Universidad Politécnica
de Madrid

FLUJOS
COSTE MÍNIMO

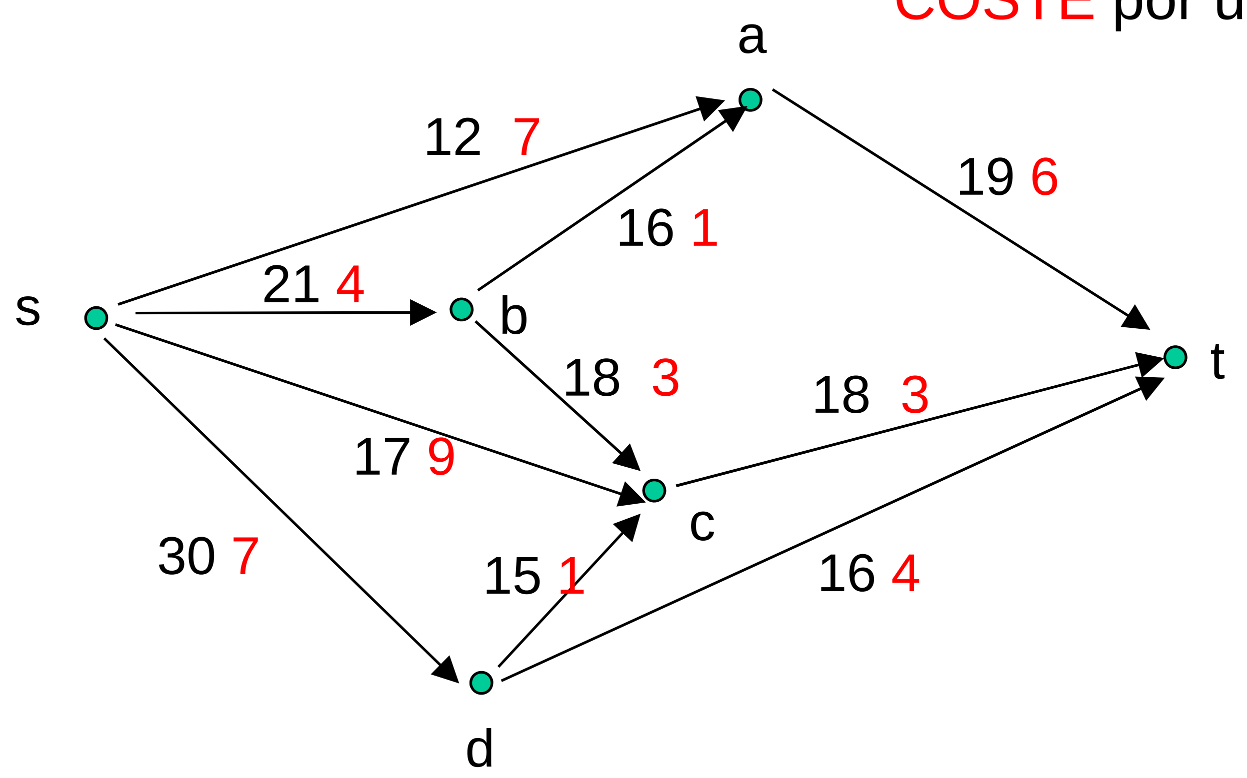
Gregorio Hernández
UPM

Optimización Combinatoria, 2016

FLUJOS EN REDES CON COSTE

En cada arco de la red tenemos: CAPACIDAD limitada

COSTE por unidad de flujo



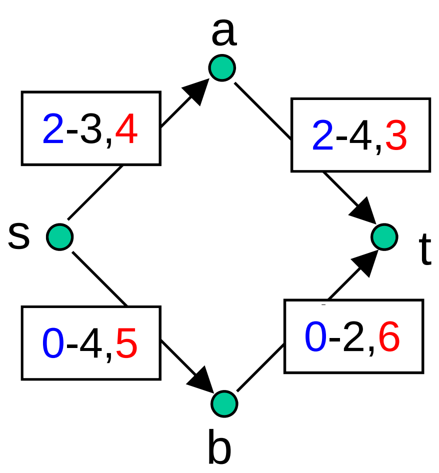
FLUJOS. MÍNIMO COSTE

Red: Digrafo $D = (V, A)$, vértices s, t, capacidad $c: A \to \mathbf{R}^+$
coste $k: A \to \mathbf{R}^+$

El coste de un flujo $f: A \to \mathbf{R}^+$ es $$\mathrm{cost}(f) = \sum_{e \in A} k(e) f(e)$$

OBJETIVO: Hallar un flujo s – t tal que val(f) sea **MÁXIMO**
y que tenga **MÍNIMO** coste entre todos los flujos de valor val(f)

Un st – flujo se dice **óptimo** si cost(f) es mínimo entre todos flujos de s a t de valor val(f)



Notación | 2 - 4, 3 | flujo - capacidad, coste |

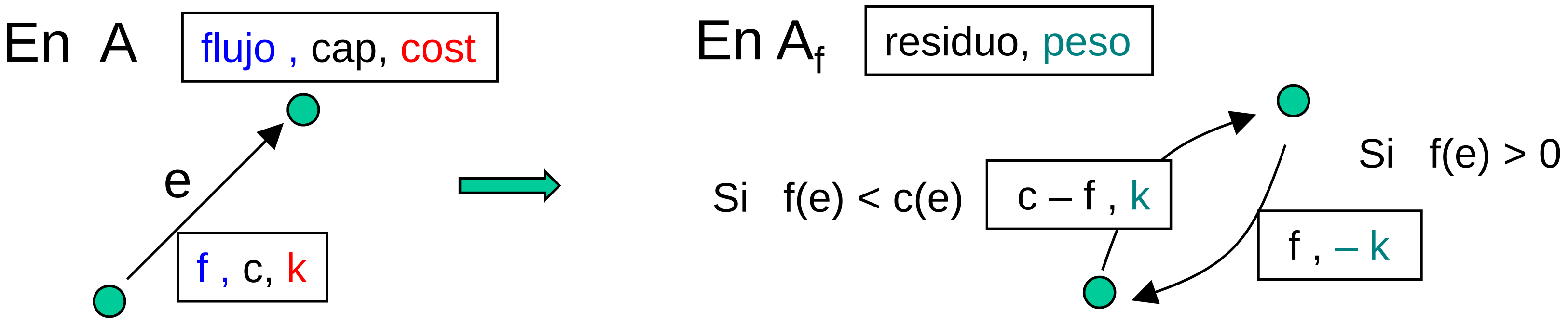
Flujo óptimo de valor 2 y coste mínimo 14

