CAPÍTULO 5

FLUJO EN REDES

Flujo en Redes

Flujo en redes corresponde al problema de pasar un flujo por una red de nodos interconectados.

Los problemas de Flujos en Redes son comunes en el área de transporte, comunicaciones, electricidad e informática.

Para resolver el problema de flujo en redes se utiliza la **teoría de Grafos**, cuyo pionero fue Leonhard Euler (1736).

En este capítulo veremos el problema del flujo máximo y de la ruta más corta, a partir de los cuales se puede obtener solución a otros problemas de flujo en redes.

Conceptos Básicos

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 denominados **arcos** o **aristas**.

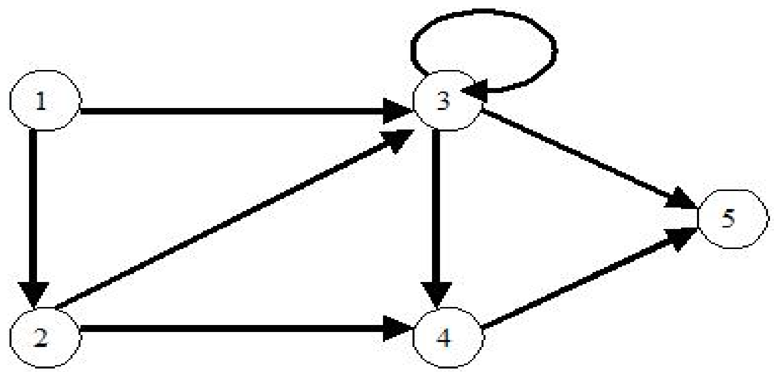
Si los arcos son flechas el grafo es **orientado**, si corresponde solo a lineas es un grafo **no orientado**.
En este capítulo trabajaremos con grafos orientados.

Veamos formalmente como se define un grafo orientado:
Definición:
Un grafo orientado $G = [N, A]$ es un sistema formado por un conjunto finito no vacío $N = \{1, 2, \ldots, n\}$ cuyos elementos se llaman nodos y un conjunto A de pares $ordenados$ de elementos de N llamados arcos.

Observación: Se denota $|N| = n$ y $|A| = m$.

Conceptos Básicos

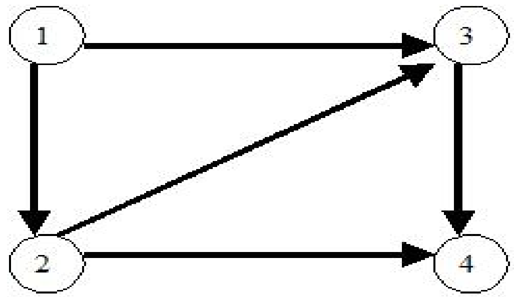
Ej: $G = [N, A]$ con $N = \{1, 2, 3, 4, 5\}$ y
$A = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 3), (3, 4), (3, 5), (4, 5)\}$



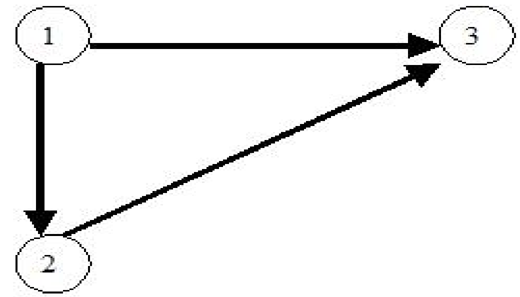
Observación: El arco (i, j) denota un arco que va desde el nodo i al nodo j. Este arco se denomina $incidente$ en los nodos i y j. Además i y j se denominan $nodos$ $adyacentes$ dado que existe un arco entre ellos.

Definición:
Dado un grafo $G = [N, A]$, un **subgrafo** es un grafo $G' = [N', A']$ donde $N' \subseteq N$ y $A' \subseteq A$.

Conceptos Básicos



G



Subgrafo de G

Definición:
Un **camino** entre dos nodos i y j, en un grafo orientado G, es una secuencia de nodos y arcos de G que comunica los 2 nodos.

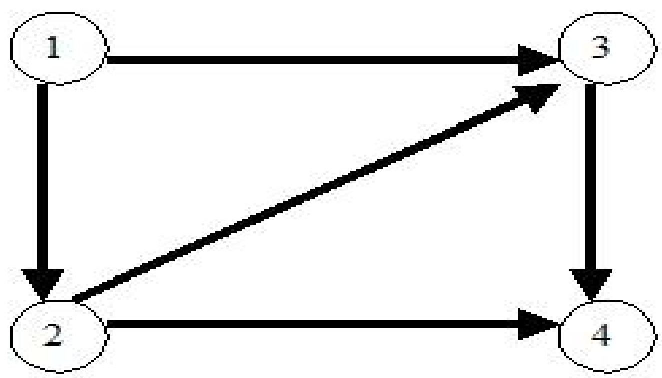
Observación: Si los nodos l y k pertenecen al camino definido anteriormente y son consecutivos en la secuencia, entonces el arco (l, k) o el (k, l) pertenece al camino. Si el arco (l, k) pertenece al camino, se dice que es un arco $hacia\ adelante$ y si el arco (k, l) pertenece al camino, se dice que es un arco $hacia\ atrás$ o en reversa. Estrictamente una secuencia que solo contenga arcos hacia adelante se denomina camino y una secuencia que contiene arcos hacia adelante y hacia atrás se denomina **cadena**, sin embargo, en este capítulo se hablará indistintamente de camino en ambos casos.

Conceptos Básicos

Definición:
Un **circuito** en un grafo orientado es un camino donde el nodo inicial y el nodo final coinciden.

Observación: Un circuito elemental es aquel circuito que sólo posee un nodo. Por lo tanto, posee solo un arco que comienza y termina en el mismo nodo.

