f'ij : (i,j) \in C'\}$$

3. Retornar a la red inicial y actualizar el flujo de la siguiente forma:

$F \leftarrow F + \varepsilon$

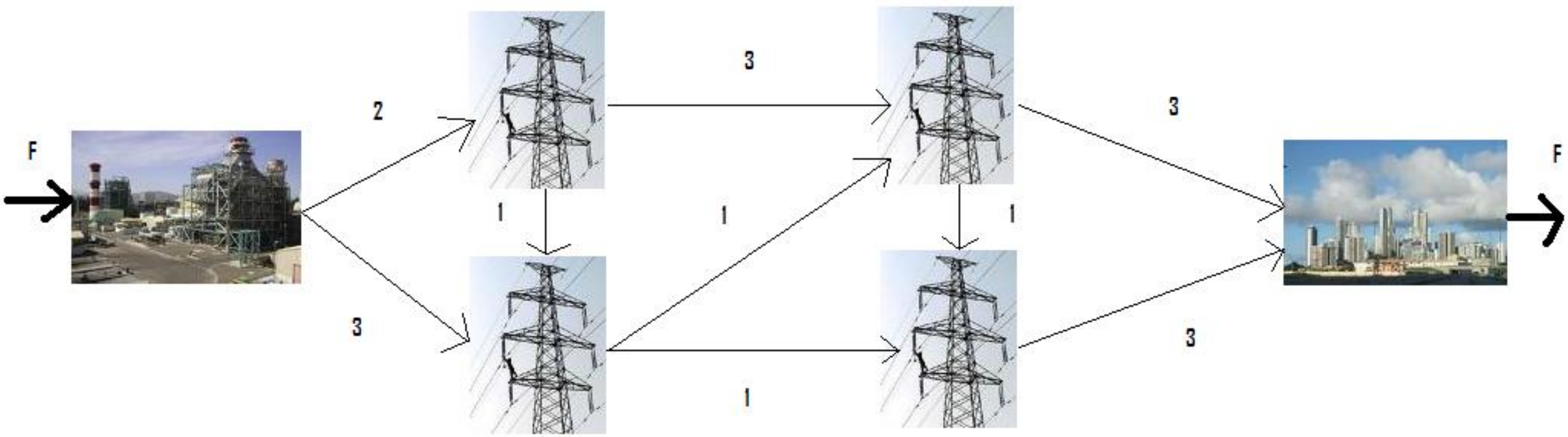
$F_{ij} \leftarrow f_{ij} + \varepsilon$ si $(i,j) \in C' \cap E'^{+}$

$f_{ij} \leftarrow f_{ij} - \varepsilon$ si $(i,j) \in C' \cap E'^{-}$

Volver a 1.

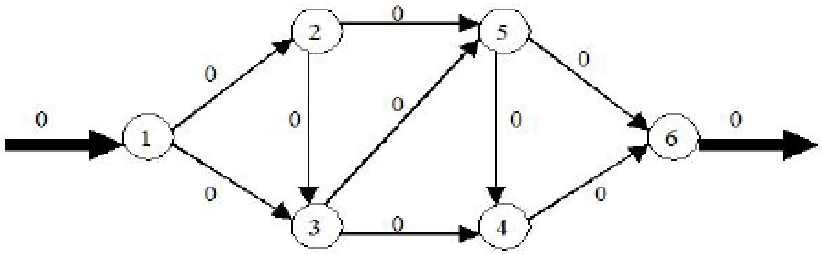
- # EJEMPLO

El valor sobre los arcos es u_{ij}



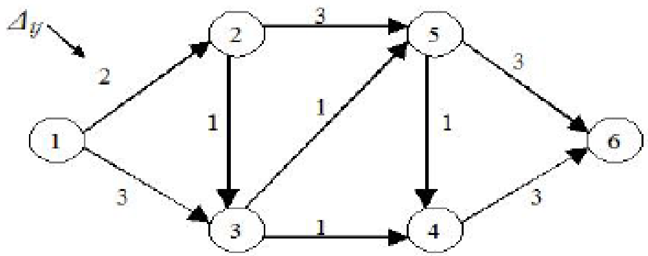
- ## **Solución**

Iteración 1:



G

El grafo auxiliar G' sería:



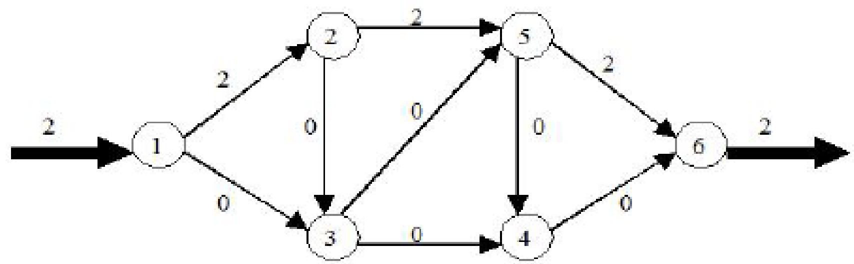
G'

Tomemos $C = 1 - 2 - 5 - 6$, vemos que lo máximo que puede aumentar el flujo es $\varepsilon = 2$.

- ## Solución

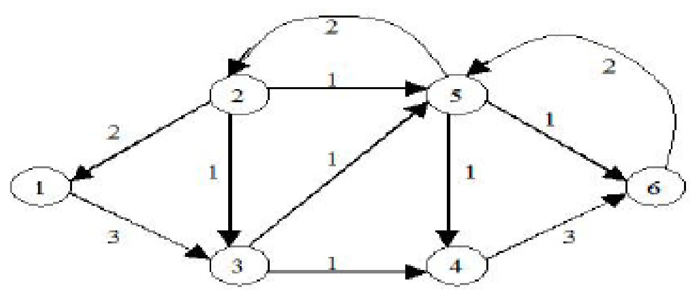
Iteración 2:

El nuevo flujo queda:



G

El grafo auxiliar queda:

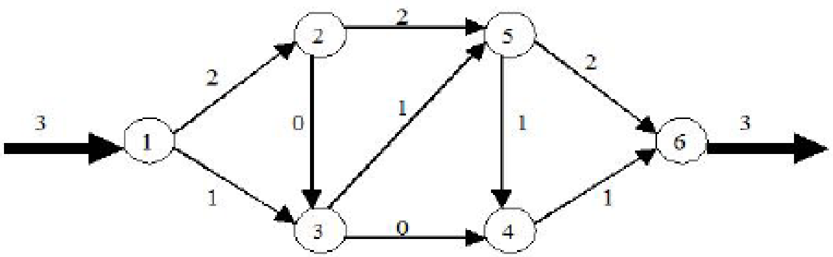


G'

Tomemos $C = 1 - 3 - 5 - 4 - 6$, vemos que $\varepsilon = 1$.

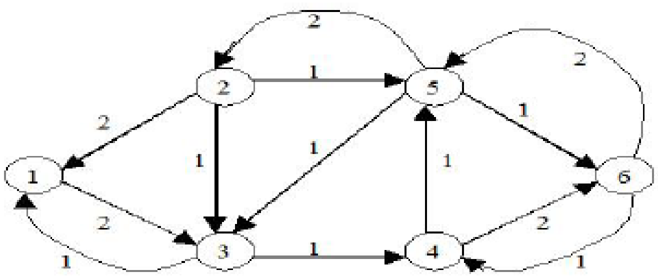
- **Solución**

Iteración 3:
El nuevo flujo queda:



G

El grafo auxiliar queda:



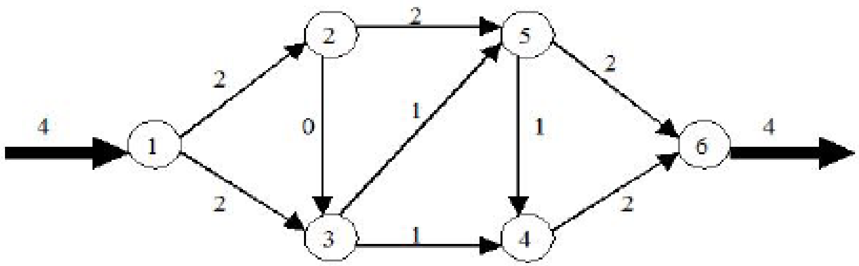
G'

Tomemos $C = 1 - 3 - 4 - 6$, vemos que $\varepsilon = 1$.