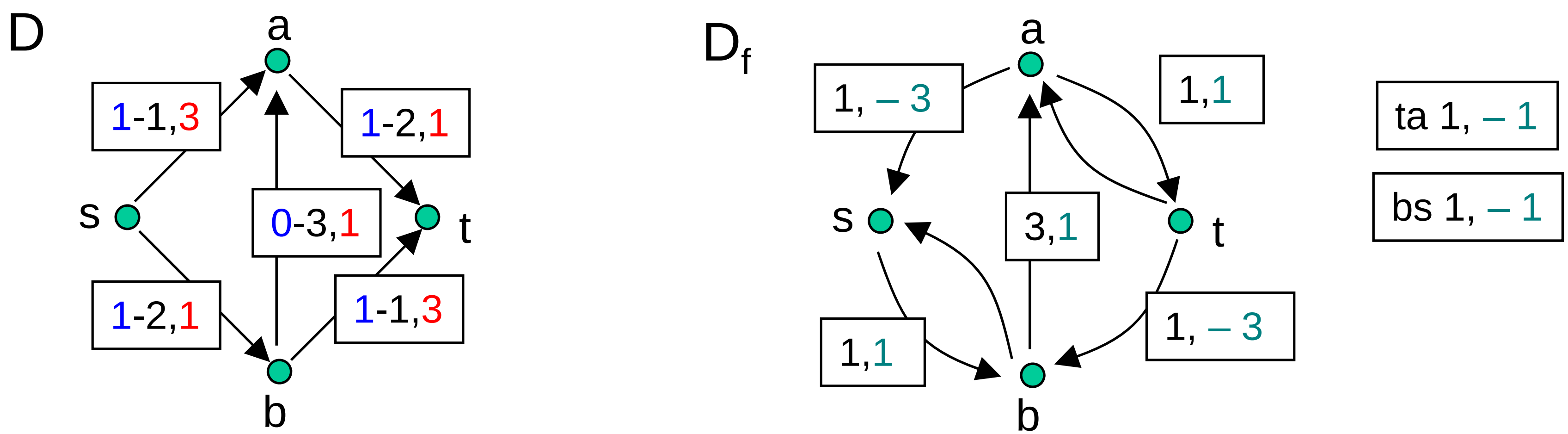
FLUJOS. MÍNIMO COSTE

Red auxiliar (o residual)

Dado f: $A \to \mathbf{R}^+$ flujo en D se construye otra red $D_f = (V, A_f)$

Las aristas en A_f son:





FLUJOS. MÍNIMO COSTE

Lema básico

f es un flujo óptimo $\Leftrightarrow$ D_f no tiene ciclos dirigidos de peso negativo

Dem.:

$\Rightarrow$) Supongamos que D_f tiene un ciclo C de peso negativo, $w(C) < 0$

Cada arco e en A_f tiene residuo positivo, $res(e) > 0$

Llamamos $\alpha = \min\{res(e) \ / \ e \in A_f\}$

Construimos un nuevo flujo g en D modificando el valor de f solo en los arcos relacionados con el ciclo C

$g(a) = f(a) + \alpha$ si $a \in C$

$g(a) = f(a) - \alpha$ si $a^{-1} \in C$

$g(a) = f(a)$ en otro caso

Así g es un flujo s – t en D con valor $val(g) = val(f)$

Pero $cost(g) = cost(f) + \alpha w(C) < cost(f)$

Luego f no es óptimo en contradicción con la hipótesis inicial

FLUJOS. MÍNIMO COSTE

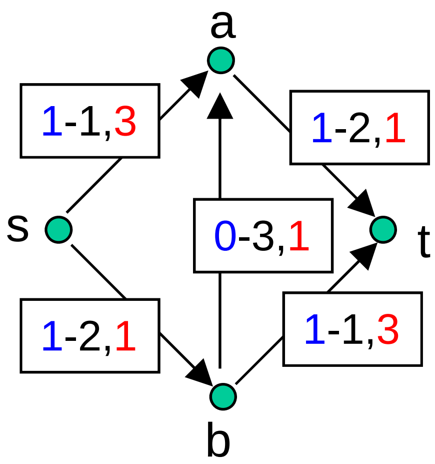
Lema básico

f es un flujo óptimo $\Leftrightarrow$ D_f no tiene ciclos dirigidos de peso negativo

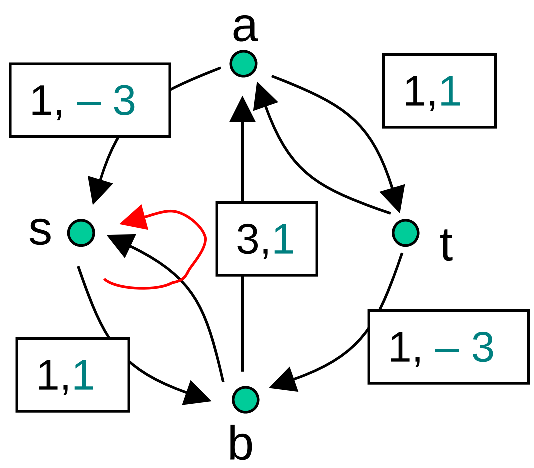
Dem.:

$\Rightarrow$)

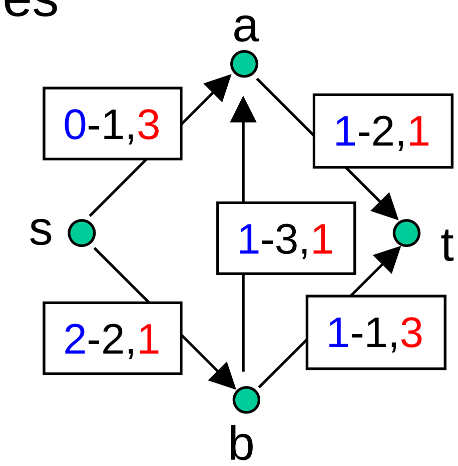
D



D_f



El flujo g en D es



En D_f tenemos el ciclo dirigido C = sbas de residuo 1 y peso negativo w(C) = – 1

FLUJOS. MÍNIMO COSTE

Lema básico

f es un flujo óptimo $\Leftrightarrow$ D_f no tiene ciclos dirigidos de peso negativo

Dem.:

$\Leftarrow$) Sea g un flujo de valor val(f) = val(g)

Comprobaremos que cost(g) > cost(f) y así f será óptimo

Definimos un nuevo flujo h en D_f

$h(e) = g(e) - f(e)$ si $g(e) > f(e)$ en D

$h(e^{-1}) = f(e) - g(e)$ si $g(e) < f(e)$ en D

$h(e) = 0$ para el resto de arcos de A_f

Así h es una circulación en D_f que es no negativa por construcción.

Por tanto, h se puede expresar como $f = \lambda_1 f_1 + \lambda_2 f_2 + \ldots + \lambda_k f_k$
donde $\lambda_i > 0$ y cada f_i es una circulación elemental correspondiente a un ciclo dirigido C_i ($f_i = 1$ en el ciclo y $f_i = 0$ fuera)

Así cost(g) – cost(f) = cost(h) = $\sum_{i=1}^{k} \lambda_i w(C_i) > 0$ (no hay ciclos dirigidos de peso negativo)

Luego f óptimo

FLUJOS. MÍNIMO COSTE

Algoritmo 1 (**PRIMAL**, ciclos negativos)

"A primal method for maximal cost flows", Klein, 1967

PRIMAL significa que en todo momento se tiene una solución factible que se intenta mejorar

Paso 1. Utilizar un algoritmo de máximo flujo para hallar un flujo de valor máximo.

Paso 2. Mientras haya ciclos negativos en el digrafo auxiliar D_f modificar el flujo a lo largo del ciclo negativo, construyendo un nuevo flujo del mismo valor y menor coste, según se ha explicado en la demostración del lema.

Complejidad

El algoritmo NO es polinómico, depende de los valores de las funciones coste y capacidad y del valor del flujo máximo.

