6. Problema del flujo de coste mínimo

El problema de flujo de costo mínimo tiene una posición medular entre los problemas de optimización de redes; abarca una clase amplia de aplicaciones y su solución es muy eficiente. Igual que el problema del flujo máximo, toma en cuenta un flujo en una red con capacidades limitadas en sus arcos. Y, como problema de la ruta más corta, considera un costo (o distancia) para el flujo a través de un arco. Al igual que el problema de transporte o el de asignación, puede manejar varios orígenes (nodos fuente) y varios destinos (nodos demandas o terminales) para el flujo, de nuevo, con costos asociados. De hecho, estos cuatro problemas son casos especiales del problema de flujo de costo mínimo.

6.1. Descripción del problema

Las características que describen el problema del flujo de costo mínimo son:

1. La red es una red dirigida conexa.
2. Al menos uno de los nodos es nodo fuente.
3. Al menos uno de los nodos es nodo demanda.
4. El resto de los nodos son nodos de trasbordo.
5. Se permite el flujo a través de un arco sólo en la dirección indicada por la flecha, donde la cantidad máxima de flujo está dada por la capacidad del arco. (Si el flujo puede ocurrir en ambas direcciones, debe representarse por un par de arcos con direcciones opuestas.)
6. La red tiene suficientes arcos con suficiente capacidad para permitir que los flujos generados por los nodos fuente lleguen a los nodos demanda.
7. El costo del flujo a través del arco es proporcional a la cantidad de ese flujo, es decir, se conoce el costo por unidad.
8. El objetivo es minimizar el costo total de enviar el suministro disponible a través de la red para satisfacer la demanda dada. (Un objetivo alternativo es maximizar la ganancia total del envío).

6.2. Algoritmo de Dijkstra

El algoritmo que se presenta para la resolución de este problema es el de Dijkstra (1959). Hay otros algoritmos que resuelven el mismo problema como los de Bellman (1958), Dantzig (1960), Ford y Fulkerson (1962).

El objetivo es determinar la ruta más económica entre el Nodo Fuente y el Nodo Términal de una red.

Consideramos que los arcos de una red pueden pertenecer a uno sólo de los siguientes conjuntos mutuamente excluyentes, a saber:

a) El arco pertenece a un árbol.

b) El arco no pertenece a un árbol.

Al inicio, los arcos no pertenecen al árbol. En cada Paso o Iteración, el algoritmo incrementa en uno los arcos del árbol, hasta llegar a $n - 1$ arcos, donde n es el número de arcos de la red.

Cuando el árbol tiene $n-1$ arcos el proceso se para y queda determinada la solución del problema.

Pasos a seguir:

Paso 1 Sea F el Nodo Fuente. El Nodo F pasa a ser un elemento del árbol. Definimos $C'_{Fk} = c_{Fk}$ para todo arco (F, k) que esté definido en la red. Se inicializa $C_{FF} = 0$. El significado de C'_{Fk} es el coste temporal de ir del Nodo Fuente al Nodo k y el de C_{Fk} el coste permanente de ir del Nodo F al Nodo k. Si el coste no está definido se toma éste como ∞.

Paso 2 Sea

$$C_{Fr} = \min_k \{C'_{Fk}\} = \min_k \{C_{Fj} + c_{jk}\}$$

donde k representa el índice de los nodos para los cuales existe un arco que los une con el árbol. Llamaremos r al índice del Nodo en el que se alcanza dicho mínimo.

Paso 3 El arco (j, r) pasa a ser un elemento del árbol. Se etiqueta al Nodo r con $[C_{Fr}, N_j]$ donde el segundo valor nos indica de donde procede el coste.

Paso 4 Si el árbol tiene $n-1$ arcos el proceso se para, la solución óptima ha sido encontrada. E caso contrario continuamos al paso siguiente.

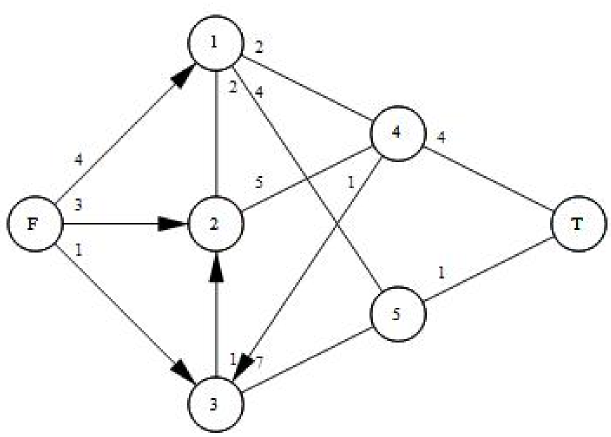
Paso 5 Sea

$$C'_{Fk} = \min_k \{C'_{Fk}, C_{Fr} + c_{rk}\}$$

donde k representa el índice de los nodos para los cuales existe un arco que los une con el árbol. Regrese al Paso 2.