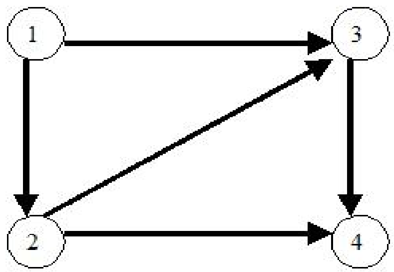
Ej:



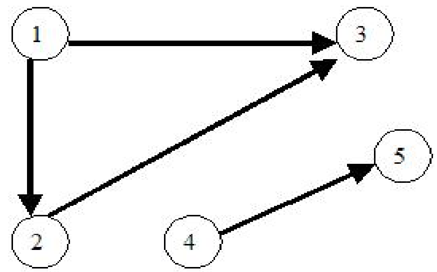
En la figura, los arcos $(1,2)$ y $(1,3)$ corresponden a un camino entre los nodos 1 y 3, mientras que los arcos $(1,2)$, $(2,4)$, $(3,4)$ y $(1,3)$ forman un circuito.

Conceptos Básicos

Definición:
Un grafo es **conexo** si existe un camino entre cada par de nodos.

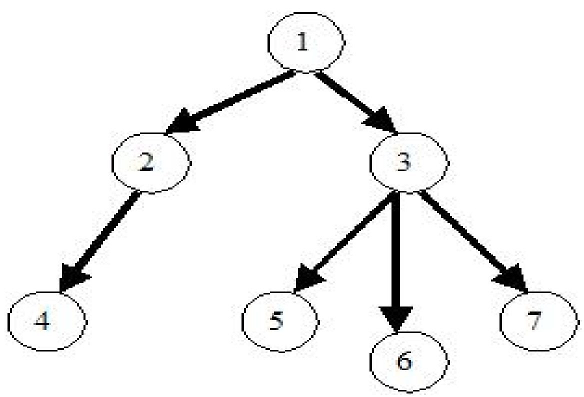


Grafo Conexo



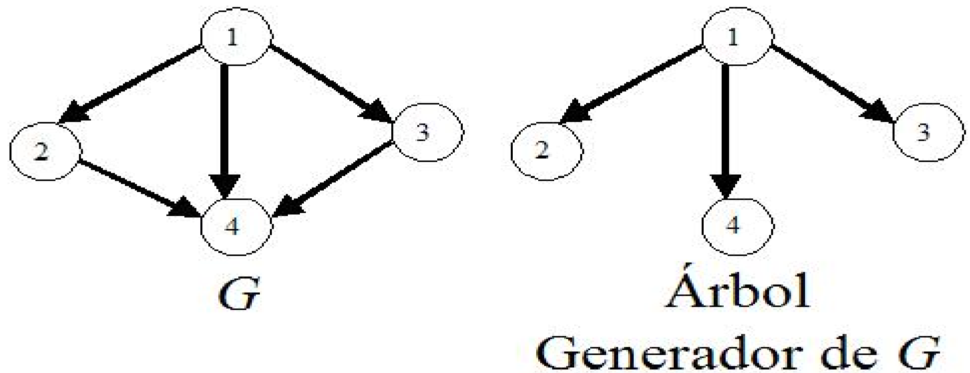
Grafo No Conexo

Definición:
Un **árbol** es un grafo conexo y sin circuitos.



Conceptos Básicos

Definición:
Se denomina **árbol generador** de $G = [N, A]$ a un subgrafo $G' = [N, A']$ de G tal que G' es un árbol.



Observación: Notar que el conjunto de nodos de G y del árbol generador de G debe ser el mismo.

Propiedades de un Árbol:

1. Contiene $n-1$ arcos.

2. Existe un único camino entre cada par de nodos.

Conceptos Básicos

3. Si se elimina un arco, se obtiene un grafo no conexo compuesto por 2 árboles.

4. Si se agrega un arco cualquiera, se forma un circuito único, obteniendose un grafo que no es un árbol.

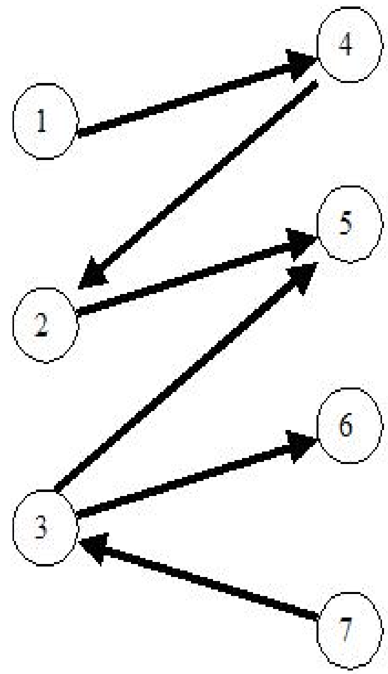
5. Existe al menos un nodo cuyo grado (cantidad de arcos que inciden en un nodo) es igual a 1.

Definición:
Grafo Completo es áquel en que cada par de nodos está conectado por un arco.

Definición:
Un grafo es **bipartito** si el conjunto N de nodos se puede particionar en 2 conjuntos N_1 y N_2, es decir, $N = N_1 \bigcup N_2$, $N_1 \bigcap N_2 = \phi$, de modo que cada arco del grafo tenga un nodo en N_1 y otro en N_2

Observación: En un grafo bipartito no existen arcos con nodo inicial y final en el mismo conjunto de la partición.

Conceptos Básicos



Representación de un Grafo

Al tener un grafo con muchos nodos se vuelve inmanejable su representación gráfica.

Además existen muchas formas gráficas de representar un grafo, algunas entregan soluciones más claras que otras.

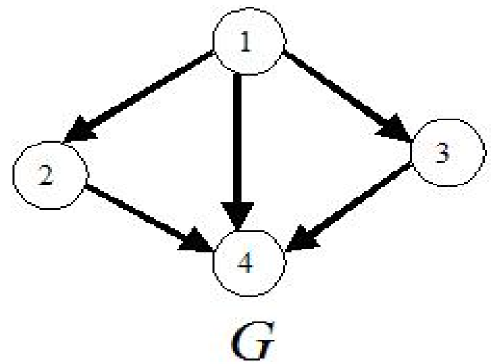
Se mostrará una forma de representar un grafo prescindiendo de su representación gráfica.

Definición:
Una **matriz de incidencia** E de un grafo es una matriz que refleja cómo inciden los arcos en los nodos, conteniendo en cada fila un nodo y en cada columna un arco del grafo que representan.
Sus valores se construyen de la siguiente forma:
Sea a un arco del grafo G, entonces se define:

$$E = (e_{ka}) = \begin{cases} 1 & \text{si el arco } a \text{ comienza en el nodo } k \\ -1 & \text{si el arco } a \text{ termina en el nodo } k \\ 0 & \text{en otro caso} \end{cases}$$

Matriz de Incidencia

Ej:



	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)
1	1	1	0	0	0
2	-1	0	1	1	0
3	0	-1	-1	0	1
4	0	0	0	-1	-1

Matriz de Incidencia

Teorema:
Si E es la matriz de incidencia nodo-arco de un grafo conexo de n nodos, entonces el rango de E es $n - 1$.