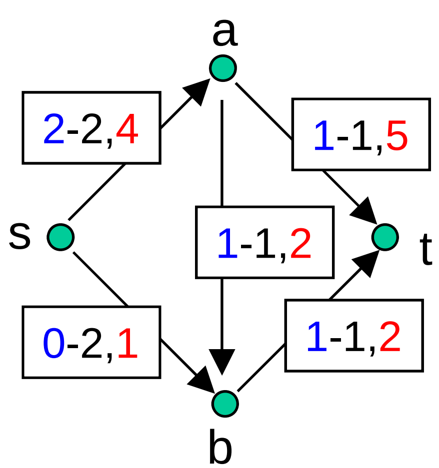
Si se eligen los ciclos negativos adecuadamente, SÍ se consigue complejidad polinómica. El número de iteraciones es a lo más $4nm^2\lceil \ln m \rceil$

"Finding minimum-cost circulations by canceling negative cycles", Goldberg, Tarjan, 1989

MÍNIMO COSTE

Algoritmo 1. Ciclos negativos

D



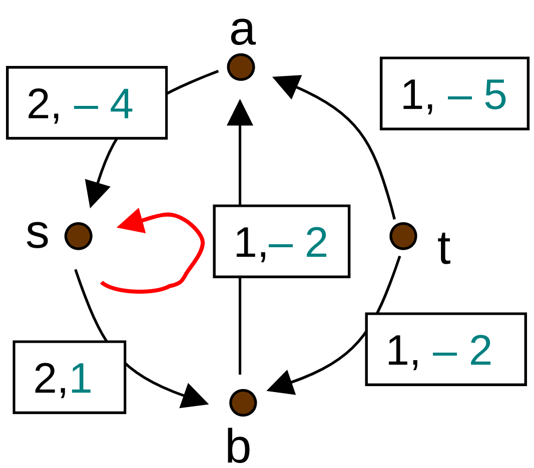
Flujo f de valor máximo 2 y coste 17

Caminos sat flujo 1 coste 9

sabt flujo 1 coste 8

Construimos el digrafo auxiliar D_f

D_f



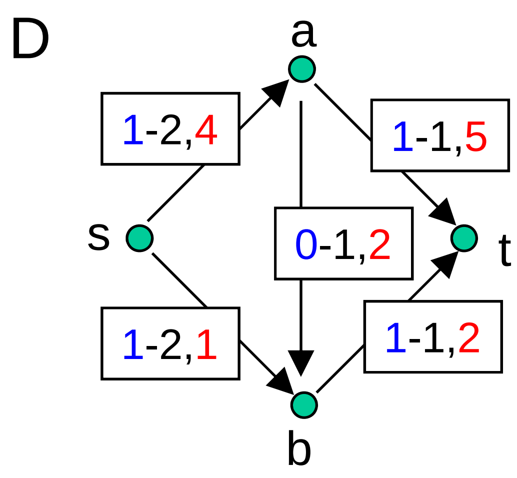
Se detecta el ciclo negativo **C =** sbas de **residuo** 1 y **peso – 5**

Un flujo de 1 unidad a lo largo de C disminuirá el coste en 5

sb aumenta flujo en 1

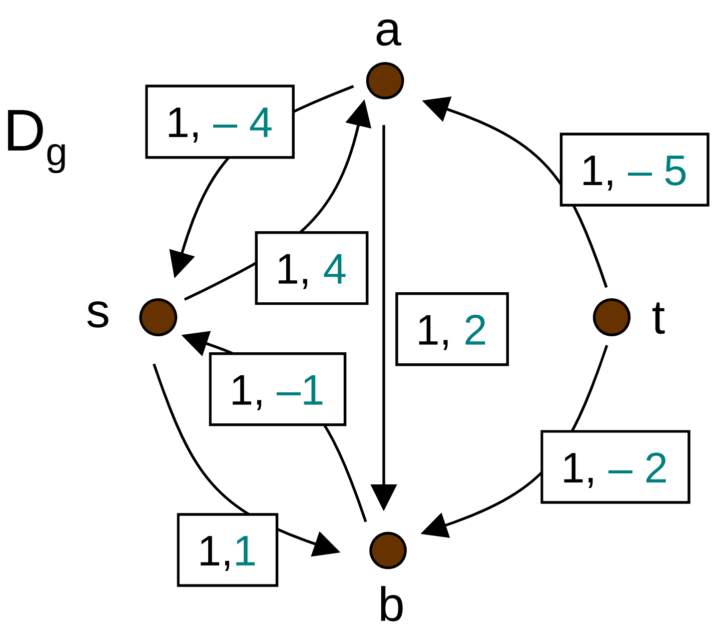
ab y sa disminuyen flujo en 1

MÍNIMO COSTE Algoritmo 1. Ciclos negativos



Flujo g de valor máximo 2 y coste 12

Construimos el digrafo auxiliar D_g



NO hay ciclos negativos en D_g

El flujo g es óptimo y de valor máximo.
Por tanto es el flujo de mínimo coste

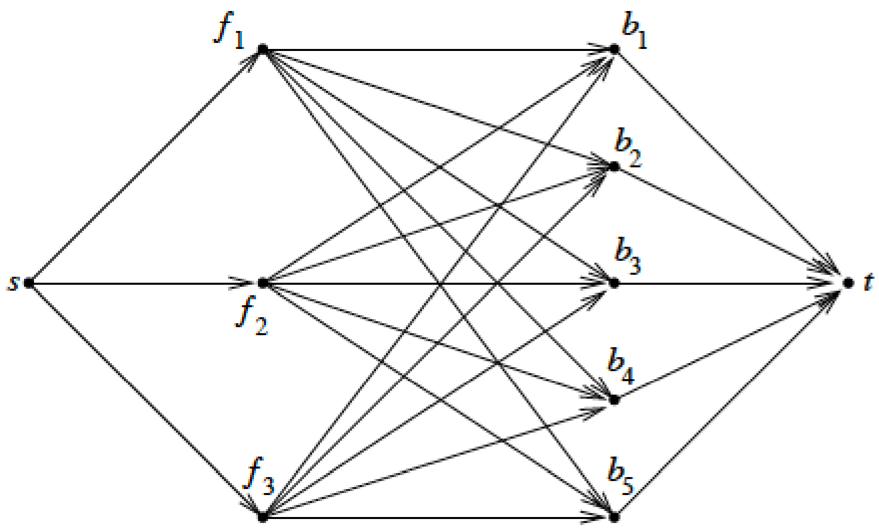
FLUJOS. Problema de transporte

Application 4.2: Transportation problem. Suppose that there are m factories, that all produce the same product, and n customers that use the product. Each month, factory i can produce s_i tons of the product. Customer j needs every month d_j tons of the product. From factory i to customer j we can transport every month at most $c_{i,j}$ tons of the product. The problem is: can the needs of the customers be fulfilled?