6.2.1. Ejemplo introductorio

Dado el grafo



Hallar la ruta más económica desde el Nodo F al Nodo T.

▶ **Iteración 1**.

Paso 1. El Nodo F pasa a formar parte del árbol esto es

$$\text{Árbol}= \{\varnothing\}, \text{ además } C_{FF} = 0.$$

Los costes temporales para los arcos dirigidos:
$C'_{F1} = 4$, $C'_{F2} = 3$, $C'_{F3} = 1$, $C'_{32} = 1$, $C'_{43} = 1$, con $C'_{1F} = C'_{2F} = C'_{3F} = C'_{23} = C'_{34} = \infty$

Para los arcos no dirigidos:
$C'_{12} = C'_{21} = 2$, $C'_{15} = C'_{51} = 4$, $C'_{14} = C'_{41} = 2$, $C'_{24} = C'_{42} = 5$, $C'_{35} = C'_{53} = 7$, $C'_{4T} = C'_{T4} = 4$, $C'_{5T} = C'_{T5} = 1$.

El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{1, 2, 3\}$. En este caso los costes temporales son: $C'_{F1} = 4$, $C'_{F2} = 3$, $C'_{F3} = 1$.

La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=1,2,3}\{C'_{Fk}\} = \min_{k=1,2,3}\{C_{FF} + c_{Fk}\} = \min\{4, 3, 1\} = 1$$

deduciendo que $r = 3$ y $C_{F3} = C'_{F3} = 1$.

Paso 3. El arco $(F, 3)$ pasa a formar parte del árbol. Se etiqueta al Nodo 3 con $[1, F]$.

Paso 4. Como Árbol= $\{3\}$ no contiene $n - 1 = 7 - 1 = 6$ elementos, se continua al siguiente paso.

Paso 5. El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{1, 2, 4, 5\}$. Dado que $r = 3$, los nuevos costes temporales se recalculan mediante

$$C'_{Fk} = \min_{k=1,2,4,5}\{C'_{Fk}, C_{F3} + c_{3k}\}$$

en nuestro caso concreto tenemos que

$C'_{F1} = \min\{C'_{F1}, C_{F3} + c_{31}\} = \min\{4, 1 + \infty\} = 4$
$C'_{F2} = \min\{C'_{F2}, C_{F3} + c_{32}\} = \min\{3, 1 + 1\} = 2$
$C'_{F4} = \min\{C'_{F4}, C_{F3} + c_{34}\} = \min\{\infty, 1 + \infty\} = \infty$
$C'_{F5} = \min\{C'_{F5}, C_{F3} + c_{35}\} = \min\{\infty, 1 + 7\} = 8$

La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞

▶ **Iteración 2**.

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=1,2,4,5}\{C'_{Fk}\} = \min_{k=1,2,4,5}\{C_{F3} + c_{3k}\} = \min\{4, 2, \infty, 8\} = 2$$

deduciendo que $r = 2$ y $C_{F2} = C'_{F2} = 2$.

Paso 3. El arco $(3, 2)$ (i.e. r antigua es 3 y la r nueva es 2) pasa a formar parte del árbol. Se etiqueta al Nodo 3 con $[2, 3]$.

Paso 4. Como Árbol= $\{2, 3\}$ no contiene 6 elementos, se continua al siguiente paso.

Paso 5. El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{1, 4, 5\}$. Dado que $r = 2$, los nuevos costes temporales se recalculan mediante

$$C'_{Fk} = \min_{k=1,4,5}\{C'_{Fk}, C_{F2} + c_{2k}\}$$

en nuestro caso concreto tenemos que

$C'_{F1} = \min\{C'_{F1}, C_{F2} + c_{21}\} = \min\{4, 2 + 2\} = 4$
$C'_{F4} = \min\{C'_{F4}, C_{F2} + c_{24}\} = \min\{\infty, 2 + 5\} = 7$
$C'_{F5} = \min\{C'_{F5}, C_{F2} + c_{25}\} = \min\{8, 2 + \infty\} = 8$

La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞
2	4	2	1	7	8	∞

▶ **Iteración 3**.