Definición:
Una matriz A es **totalmente unimodular** si cada subdeterminante de es 0, 1 o -1.

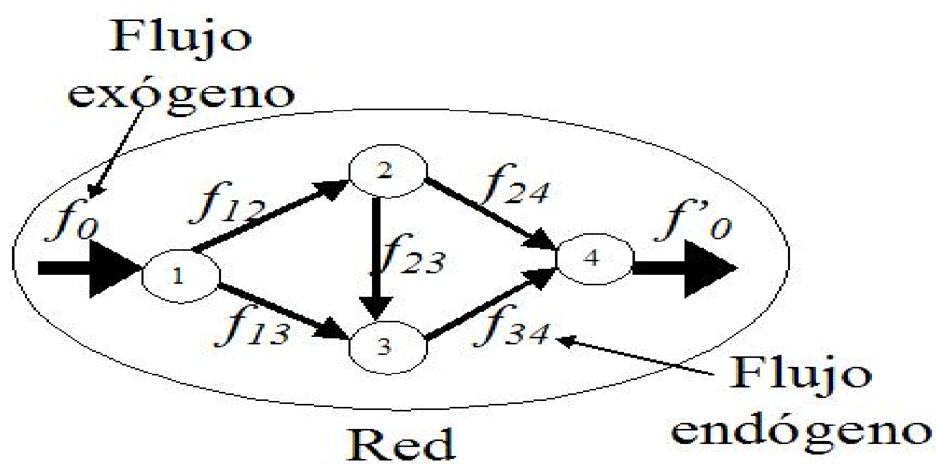
Teorema:
La matriz de incidencia nodo-arco de un grafo dirigido E es totalmente unimodular.

Conceptos Básicos de Flujo en Redes

Definición:
Dado un grafo $G = [N, A]$ conexo se denomina **flujo** a la función que hace corresponder a cada arco (i, j) del grafo un valor real f_{ij}.

Definición:
Una **red** es un sistema formado por un grafo $G = [N, A]$ y una función de flujos.

Definición:
Un **flujo exógeno** corresponde a un flujo que proviene o términa en el exterior del grafo.



En adelante se asume que G es un grafo conexo, salvo que se indique lo contrario.

Restricciones del Problema de Flujo en Redes

1. **Flujos Limitados por Capacidad:**
   Sea u_{ij} la cantidad máxima de flujo que puede transportar el arco (i, j). Entonces se tiene que:
   $$f_{ij} \leq u_{ij}$$
   Si además el flujo no puede ser negativo se agrega la restricción:
   $$f_{ij} \geq 0$$
   Por otra parte, si el también hay requerimientos de flujo se agrega la siguiente restricción:
   $$f_{ij} \geq l_{ij}$$
   Con l_{ij} el valor del flujo minimo entre los nodos i y j.

2. **Conservación de Flujo en los Nodos:**
   Lo que entra a un nodo debe ser igual a lo que sale
   $$\sum_{j:(i,j)\in A} f_{ij} - \sum_{k:(k,i)\in A} f_{ki} = 0$$

Restricciones del Problema de Flujo en Redes

En esta ecuación se utiliza la convención de que los flujos salientes de un nodo se suman y los que entran se restan.

Si existiera un flujo exógeno, la convención se mantiene, pero como los flujos exógenos son generalmente conocidos al pasarlos al lado derecho de la igualdad cambian de signo.

La conservación de flujo cuando existe flujo exógeno sería:

$$\sum_{j:(i,j)\in A} f_{ij} - \sum_{k:(k,i)\in A} f_{ki} = \pm q_i \quad \forall i \in N$$

Donde $q_i = 0$ si no hay flujo exógeno asociado al nodo, q_i con signo $+$ si el flujo exógeno entra al nodo y q_i con signo $-$ si el flujo exógeno sale del nodo.

Restricciones del Problema de Flujo en Redes

Definición:
Un flujo f que satisface las condiciones relativas a las capacidades inferiores y superiores y las condiciones de conservación en los nodos se dice que es un **flujo factible**.

La notación matricial de la restricción de conservación de flujo es:

$$Ef = q$$

Con E la matriz de incidencia nodo-arco, f el vector de flujos entre los nodos ordenado de acuerdo a las columnas de E y q el vector de flujos exógenos.

Problema del Flujo Máximo

Este problema consiste en llevar la máxima cantidad de flujo a través de una red dada $G = [N, A]$.

Se tiene en cada arco (i, j) una capacidad máxima u_{ij} (y eventualmente una mínima l_{ij}) y dos nodos con flujo exógeno, uno donde entra y otro por donde sale.

Generalmente se identifica al nodo origen (donde entra el flujo) como el nodo 1 y el nodo destino (donde sale el flujo) como el nodo n y el flujo exógeno se denota por F.

Modelamiento del Problema:

$$
\begin{array}{ll}
 & \max F \\
\text{s.a.} & \sum_{j:(1,j)\in A} f_{1j} - \sum_{k:(k,1)\in A} f_{k1} - F = 0 \\
 & \sum_{j:(i,j)\in A} f_{1j} - \sum_{k:(k,i)\in A} f_{k1} = 0 \qquad i = 2, \ldots, n-1 \\
 & \sum_{j:(n,j)\in A} f_{nj} - \sum_{k:(k,n)\in A} f_{kn} + F = 0 \\
 & f_{ij} \le u_{ij} \qquad \forall (i,j) \in A \\
 & f_{ij} \ge l_{ij} \qquad \forall (i,j) \in A
\end{array}
$$