In order to solve the problem with the maximum-flow algorithm, we make the graph as in Figure 4.2 (for $m = 3$, $n = 5$). We define a capacity function c on the arcs as follows:

$$\begin{aligned} c(s, f_i) &:= s_i && \text{for } i = 1, \ldots, m, \\ c(f_i, b_j) &:= c_{i,j} && \text{for } i = 1, \ldots, m; j = 1, \ldots, n, \\ c(b_j, t) &:= d_j && \text{for } j = 1, \ldots, n. \end{aligned} \tag{20}$$

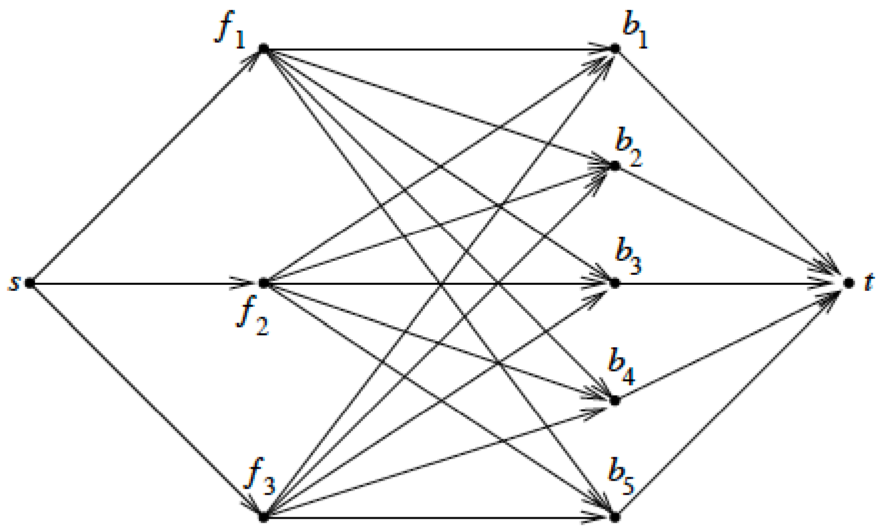
The needs of the customers can be fulfilled $\Leftrightarrow$ there is an s-t flow under c with value $d_1 + d_2 + \ldots + d_n$



FLUJOS con COSTE. Problema de transporte

Application 4.3: Minimum-cost transportation problem. Beside the data in Application 4.2 one may also have a cost function $k_{i,j}$, giving the cost of transporting 1 ton from factory i to costumer j. Moreover, there is given a cost k_i of producing 1 ton by factory i (for each i). We want to make a production and transportation plan that minimizes the total cost.

This problem can be solved by assigning also costs to the arcs in Application 4.2. We can take the costs on the arcs from b_j to t equal to 0.



A. Schrijver, “A Course in Combinatorial Optimization”
http://homepages.cwi.nl/~lex/files/dict.pdf

FLUJOS con COSTE. Aplicaciones

A. Schrijver, “A Course in Combinatorial Optimization”
http://homepages.cwi.nl/~lex/files/dict.pdf

Application 4.4: Routing empty freighters. Historically, in his paper “Optimum utilization of the transportation system”, Koopmans [1948] was one of the first studying the minimum-cost transportation problem, in particular with application to the problem of routing empty freighters. Koopmans considered the surplus and need of register ton of ship capacity at harbours all over the world, as given by the following table (data are aggregated to main harbours):

Net receipt of dry cargo in overseas trade, 1925

Unit: Millions of metric tons per annum

Harbour	Received	Dispatched	Net receipts
New York	23.5	32.7	−9.2
San Francisco	7.2	9.7	−2.5
St. Thomas	10.3	11.5	−1.2
Buenos Aires	7.0	9.6	−2.6
Antofagasta	1.4	4.6	−3.2
Rotterdam	126.4	130.5	− 4.1
Lisbon	37.5	17.0	20.5
Athens	28.3	14.4	13.9
Odessa	0.5	4.7	−4.2
Lagos	2.0	2.4	−0.4
Durban	2.1	4.3	−2.2
Bombay	5.0	8.9	−3.9
Singapore	3.6	6.8	−3.2
Yokohama	9.2	3.0	6.2
Sydney	2.8	6.7	−3.9
Total	266.8	266.8	0.0

FLUJOS con COSTE. Aplicaciones

Application 4.5: Routing of railway stock. NS (Nederlandse Spoorwegen = Dutch Railways) performs a daily schedule on its line Amsterdam–Vlissingen, with the following (weekday) timetable:

ride number		2123	2127	2131	2135	2139	2143	2147	2151	2155	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam	d		6.48	7.55	8.56	9.56	10.56	11.56	12.56	13.56	14.56	15.56	16.56	17.56	18.56	19.56	20.56	21.56	22.56
Rotterdam	a		7.55	8.58	9.58	10.58	11.58	12.58	13.58	14.58	15.58	16.58	17.58	18.58	19.58	20.58	21.58	22.58	23.58
Rotterdam	d	7.00	8.01	9.02	10.03	11.02	12.03	13.02	14.02	15.02	16.00	17.01	18.01	19.02	20.02	21.02	22.02	23.02	
Roosendaal	a	7.40	8.41	9.41	10.43	11.41	12.41	13.41	14.41	15.41	16.43	17.43	18.42	19.41	20.41	21.41	22.41	23.54	
Roosendaal	d	7.43	8.43	9.43	10.45	11.43	12.43	13.43	14.43	15.43	16.45	17.45	18.44	19.43	20.43	21.43			
Vlissingen	a	8.38	9.38	10.38	11.38	12.38	13.38	14.38	15.38	16.38	17.40	18.40	19.39	20.38	21.38	22.38			

ride number		2108	2112	2116	2120	2124	2128	2132	2136	2140	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen	d			5.30	6.54	7.56	8.56	9.56	10.56	11.56	12.56	13.56	14.56	15.56	16.56	17.56	18.56	19.55	
Roosendaal	a			6.35	7.48	8.50	9.50	10.50	11.50	12.50	13.50	14.50	15.50	16.50	17.50	18.50	19.50	20.49	
Roosendaal	d		5.29	6.43	7.52	8.53	9.53	10.53	11.53	12.53	13.53	14.53	15.53	16.53	17.53	18.53	19.53	20.52	21.53
Rotterdam	a		6.28	7.26	8.32	9.32	10.32	11.32	12.32	13.32	14.32	15.32	16.32	17.33	18.32	19.32	20.32	21.30	22.32
Rotterdam	d	5.31	6.29	7.32	8.35	9.34	10.34	11.34	12.34	13.35	14.35	15.34	16.34	17.35	18.34	19.34	20.35	21.32	22.34
Amsterdam	a	6.39	7.38	8.38	9.40	10.38	11.38	12.38	13.38	14.38	15.38	16.40	17.38	18.38	19.38	20.38	21.38	22.38	23.38

The rides are carried out by one type of stock, that consists of two-way units that can be coupled with each other. The length of the trains can be changed at the end stations and at two intermediate stations: Rotterdam and Roosendaal. So in this example, each train ride consists of three ride 'segments'.

FLUJOS con COSTE. Aplicaciones

Application 4.5: Routing of railway stock.

Based on the expected number of passengers, NS determines for each ride segment a minimum number of units that should be deployed for that segment:

ride number	2123	2127	2131	2135	2139	2143	2147	2151	2155	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam-Rotterdam		3	5	4	3	3	3	3	3	3	4	5	5	3	2	2	2	1
Rotterdam-Roosendaal	2	3	4	4	2	3	3	3	3	4	5	5	4	2	2	2	1	
Roosendaal-Vlissingen	3	2	2	2	2	3	2	3	3	3	4	4	3	2	1			

ride number	2108	2112	2116	2120	2124	2128	2132	2136	2140	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen-Roosendaal			2	4	4	4	2	2	3	2	2	2	3	3	2	2	2	
Roosendaal-Rotterdam		2	4	5	4	5	3	3	3	2	3	3	4	3	2	2	2	2
Rotterdam-Amsterdam	1	3	5	4	4	5	3	3	3	3	3	4	5	3	2	2	2	2

A unit uncoupled from a train at a station can be coupled at any other later train, in the same direction or the other. Moreover, for each segment there is a maximum number of units given that can be used for that segment (depending for instance on the length of station platforms).

The company now wishes to find the minimum number of units that should be used to run the schedule (excluding maintenance).