- ## Solución

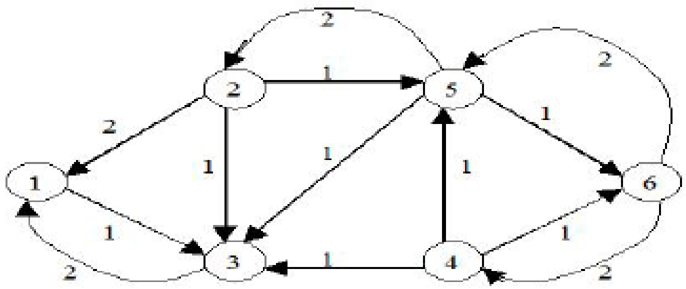
Iteración 4:

El nuevo flujo queda:



G

El grafo auxiliar queda:



G'

En este caso no existe C por lo que llegamos al óptimo, siendo 4 el flujo máximo.

Cómo encontrar un flujo inicial factible cuando no ocurre que $l_{ij}=0$ $\forall(i,j)$?

1. Se agrega al grafo $G=[V,E]$ un arco que vaya del nodo destino (t) al nodo inicial (s)
2. Se agrega un nodo artificial origen a y un nodo artificial destino b
3. Por cada nodo $i \in V$ se agregan 2 arcos: (a,i) y (i,b)
4. Las capacidades inferiores (l_{ij}) de los arcos de este nuevo grafo $G^*=[V^*,E^*]$ se definen iguales a 0. Y las superiores se definen como sigue:

$u^*_{ij}=u_{ij}-l_{ij}$ para $(i,j) \in E$

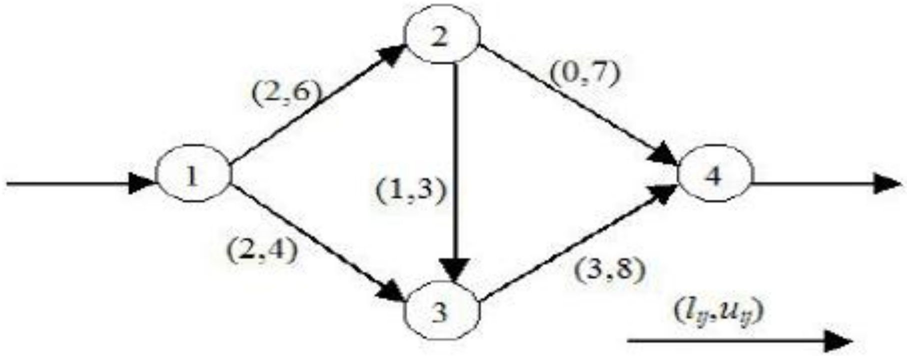
$u^*_{ai} = \sum_{k\in V} l_{ki}$ para (a,i) tal que $i \in V$