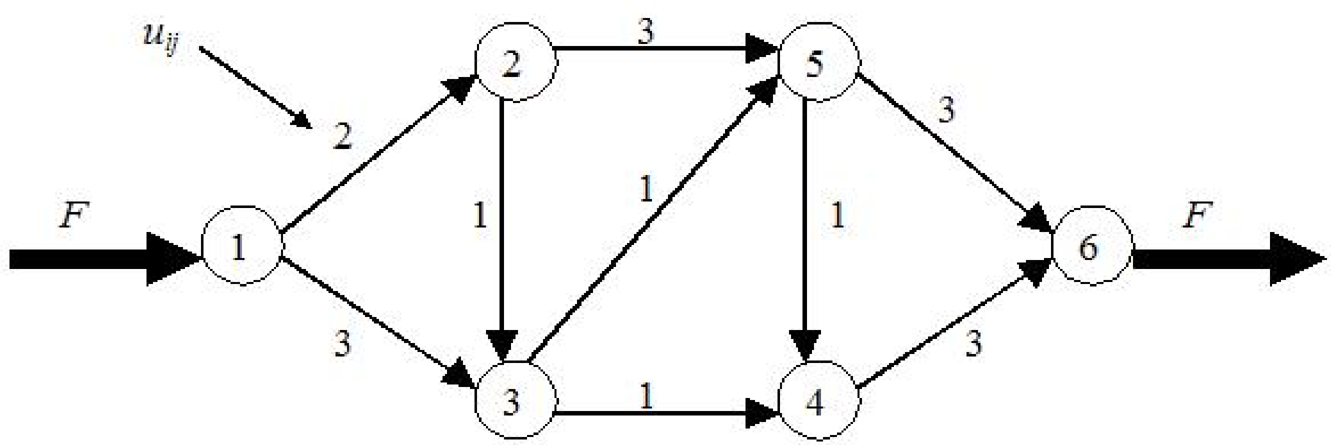
Corte mínimo = Flujo máximo

Dado un conjunto de nodos S, donde el nodo 1 pertenece a S y el nodo n no pertenece a S, se define la capacidad del corte $c(S)$ como la suma de las capacidades de los arcos que van de S a $\overline{S}$ (donde $\overline{S}$ son los nodos de la red que no están en S).

Propiedad importante:

En cualquier red se verifica que el flujo máximo es igual a la capacidad del corte de capacidad mínima.

Ejemplo:



G

Ford y Fulkerson

Para resolver el problema anteriormente planteado se usa el algoritmo Ford y Fulkerson:

Supongamos que se tiene un flujo factible en el grafo $G = [N, A]$. Si todas las cotas inferiores son cero, entonces el flujo factible inicial es $F = 0$, con $f_{ij} = 0 \qquad \forall (i, j)$.

Definición:
Dado un flujo factible f, un camino C desde el origen al nodo destino es un **camino de aumento** con respecto a f si:

$$f_{ij} < u_{ij} \qquad \forall \text{ arco } (i, j) \text{ hacia adelante en } C, \text{ o}$$

$$f_{ij} > l_{ij} \qquad \forall \text{ arco } (i, j) \text{ hacia atrás en } C$$

Observación: C en realidad es una cadena en G.

Luego, dado un flujo factible f en G, con flujo exógeno F, si no existe camino de aumento de flujo con respecto a f, entre el nodo origen y el destino, entonces F es máximo.

Ford y Fulkerson

La idea del algoritmo es construir iterativamente el flujo máximo, a partir de un flujo factible, determinando en cada iteración un camino de aumento de flujo con respecto al flujo existente. De esta forma, se puede incrementar el flujo existente en una cantidad ε determinada por las capacidades $residuales$ de los arcos pertenecientes al camino. Esto se repite hasta que no exista ningún camino de aumento de flujo entre el nodo de origen y el destino.

Cada iteración del algoritmo consta de 2 partes:

- Búsqueda de un camino de aumento de flujo con respecto al flujo existente.
- Determinación de la cantidad de flujo adicional máxima que puede transportar el camino determinado en la primera parte.

Para determinar un camino de aumento de flujo se utiliza un grafo auxiliar G', denominado grafo incremental o residual, donde los nodos son los mismos de G, pero los arcos se determinan de la siguiente forma:

Ford y Fulkerson

- Si $f_{ij} < u_{ij}$ en G, entonces el arco (i, j) se incorpora a G' con capacidad residual o incremental $\Delta_{ij} = u_{ij} - f_{ij}$.
- Si $f_{ij} > l_{ij}$ en G, entonces el arco (j, i) se incorpora a G' con capacidad residual o incremental $\Delta_{ij} = f_{ij} - l_{ij}$.

Observación: Si $l_{ij} < f_{ij} < u_{ij}$ en G, se generan 2 arcos en G'.

Una vez construido el grafo auxiliar se procede a buscar un camino entre el nodo inicial y el final en G'. Si este camino no existe nos encontramos en el óptimo y finaliza el algoritmo. En caso contrario se determina la cantidad máxima ε de flujo adicional que el camino encontrado admite, para lo cual se hace lo siguiente:

Sea C' el camino encontrado en G' desde el nodo inicial al final, sea C la cadena correspondiente a C' en G y sea B_1 el conjunto de arcos hacia adelante en C y B_2 el conjunto de arcos en reversa de C. Entonces:

$$\varepsilon = \min\{ \min_{(i,j)\in C\cap B_1} (u_{ij} - f_{ij}), \min_{(i,j)\in C\cap B_2} (f_{ij} - l_{ij})\}$$

Ford y Fulkerson

Luego, se determina un nuevo flujo factible en G de la siguiente forma:

- Arcos hacia adelante en C: se suma ε al flujo existente en G.
- Arcos hacia atrás en C: se resta ε al flujo existente en G.
- Flujo exógeno: $F' = F + \varepsilon$.

Más formalmente, el algoritmo Ford y Fulkerson es el siguiente:

0. Inicialización: determinar un flujo factible f.

1. Construir el grafo auxiliar $G' = [N', A']$, donde $N' = N$ y definir $B_1 = \phi$, $B_2 = \phi$. Para determinar A' se examinan todos los arcos (i, j) de G: si $f_{ij} < u_{ij}$, entonces el arco (i, j) se incorpora a A' y $B_1 = B_1 \cup \{(i, j)\}$ y $\Delta_{ij} = u_{ij} - f_{ij}$, si $f_{ij} > l_{ij}$, entonces el arco (j, i) se incorpora a A' y $B_2 = B_2 \cup \{(j, i)\}$ y $\Delta_{ij} = f_{ij} - l_{ij}$.