FLUJOS. MÍNIMO COSTE

Algoritmo 2 (**DUAL**, sucesivos caminos de peso mínimo)

“A procedure for determining a family of minimum cost flow networks”, Busacker, Gowen, 1961

Algoritmo 1. PRIMAL
Se parte de una solución de máximo flujo y en cada paso se intenta rebajar el coste hasta alcanzar el flujo óptimo (de mínimo coste)

Algoritmo 2. DUAL
Se parte de un flujo óptimo (mínimo coste) y en cada paso se intenta aumentar su valor hasta alcanzar   flujo óptimo de valor máximo

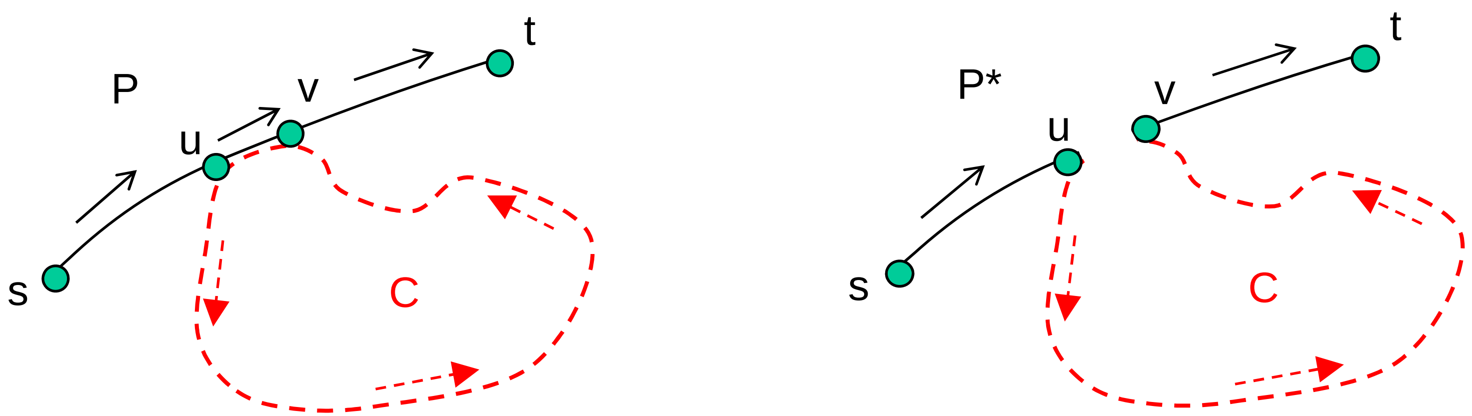
El algoritmo 1 se puede aplicar a redes con acotación inferior, pero el algoritmo 2 necesita que la acotación inferior sea d(e) = 0 para toda arista. En este caso el flujo nulo es óptimo.

MÍNIMO COSTE Algoritmo 2 (Caminos mínimos)

Lema básico

Si f es un flujo óptimo de valor val(f) con red asociada D_f, P es un camino de f-aumento de peso mínimo en D_f con residuo α y f* el flujo obtenido aumentando f a lo largo de P (α unidades) entonces f* es óptimo y su valor es val(f*) = val(f) + α

Dem.: Si f* no fuera óptimo entonces la red auxiliar D_{f*} tendría un ciclo C de peso negativo. Como este ciclo no existe en D_f , C contiene un arco e = vu de peso w(e) = – k, correspondiente al arco uv de P



$$w(P^*) - w(P) = -k + w(C - vu) = w(C) < 0$$

Y así P no sería el camino de aumento de peso mínimo

MÍNIMO COSTE   Algoritmo 2 (Caminos mínimos)

Estrategia

Seguir los pasos del algoritmo de Edmonds-Karp de aumento de flujo eligiendo en cada paso un camino de peso mínimo en la red auxiliar D_f

El primer flujo f_0 (flujo nulo) es un flujo óptimo. Los sucesivos flujos f_1, f_2, f_3, … seguirán siendo óptimos por el lema anterior.
Por tanto, el último flujo, de valor máximo, será también óptimo.
Será el flujo de valor máximo y mínimo coste.

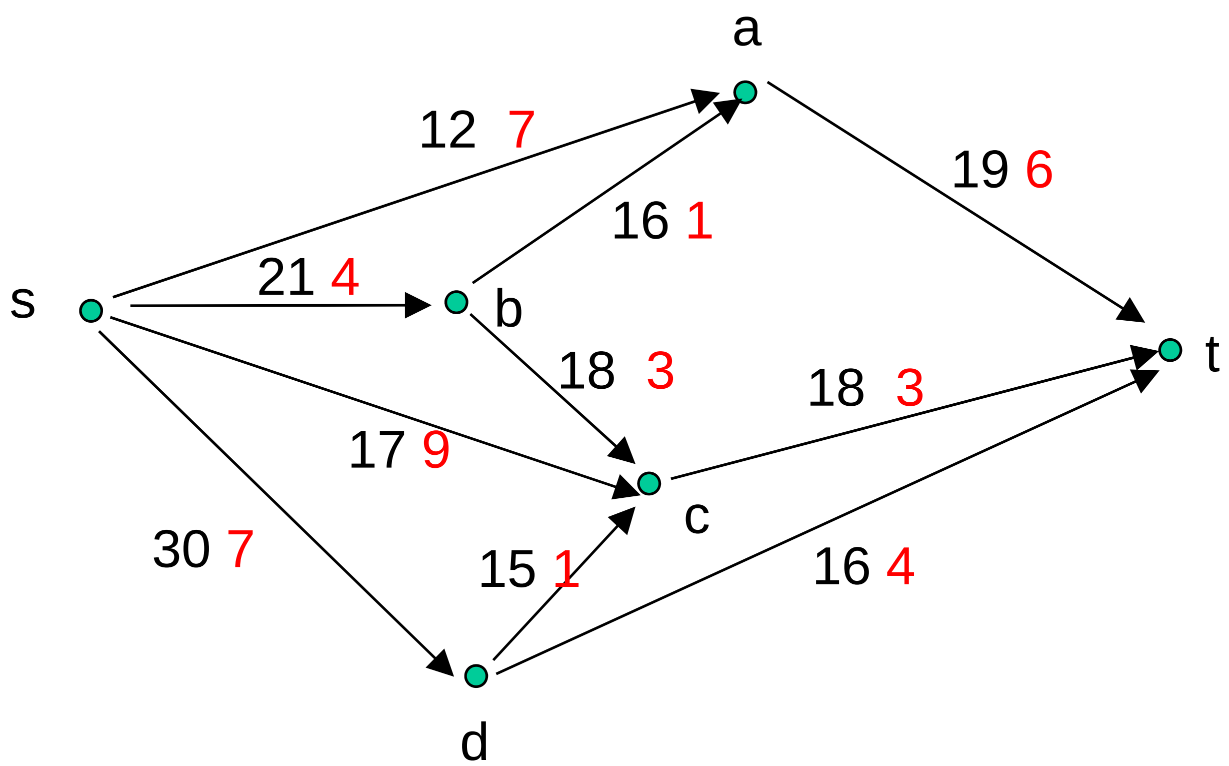
Complejidad

El algoritmo NO es polinómico, depende del valor del flujo máximo M.
La construcción de D_f en cada paso se realiza en O(m)
El camino de peso mínimo se realiza en $O(n^3)$ (Alg. Floyd-Warsahll)
En total $O(Mn^3)$ que se puede rebajar a O(M(m + nlog n)) modificando la red auxiliar para que todos los pesos sean positivos y así aplicar Dijkstra