$u^*_{ib}= \sum_{k\in V} l_{ik}$ para (i,b) tal que $i \in V$

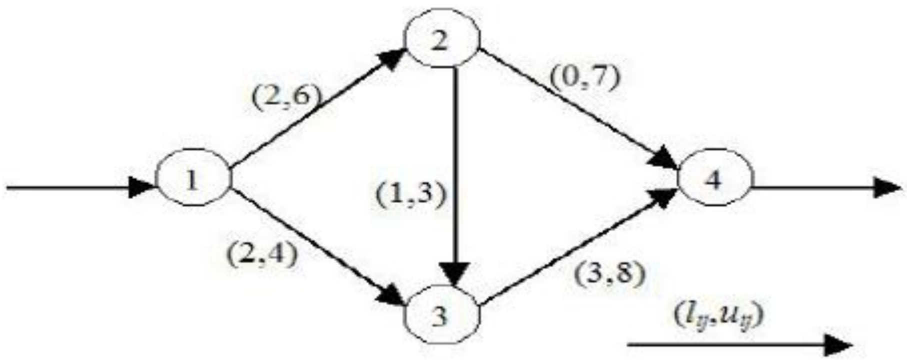
$u^*_{ts}= +\infty$

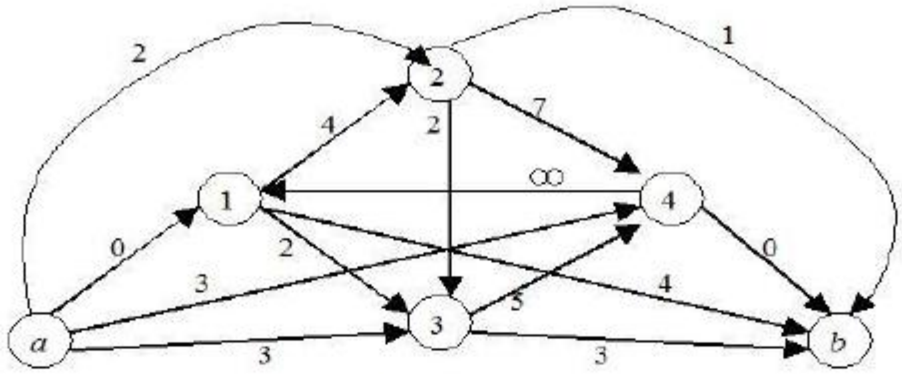
5. Luego se aplica F&F sobre la red G*, partiendo con F=0
6. Si F* (flujo máximo de la nueva red) < $\sum_{(i,j)\in E} l_{ij}$ entonces la red original no admite flujo inicial factible. En caso contrario el vector de flujos $f'_{ij}= f^*_{ij} + l_{ij}$ para todo $(i,j) \in E$, es un vector de flujo inicial factible para la red original

- EJEMPLO



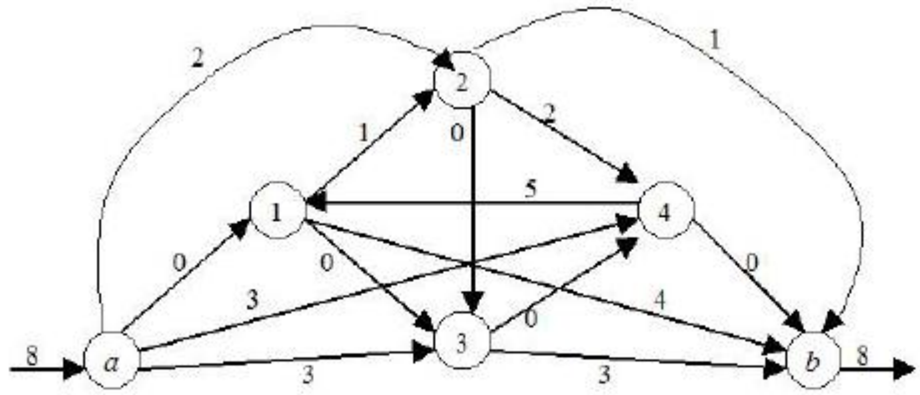
- EJEMPLO





G^*

- EJEMPLO



- F*=8. Lo que equivale a la suma de las capacidades inferiores de los arcos de la red original

$$
\begin{aligned}
f'_{12} &= f^*_{12} + l_{12} = 1 + 2 = 3\\
f'_{13} &= f^*_{13} + l_{13} = 0 + 2 = 2\\
f'_{23} &= f^*_{23} + l_{23} = 0 + 1 = 1\\
f'_{24} &= f^*_{24} + l_{24} = 2 + 0 = 2\\
f'_{34} &= f^*_{34} + l_{34} = 0 + 3 = 3
\end{aligned}
$$

- Comentarios finales
  - La utilización de algoritmos como Dijkstra y F&F puede ser muy útil para dar una solución más rápida a este tipo de problemas
  - Muchas veces problemas que no tienen las características del problema estándar de flujo en redes son transformados en este tipo de problemas por las ventajas de resolución que esto implica

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = q_i \ \forall i$$

El problema estándar de Flujo

- Ejemplo: Qué pasa si el flujo total que sale de un nodo también esta acotado inferior y superiormente? Cómo traducimos esta situación a la de un problema estándar de flujo en redes?