Ford y Fulkerson

2. Determinar un camino C' en G' desde el nodo 1 al n.
   Si no existe ningún camino en G', parar. La solución es óptima.
   De lo contrario, determinar ε a partir de la cadena C en G de la siguiente forma:

$$\varepsilon = \min\{ \min_{(i,j)\in C\cap B_1} (u_{ij} - f_{ij}), \min_{(i,j)\in C\cap B_2} (f_{ij} - l_{ij})\}$$

3. Nuevo flujo:

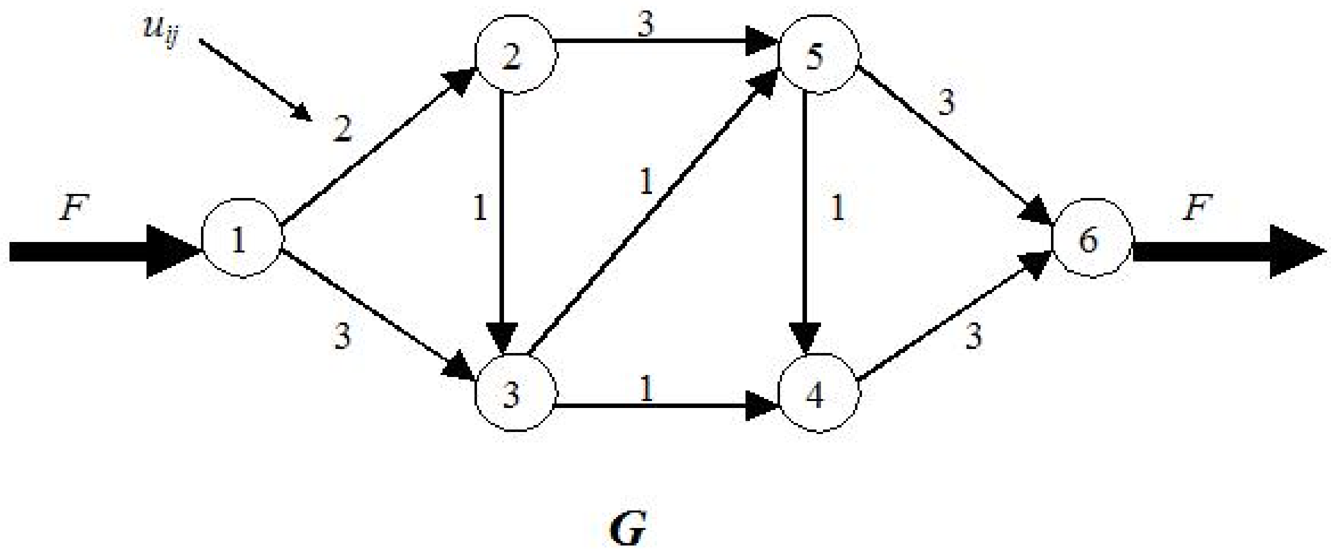
$$\begin{aligned} F &\longleftarrow F + \varepsilon \\ f_{ij} &\longleftarrow f_{ij} + \varepsilon \quad \text{si } (i,j) \in C \cap B_1 \\ f_{ij} &\longleftarrow f_{ij} - \varepsilon \quad \text{si } (i,j) \in C \cap B_2 \end{aligned}$$

   Volver a (1).

Observación: En el paso (2) se pueden construir caminos de muchas formas distintas. Dependiendo de la elección, se obtendrán variantes del algoritmo básico de Ford y Fulkerson.

Ford y Fulkerson

Ej: Sea el siguiente grafo:

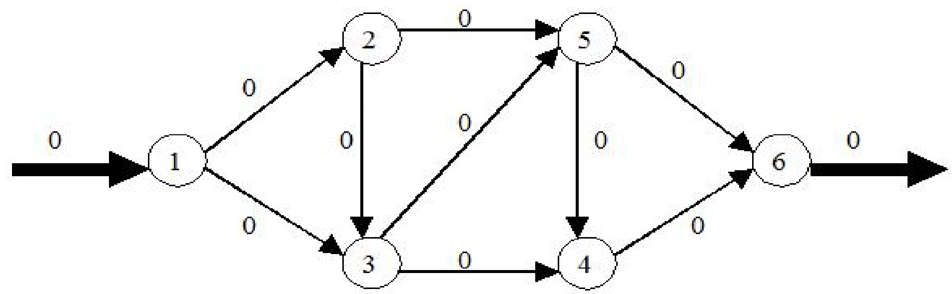


Donde $l_{ij} = 0$, $\forall(i, j)$ y se desea determinar el flujo máximo F que admite la red.

El flujo factible viene dado por $f = 0$, ya que $l_{ij} = 0$. Luego la solución inicial es:

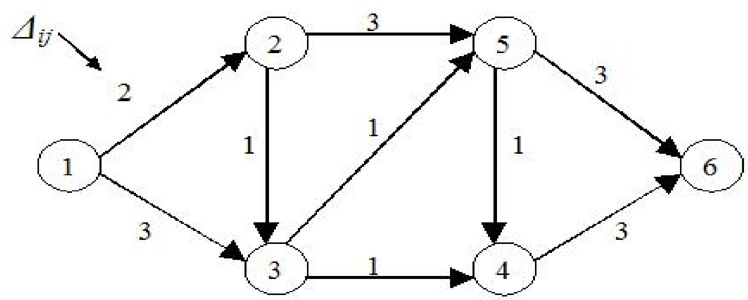
Ford y Fulkerson

Iteración 1:



G

El grafo auxiliar G' sería:



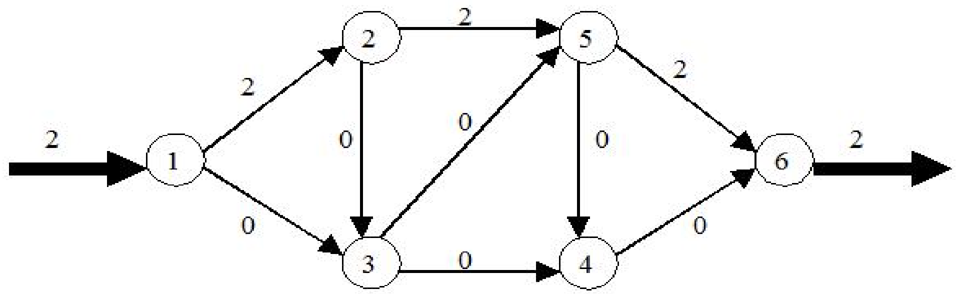
G'