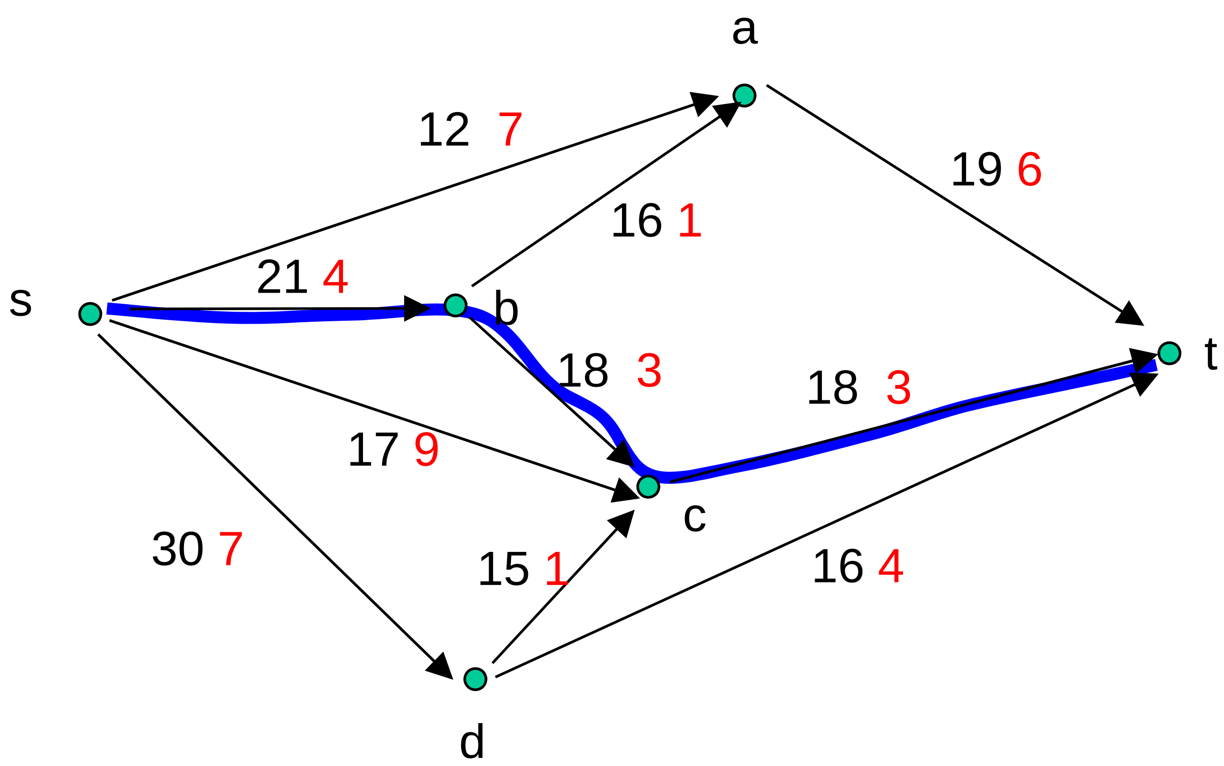
MÍNIMO COSTE Algoritmo 2 (Caminos mínimos)