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=1,4,5}\{C'_{Fk}\} = \min_{k=1,4,5}\{C_{F2} + c_{2k}\} = \min\{4, 7, 8\} = 4$$

deduciendo que $r = 1$ y $C_{F1} = C'_{F1} = 4$.

A este resultado podríamos llegar mediante la cadena $F \to 3 \to 2 \to 1$ arbitrariamente se escoge el nuevo elemento.

Paso 3. El arco $[F, 1]$ pasa a formar parte del árbol. Se etiqueta al Nodo 1 con $[4, F]$.

Paso 4. Como Árbol= $\{1, 2, 3\}$ no contiene 6 elementos, se continua al siguiente paso.

Paso 5. El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{4, 5\}$. Dado que $r = 1$, los nuevos costes temporales se recalculan mediante

$$C'_{Fk} = \min_{k=4,5}\{C'_{Fk}, C_{F1} + c_{1k}\}$$

en nuestro caso concreto tenemos que
$C'_{F4} = \text{mín}\{C'_{F4}, C_{F1} + c_{14}\} = \text{mín}\{7, 4+2\} = 6$
$C'_{F5} = \text{mín}\{C'_{F5}, C_{F1} + c_{15}\} = \text{mín}\{8, 4+4\} = 8$
La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞
2	4	2	1	7	8	∞
3	4	2	1	6	8	∞

▶ **Iteración 4**.

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=4,5}\{C'_{Fk}\} = \min_{k=4,5}\{C_{F2} + c_{2k}\} = \text{mín}\{6, 8\} = 6$$

deduciendo que $r = 4$ y $C_{Fr} = C'_{F4} = 6$.

Paso 3. El arco $(1,4)$ pasa a formar parte del árbol. Se etiqueta al Nodo 4 con $[6,1]$.

Paso 4. Como Árbol= $\{1,2,3,4\}$ no contiene 6 elementos, se continua al siguiente paso.

Paso 5. El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{5, T\}$. Dado que $r = 4$, los nuevos costes temporales se calculan mediante

$$C'_{Fk} = \min_{k=5,T}\{C'_{Fk}, C_{F4} + c_{4k}\}$$

en nuestro caso concreto tenemos que
$C'_{F5} = \text{mín}\{C'_{F5}, C_{F4} + c_{45}\} = \text{mín}\{8, 6+\infty\} = 8$
$C'_{FT} = \text{mín}\{C'_{FT}, C_{F4} + c_{4T}\} = \text{mín}\{\infty, 6+4\} = 10$
La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞
2	4	2	1	7	8	∞
3	4	2	1	6	8	∞
4	4	2	1	6	8	10

▶ **Iteración 5**.

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=5,T}\{C'_{Fk}\} = \min_{k=5,T}\{C_{F4} + c_{4k}\} = \text{mín}\{8, 10\} = 8$$

deduciendo que $r = 5$ y $C_{F5} = C'_{F5} = 8$.

Paso 3. El arco $(3, 5)$ pasa a formar parte del árbol. Se etiqueta al Nodo 5 con $[8, 3]$.

Aquí también hay empate entre cadenas y elegimos arbitrariamente.

Paso 4. Como Árbol= $\{1, 2, 3, 4, 5\}$ no contiene 6 elementos, se continua al siguiente paso.

Paso 5. El conjunto de Nodos para los cuales existen arcos desde el Árbol es $\{T\}$. Recordamos que los costes temporales existentes son

$$C_{F5} = 8.$$

En este caso y dado que $r = 5$, los nuevos costes temporales se calculan mediante

$$C'_{Fk} = \min_{k=T}\{C'_{Fk}, C_{F5} + c_{5k}\}$$

en nuestro caso concreto tenemos que

$C'_{FT} = \min\{C'_{FT}, C_{F5} + c_{5T}\} = \min\{\infty, 8 + 1\} = 9$

La tabla de costes temporales queda como sigue:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞
2	4	2	1	7	8	∞
3	4	2	1	6	8	∞
4	4	2	1	6	8	10
5	4	2	1	6	8	9

▶ **Iteración 6**.

Paso 2. Calculamos el coste mínimo

$$C_{Fr} = \min_{k=T}\{C'_{Fk}\} = \min_{k=T}\{C_{F5} + c_{5k}\} = \min\{10, 8 + 1\} = 9$$

deduciendo que $r = T$ y $C_{FT} = C'_{FT} = 9$.