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

Ford y Fulkerson

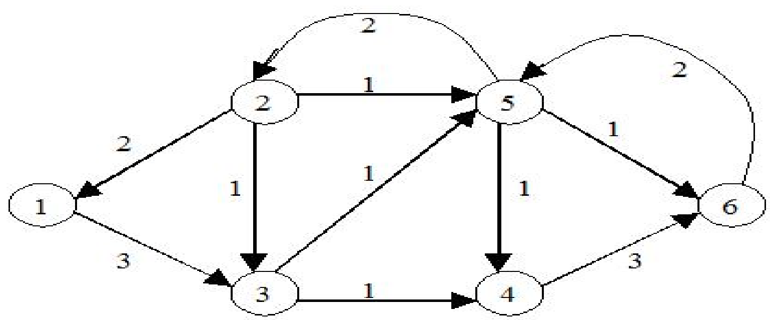
Iteración 2:

El nuevo flujo queda:



G

El grafo auxiliar queda:

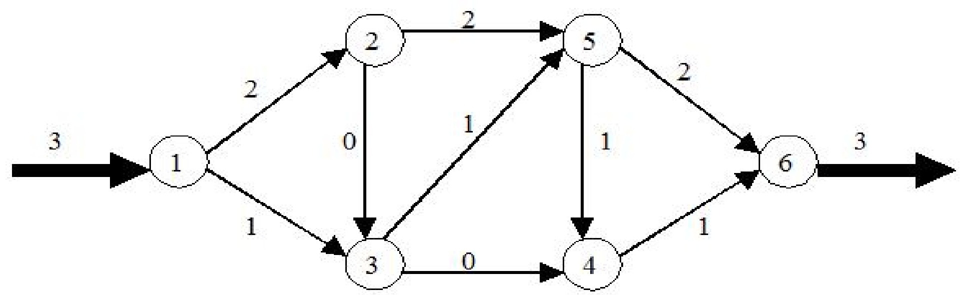


G'

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

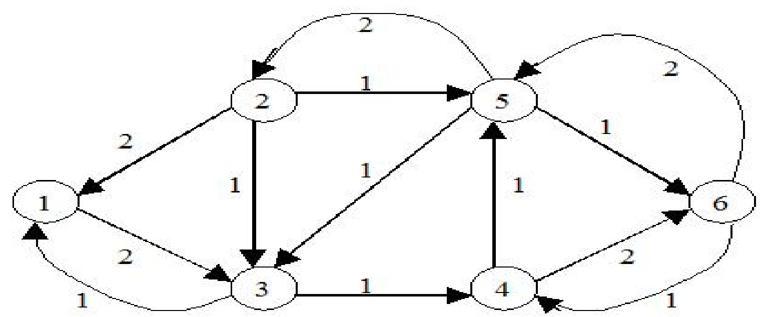
Ford y Fulkerson

Iteración 3:
El nuevo flujo queda:



G

El grafo auxiliar queda:



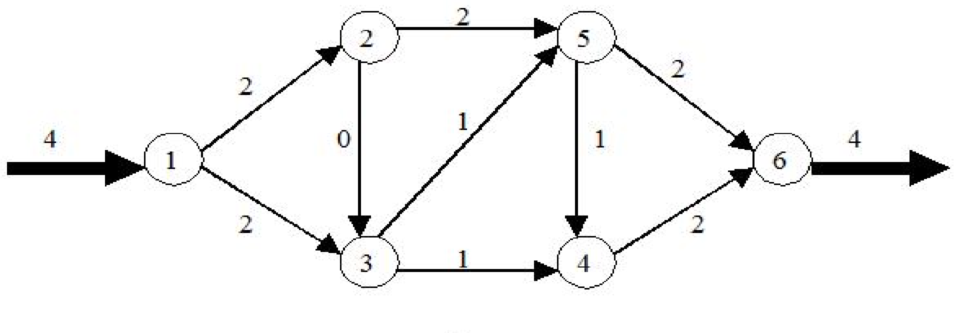
G'

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

Ford y Fulkerson

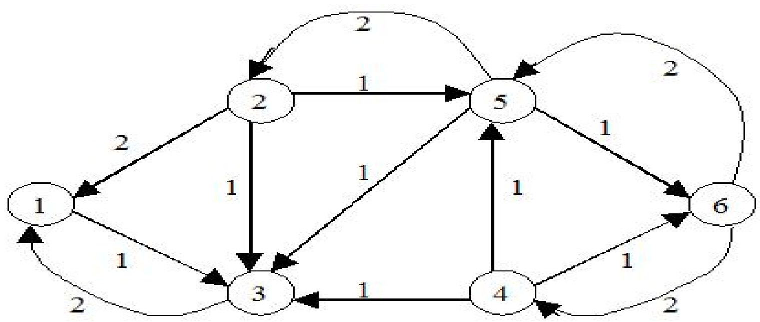
Iteración 4:

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.

Flujo Factible Inicial

Cuando existen arcos con capacidades inferiores no nulas, $F = 0$ no es un flujo factible.

Para establecer un flujo inicial factible se resuelve un problema de flujo máximo derivado del original, cuyo grafo se construye de la siguiente forma:

1. Se agrega un nuevo arco que vaya desde el nodo destino al inicial (arco $(n, 1)$).
2. Se agrega un nodo artificial origen a y un nodo artificial destino b.
3. Por cada nodo $i \in N$ se agregan 2 arcos artificiales: (a, i) y (i, b)

Así queda el grafo $G^* = [N^*, A^*]$, con $N^* = N \cup \{a, b\}$ y $A^* = A \cup (n, 1) \cup \{(a, i), (i, b) \forall i \in N\}$.