Ejemplo: Flujo máximo de mínimo coste de s hasta t

Etiquetas: flujo – capacidad – coste

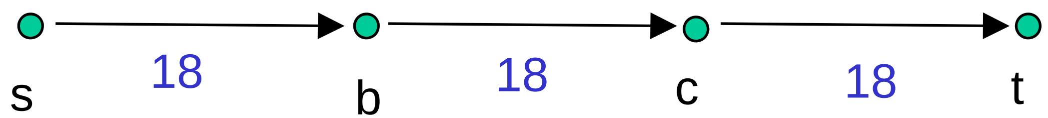


Flujo inicial $f_0 \equiv 0$

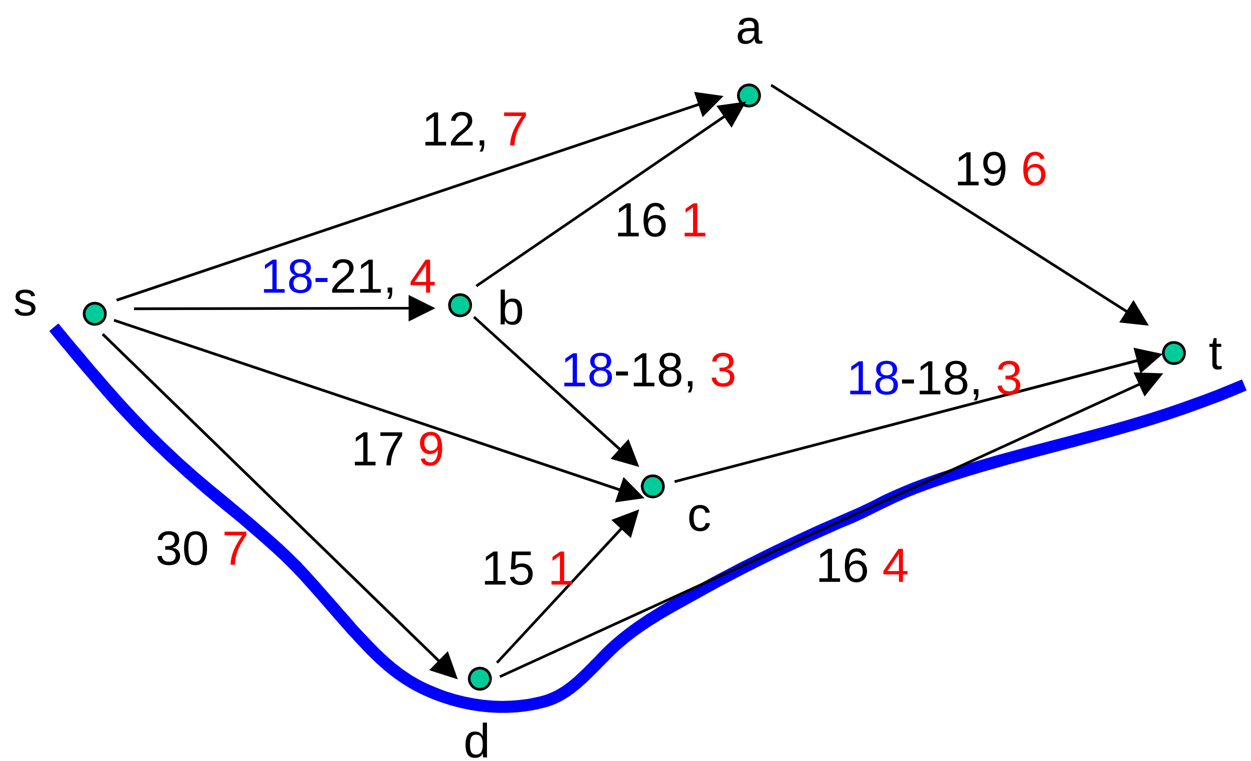


Camino de mínimo coste (coste 10) de s a t P_1 = sbct

El residuo en P_1 es 18. Se envían 18 unidades de flujo



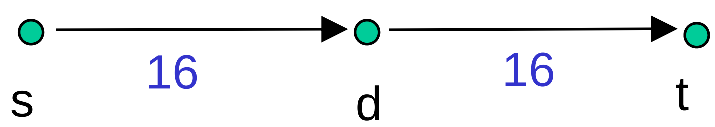
f_1



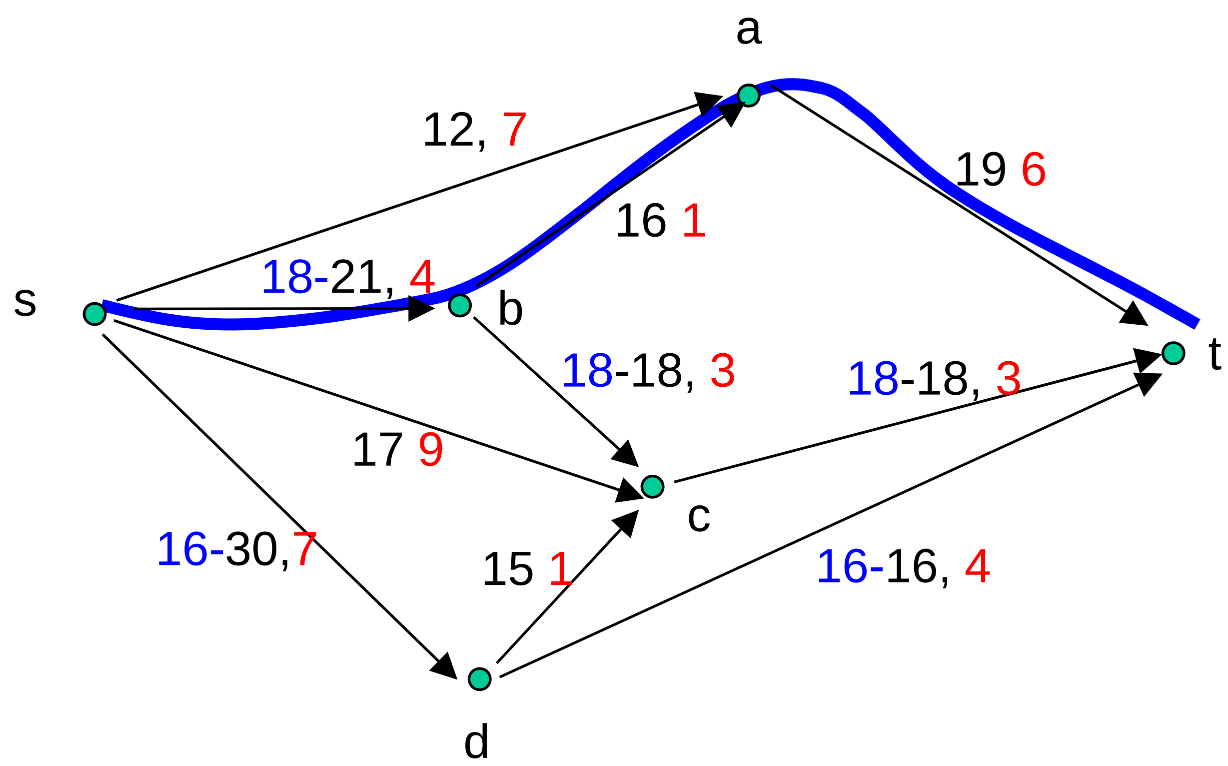
Flujo f_1

Camino de mínimo coste (coste 11) de s a t $P_2 =$ sdt

El residuo en P_1 es 16. Se envían 16 unidades de flujo



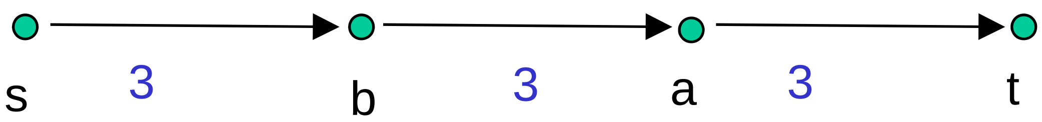
f_2



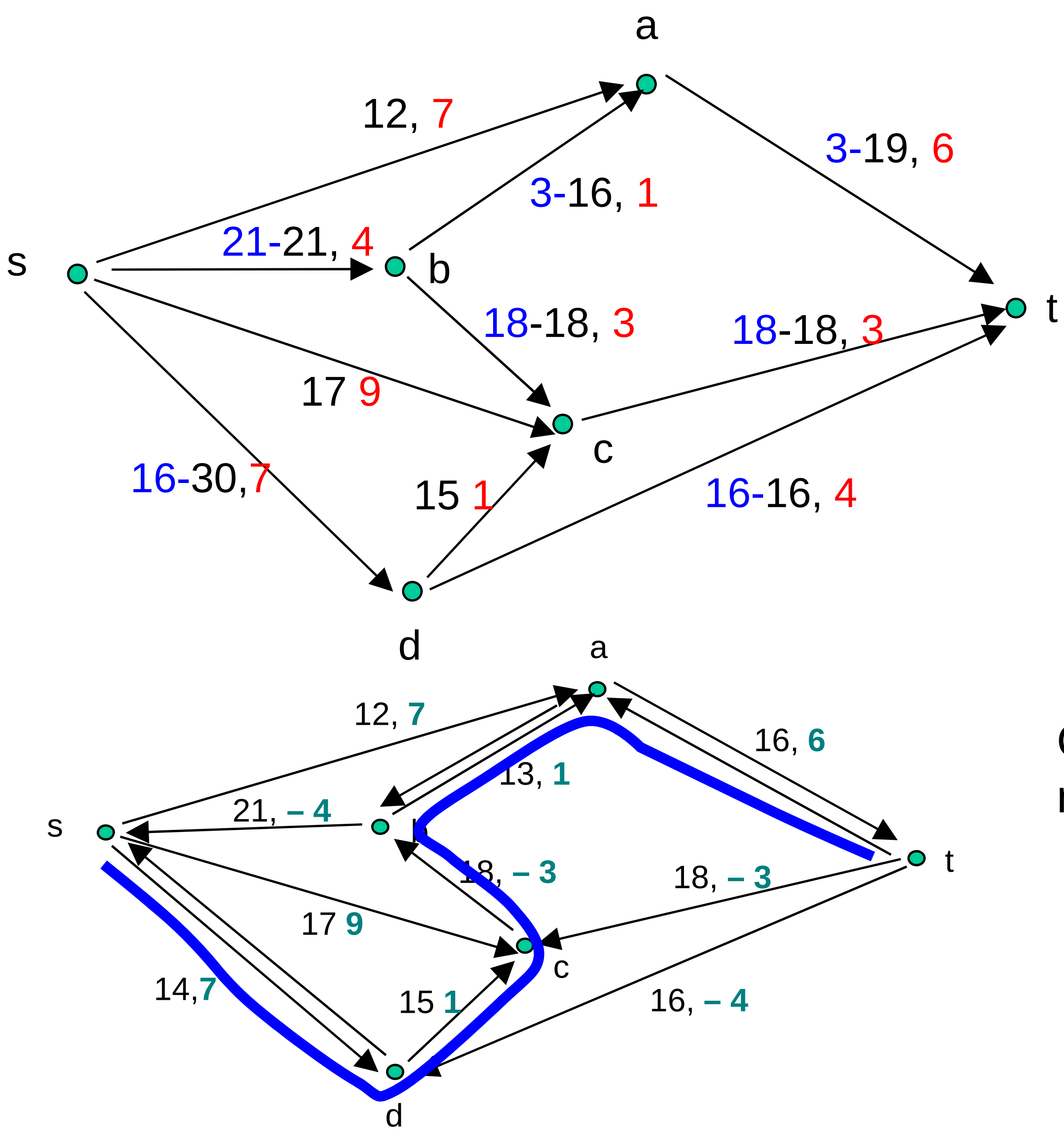