Paso 3. El arco $(5, T)$ pasa a formar parte del árbol. Se etiqueta al Nodo T con $[9, 5]$.

Paso 4. Como Árbol= $\{1, 2, 3, 4, 5, T\}$ contiene 6 elementos, se acaba el cálculo. Y se da la solución.

Para ello tendremos en cuenta:

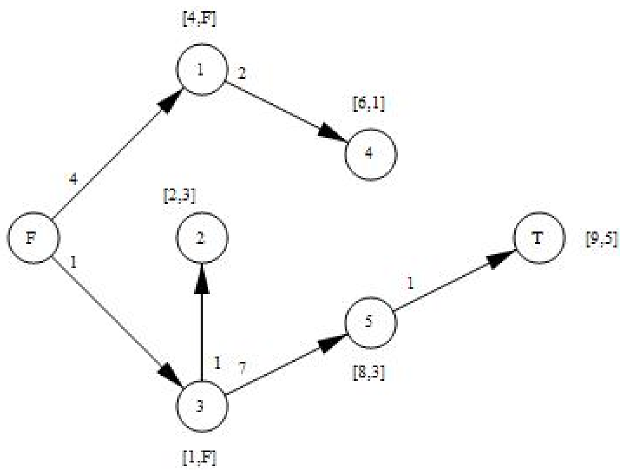
a) La tabla final de costes:

iter/cp	C'_{F1}	C'_{F2}	C'_{F3}	C'_{F4}	C'_{F5}	C'_{FT}
0	4	3	1	∞	∞	∞
1	4	2	1	∞	8	∞
2	4	2	1	7	8	∞
3	4	2	1	6	8	∞
4	4	2	1	6	8	10
5	4	2	1	6	8	9
6	4	2	1	6	8	9

b) La tabla de etiquetas:

Nodo	1	2	3	4	5	T
Etiqueta	$[4, F]$	$[2, 3]$	$[1, F]$	$[6, 1]$	$[8, 3]$	$[9, 5]$

La solución nos dice que el camino nos cuesta 9 unidades por unidad de flujo de la cadena y la cadena es $F \to 3 \to 5 \to T$. Recordar que ésta cadena no es la única que existe. En un gráfico



6.3. Máximo flujo a mínimo coste

El problema que queremos resolver es el de enviar un flujo máximo a través de una red pero a un costo mínimo.

6.3.1. Formulación en forma de programa lineal

Los elementos del problema son:

N el conjunto de nodos. Los nodos pueden ser, nodos demanda, pozos o sumideros, nodos suministradores o fuentes o nodos de transbordo.

El conjuntos de arcos es $A = \{(i, j) : \text{arcos que unen el nodo } i \text{ con el nodo } j \}$.

Por c_{ij} representamos el coste unitario del flujo por el arco (i, j).

Por u_{ij} representamos el flujo máximo que puede pasar por el arco (i, j).

Por b_i representamos el suministro neto (salida - entrada) al nodo i.

La variable se decisión es x_{ij} que representa la cantidad de flujo que pasa a través del arco (i, j).

El problema queda planteado así:

$$
\begin{array}{lll}
\text{mín} & \sum\limits_{(i,j)\in A} c_{ij}x_{ij}, & \\
\text{s.a:} & \sum\limits_{(i,k)\in A} x_{ik} - \sum\limits_{(k,j)\in A} x_{kj} = b_k & \text{para cada } k \in N, \\
 & 0 \leq x_{ij} \leq u_{ij} & \text{para todo } (i,j) \in A.
\end{array}
$$

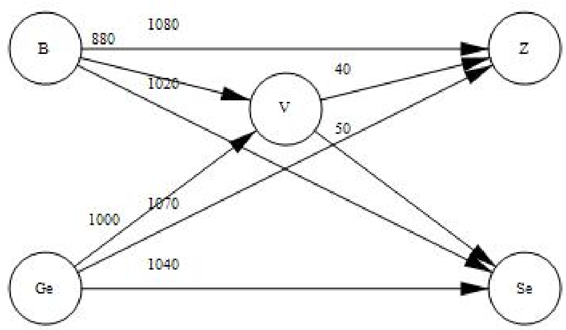
6.3.2. Ejemplo

Cada año Blue Computers produce 400 computadoras en Barcelona y 300 en Gerona. Los clientes de Zaragoza se les deben suministrar 400 computadoras y a los de Sevilla 300. Producir una computadora cuesta 800 euros en Barcelona y 900 euros en Gerona. Las computadoras se suministran por transporte urgente y se podrían enviar por Valencia. Los costes de enviar una computadora vienen mostrados en la tabla

de/a	Valencia	Zaragoza	Sevilla
Barcelona	80	220	280
Gerona	100	140	170
Valencia	-	40	50

Formular un programa lineal que se pueda usar para minimizar el coste total (producción + distribución) de satisfacer la demanda anual de Blue Computers.