Flujo Factible Inicial

Las capacidades inferiores l_{ij} de todos los arcos A^* se definen iguales a 0 y las capacidades superiores se definen de la siguiente manera:

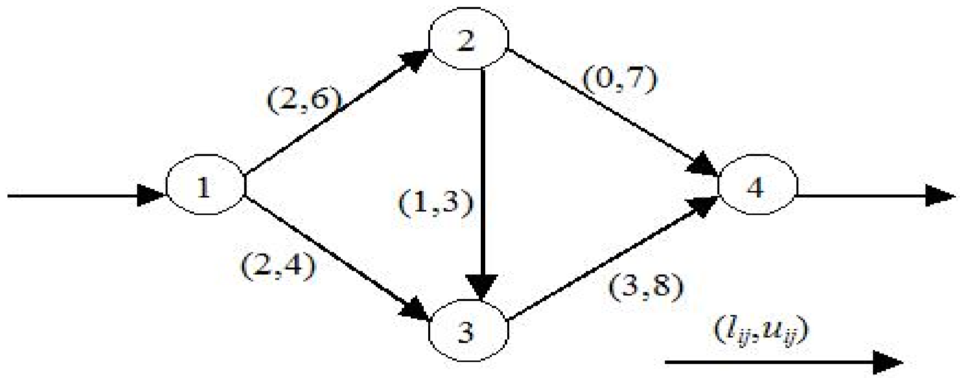
$$\begin{array}{ll} u^*_{ij} = u_{ij} - l_{ij} & \text{para } (i,j) \text{ tal que } (i,j) \in A \\ u^*_{ai} = \sum_{k \in N} l_{ki} & \text{para } (a,i) \text{ tal que } i \in N \\ u^*_{ib} = \sum_{k \in N} l_{ik} & \text{para } (i,b) \text{ tal que } i \in N \\ u^*_{n1} = +\infty & \end{array}$$

Si $\{(i,k), k \in N\} = \phi$ o $\{(k,i), k \in N\} = \phi$, entonces $u^*_{ai} = 0$ o $u^*_{ib} = 0$, respectivamente.

Luego se procede a determinar el flujo máximo F^* en G^* a través de Ford y Fulkerson. Sea el vector f^* la solución óptima del problema de flujo máximo en G^*. Se tiene que si $F^* < \sum_{(i,j) \in A} l_{ij}$ entonces la red original G no admite flujo factible.
En cambio, si $F^* = \sum_{(i,j) \in A} l_{ij}$ entonces el vector de flujos $f'_{ij} = f^*_{ij} + l_{ij}\ \forall (i,j) \in A$, es un vector de flujo factible para la red original.

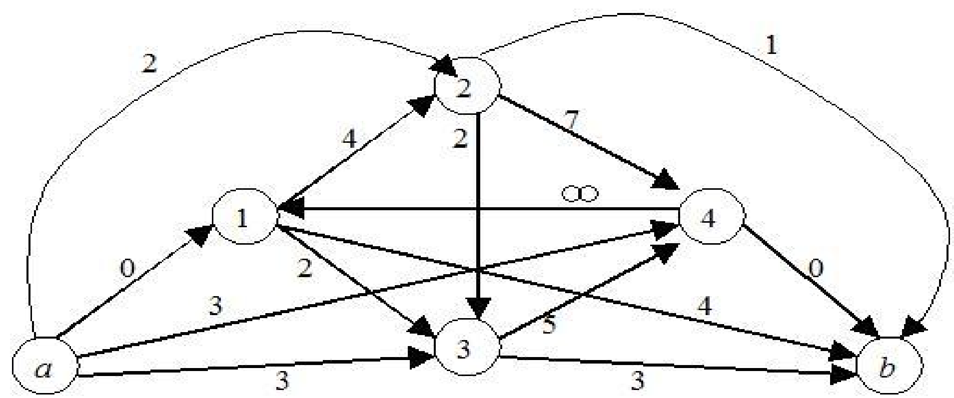
Flujo Factible Inicial

Ej: Sea la siguiente red



Se desea determinar una solución factible inicial.

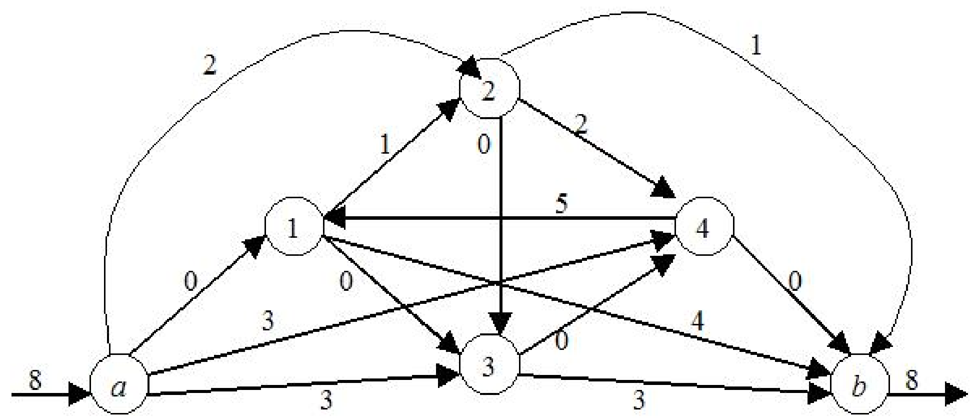
Construyamos el grafo G^*



G^*

Flujo Factible Inicial

La solución óptima de G^* es:



Vemos que $F^* = 8$ y la suma de las capacidades inferiores de todos los arcos de G también es 8, por lo que la red original tiene solución factible inicial igual a:

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

y $F' = f^*_{41} = 5$ es un flujo factible en G.

Problema de la Ruta más Corta

Se tiene una red donde cada arco tiene un costo c_{ij} de recorrerlo (tiempo, distancia, dinero, etc)

El problema consiste en llegar desde un nodo denominado origen (nodo s) a un nodo denominado destino (nodo t) al menor costo posible.

Modelamiento del Problema:

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

$$\begin{array}{lll}
\text{s.a.} & \sum_{j:(s,j)\in A} x_{sj} - \sum_{k:(k,s)\in A} x_{ks} = 1 & \\
& \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = 0 & i \neq s, t \\
& \sum_{j:(t,j)\in A} x_{tj} - \sum_{k:(k,t)\in A} x_{kt} = -1 & \\
& x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A &
\end{array}$$

Donde x_{ij} vale 1 si se recorre el nodo (i,j) y cero si no.