Flujo f_2

Camino de mínimo coste (coste 11) de s a t P_3 = sbat

El residuo en P_3 es 3. Se envían 3 unidades de flujo



Flujo f_3



Red auxiliar Df_3

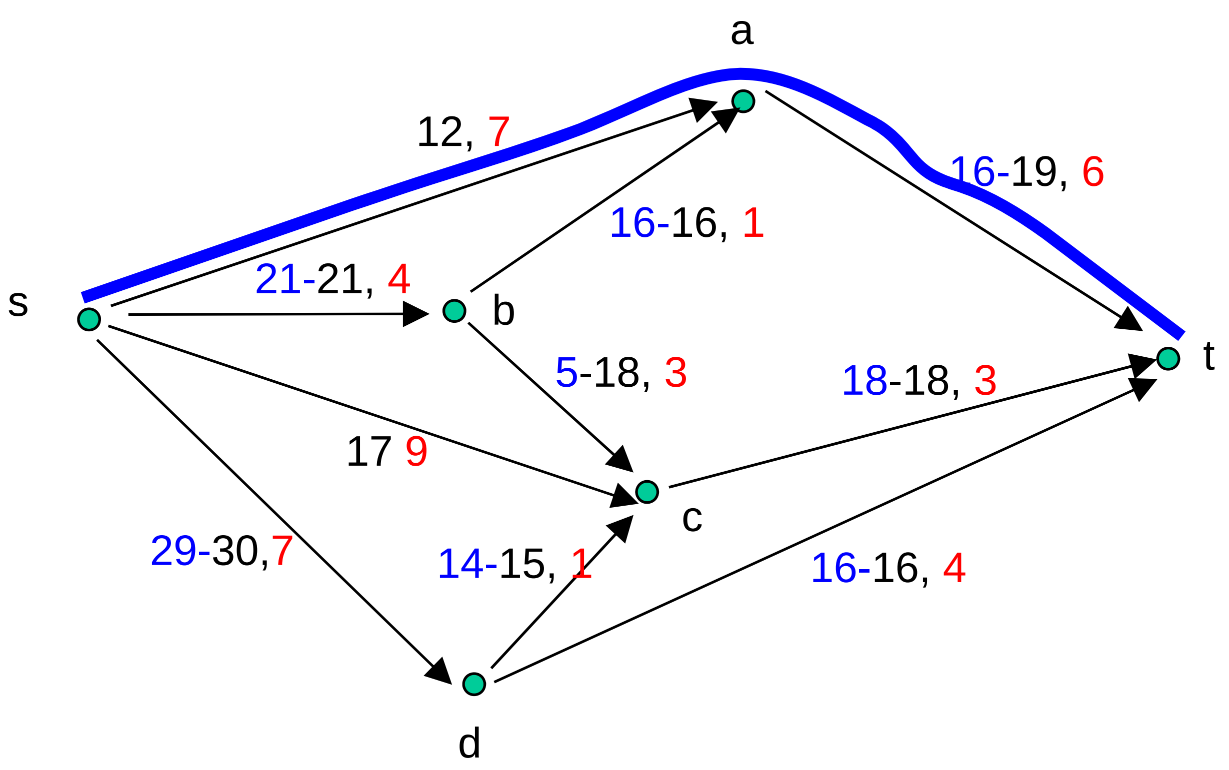
Camino de peso mínimo P_4 = sdcbat

$w(P_4) = 12$

$res(P_4) = 13$

⇓

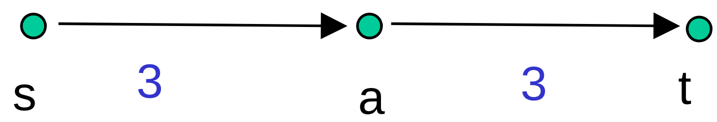
Flujo f_4



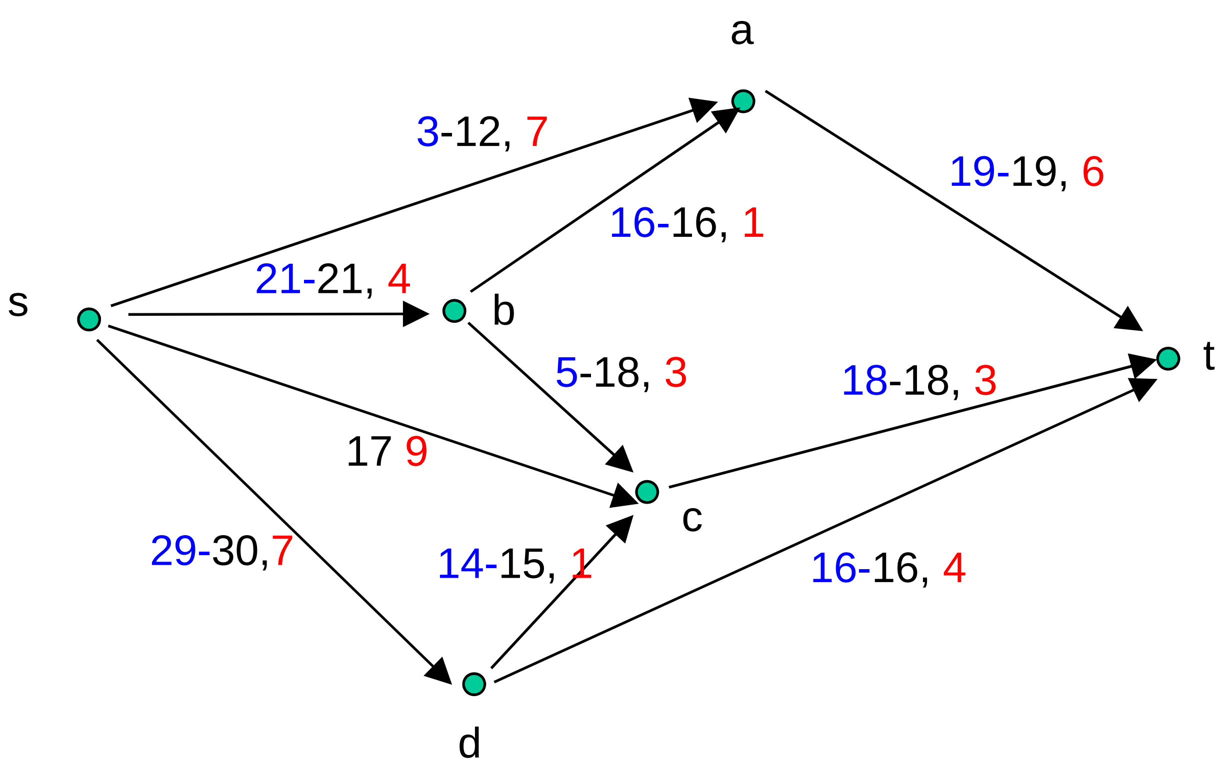
Flujo f_4

Camino de mínimo coste (coste 13) de s a t P_5 = sat

El residuo en P_5 es 3. Se envían 3 unidades de flujo



Flujo f_5



Flujo f_5

No hay caminos de f_5-aumento de s a t.

Se ha alcanzado el flujo máximo

El flujo f_5 es de valor máximo y de coste mínimo