El grafo y la tabla de costes es:



de/a	Valencia	Zaragoza	Sevilla
Barcelona	880	1080	1020
Gerona	1000	1070	1040
Valencia	-	40	50

Variables de decisión: x_{BZ}, x_{BV}, x_{BSE}, x_{GEV}, x_{GEZ}, x_{GESE}, x_{VZ} y x_{VSE} que representan el flujo de cada arco

$$
\begin{aligned}
\text{mín} \quad & 1080x_{BZ} + 880x_{BV} + 1020x_{BSE} + 1000x_{GEV} + 1070x_{GEZ} + 1040x_{GESE} + 40x_{VZ} + 50x_{VSE}, \\
\text{s.a:} \quad & x_{BZ} + x_{BV} + x_{BSE} = 400, \\
& x_{GEV} + x_{GEZ} + x_{GESE} = 300, \\
& x_{BV} + x_{GEV} = x_{VZ} + x_{VSE}, \\
& x_{BZ} + x_{VZ} + x_{GEZ} = 300, \\
& x_{BSE} + x_{VSE} + x_{GESE} = 400, \\
& x_{BZ}, x_{BV}, x_{BSE}, x_{GEV}, x_{GEZ}, x_{GESE}, x_{VZ}, x_{VSE} \geq 0.
\end{aligned}
$$

La solución es: $x_{BV} = 400$, $x_{GESE} = 300$, $x_{VZ} = 300$, $x_{VSE} = 100$ y el coste total es 681000.

6.3.3. Algoritmo para grafos

El método heurístico que vamos a usar fue dise nado por Busaker y Gowen, y publicado en el trabajo "A procedure for Determining a Family of Minimal Cost Network Patterns " Technical Report 15. Operations Research Office. The John Hopkins University, Baltimore (1965).

Introducimos la notación (l_{ij}, u_{ij}, c_{ij}) para cada arco (i, j) donde l_{ij} es el límite inferior de flujo, u_{ij} es el límite superior de flujo y c_{ij} el coste unitario del flujo.

Paso 1. $x_{ij} = 0$ para todos los arcos $(i, j) \in A$ y el flujo total que circula por la red es $v = 0$.

Paso 2. Construcción del nuevo coste $\bar{c}_{ij}$ en el arco (i, j) definido por:

(2,1)	$\bar{c}_{ij} = c_{ij}$	si	$0 \leq x_{ij} \leq u_{ij}$	el coste no cambia.
(2,2)	$\bar{c}_{ij} = \infty$	si	$x_{ij} = u_{ij}$	el arco se satura.
(2,3)	$\bar{c}_{ij} = -c_{ij}$	si	$x_{ij} > 0$	arco en sentido inverso.

Los arcos en sentido inverso es una posible reducción del flujo.

Paso 3. Encuentre la ruta más económica del Nodo fuente F al nodo destino T, basada en los costes $\bar{c}_{ij}$ utilizando el algoritmo de Dijkstra o similar.

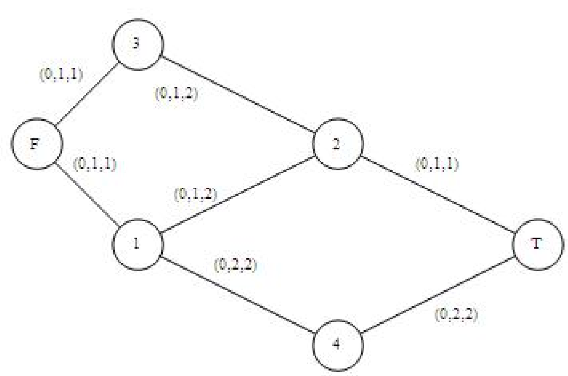
Mándese la mayor cantidad de flujo permisible por la ruta más económica que se haya encontrado (i.e. hasta que uno o varios arcos de la ruta se saturen).

A nádase al flujo actual en toda la red el flujo adicional que se encuentra en este paso.

Si todas las rutas que conducen al Nodo T están saturadas, la solución óptima se ha encontrado, en caso contrario, volvemos al Paso 2.