Problema de la Ruta más Corta

Gracias a que la matriz de incidencia de este problema es totalmente unimodular, se puede relajar la condición de binariedad de las variables (porque los vértices del poliedro factible son enteros), resultando el siguiente modelo lineal continuo:

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

$$\begin{array}{lll}
\text{s.a.} & \sum_{j:(s,j)\in A} x_{sj} - \sum_{k:(k,s)\in A} x_{ks} = 1 & \\
& \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = 0 & i \neq s,t \\
& \sum_{j:(t,j)\in A} x_{tj} - \sum_{k:(k,t)\in A} x_{kt} = -1 & \\
& 0 \leq x_{ij} \leq 1 \qquad \forall (i,j) \in A &
\end{array}$$

Este problema es un flujo de costo mínimo donde el flujo exógeno tanto en el nodo origen como en el destino es 1 y la capacidad en cada arco es 1 también.

Observación: Si el problema se modela en una red cuyo grafo es no dirigido, cada arco orientado puede sustituirse por dos arcos dirigidos de sentido opuesto.

Algoritmo Dijkstra

Un algoritmo existente para resolver el problema de la ruta más corta es el algoritmo Dijkstra.

El algoritmo necesita un grafo orientado y el valor de los costos positivo para entregar un árbol de caminos mínimos de modo de ir el nodo inicial a cualquier otro nodo.

Algoritmo Dijkstra

Algoritmo:
Sea el grafo orientado $G = [N, A]$ y la función de costos $l : A \longrightarrow R^+$

1. Se define el conjunto $S = \emptyset$, junto con $\pi(1) = 0$, $\pi(i) = +\infty\ \forall i \neq 1$ y $P(1) = 1$.

2. Repetir hasta que $S = N$:

   Encontrar j no perteneciente a S tal que

$$\pi(j) = \min_{k \in T}\{\pi(k)\}$$

   .

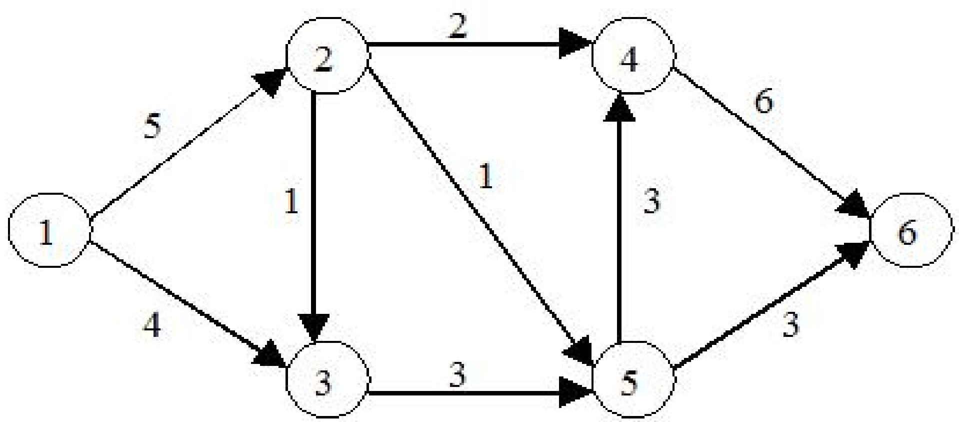
   Luego $S = S \cup \{j\}$.

   Para cada i tal que existe un arco de j a i, con i no perteneciente a S, hacer:
   Si $\pi(i) > \pi(j) + l(j, i)$ entonces $\pi(i) = \pi(j) + l(j, i)$ y $P(i) = j$

Algoritmo Dijkstra

A partir de los $P(i)$ se puede saber que nodo antecede al nodo i en la ruta más corta.

Ej: Se tiene el grafo



Resolvamoslo por Dijkstra:
Inicialización:
$S = \phi$, $\pi(1) = 0$, $\pi(i) = +\infty \; \forall i \neq 1$.

Algoritmo Dijkstra

Iteración 1:

$j = 1 \Rightarrow S = \{1\}$
$\pi(2) = 5,\ P(2) = 1$
$\pi(3) = 4,\ P(3) = 1$

Iteración 2:

$j = 3 \Rightarrow S = \{1, 3\}$
$\pi(5) = 7,\ P(5) = 3$

Iteración 3:

$j = 2 \Rightarrow S = \{1, 3, 2\}$
$\pi(4) = 7,\ P(4) = 2$
$\pi(5) = 6,\ P(5) = 2$

Iteración 4:

$j = 5 \Rightarrow S = \{1, 3, 2, 5\}$
$\pi(6) = 9,\ P(6) = 5$

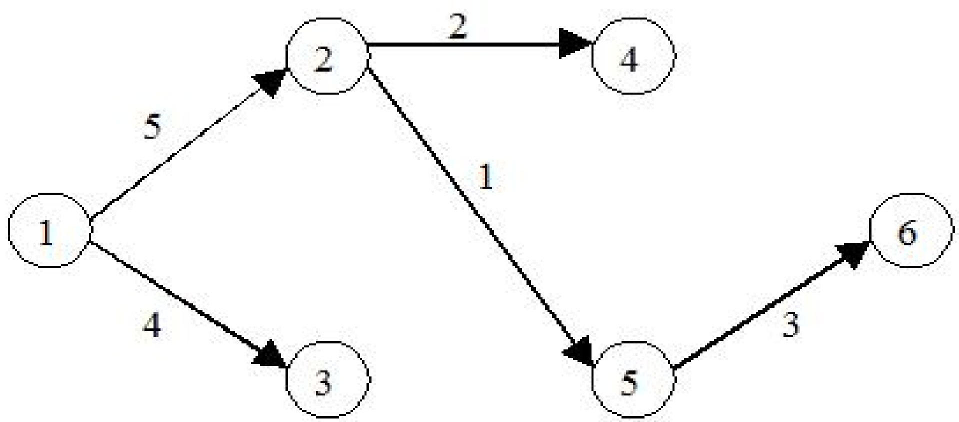
Iteración 5:

$j = 4 \Rightarrow S = \{1, 3, 2, 5, 4\}$

Algoritmo Dijkstra

Iteración 6:
$j = 5 \Rightarrow S = \{1, 3, 2, 5, 4, 6\}$

Así el árbol de la solución queda:



Observación 1: El orden del algoritmo es $O(n^2)$

Observación 2: Notar que pedimos que todos los costos sean positivos.

Ejercicio:
Por qué el algoritmo no se puede aplicar si hay costos negativos? Buscar un ejemplo donde Dijkstra no funcione con algún costo negativo.