6.3.4. Ejemplo

Dado el grafo de la figura



Encontrar el máximo flujo al mínimo coste.

Iteración 1.

Paso 1.

$v = 0$ y $x_{ij} = 0$ para todo arco $(i, j) \in A$.

Paso 2. Calculamos los nuevos costes

Costes	$\bar{c}_{F1}$	$\bar{c}_{F3}$	$\bar{c}_{12}$	$\bar{c}_{14}$	$\bar{c}_{32}$	$\bar{c}_{2T}$	$\bar{c}_{4T}$
Valor	1	1	2	2	2	1	2
Condición	(2,1)	(2,1)	(2,1)	(2,1)	(2,1)	(2,1)	(2,1)

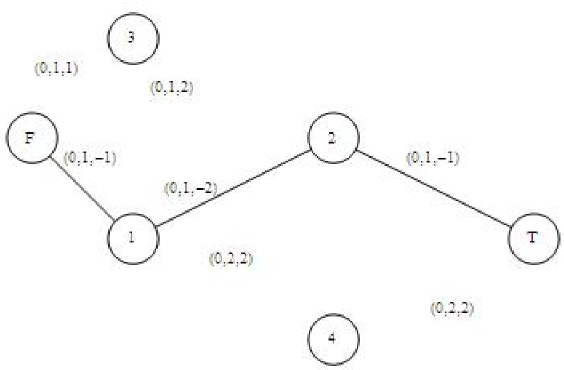
Paso 3. Utilizando Dijkstra encontramos la cadena más ecónomica desde el Nodo F al Nodo T.

(a) $F \to 1 \to 2 \to T$ con coste 4 unidades.

(b) $F \to 3 \to 2 \to T$ con coste 4 unidades.

En ambas cadenas la capacidad máxima de flujo a enviar es de 1 u. Elegimos (a) arbitrariamente.

Gráficamente resulta:



Iteración 2.

Paso 2.

Los nuevos costes son:

Costes	$\bar{c}_{F1}$	$\bar{c}_{12}$	$\bar{c}_{2T}$	$\bar{c}_{1F}$	$\bar{c}_{21}$	$\bar{c}_{T2}$
Valor	∞	∞	∞	-1	-2	-1
Condición	(2,2)	(2,2)	(2,2)	(2,3)	(2,3)	(2,3)

Los restantes permanecen igual.

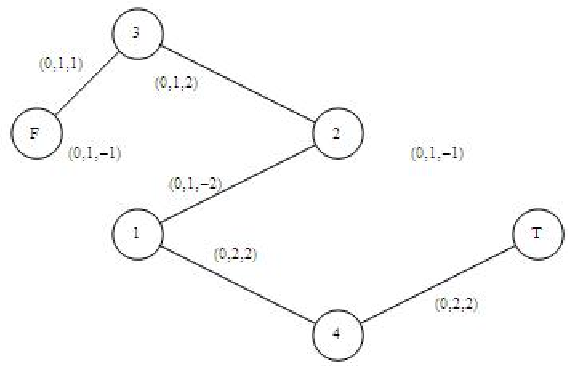
Paso 3. Utilizando Dijkstra encontramos la cadena más ecónomica desde el Nodo F al Nodo T.

$F \to 3 \to 2 \to 1 \to 4 \to T$ con coste 5 unidades.

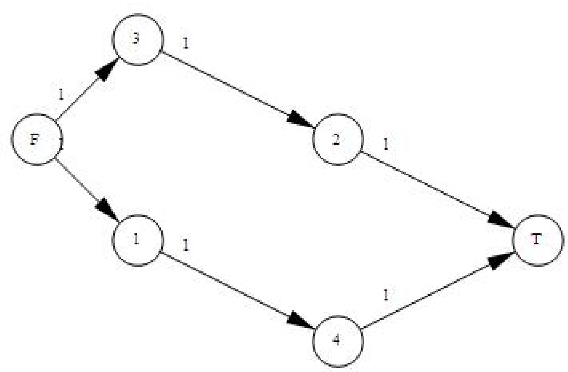
(i.e. $1 + 2 - 2 + 2 + 2 = 5$)

El flujo máximo que se puede enviar es 1 unidad. Notar que el flujo del arco $(1, 2)$ se cancela con el del arco $(2, 1)$.

Gráficamente:



Se pueden enviar 2 unidades a un coste de 9 (ver los costes en el grafo original). Resultado gráfico



Este algoritmo sólo se usa para problemas relativamente peque nos debido al uso de arcos inversos.