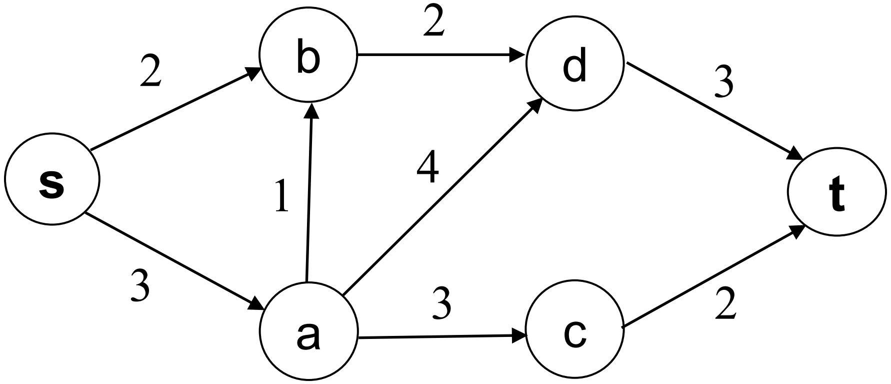
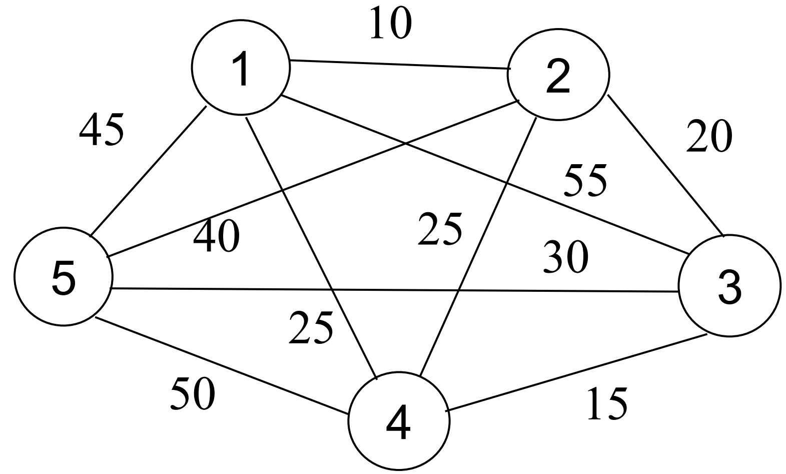
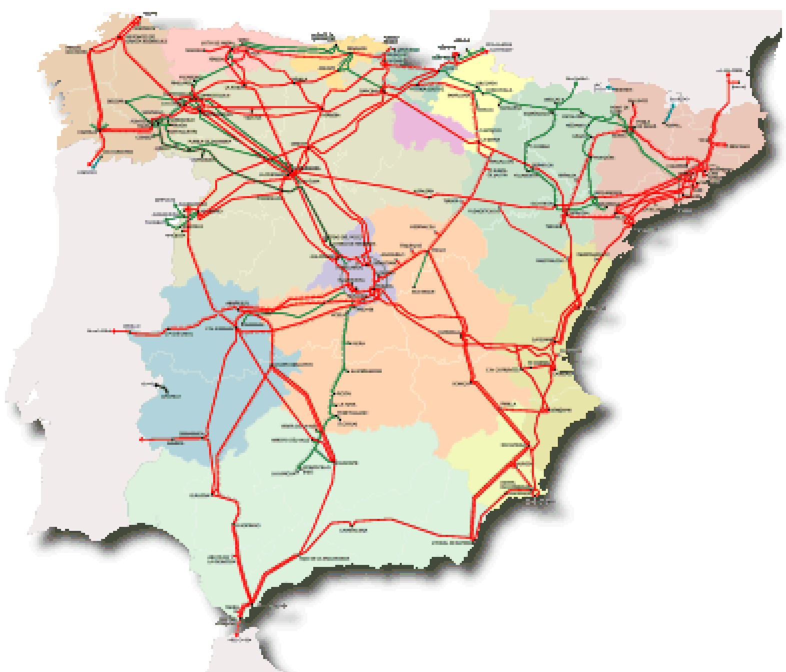
Flujo máximo: Redes de flujo y método de Ford-Fulkerson

Jose Aguilar





Flujo en Redes. Flujo máximo



Algoritmos de Flujos

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

- 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

$\Sigma f_i = \Sigma f_o$ (por nodo).

Equivalente a las leyes de conservación de la materia en física y leyes de Kirchoff en electricidad.

Algoritmos de Flujos

Usos: modelado de flujo en tuberías, piezas a través de líneas de ensamblado, corrientes en redes eléctricas, información en redes de comunicación, etc.

Una red de flujo $G=(N,A)$ es un grafo dirigido tal que cada arco $(u,v) \in A$ posee una capacidad $c(u,v) \geq 0$. Si $(u,v) \notin A$, $c(u,v)=0$.

- Se distinguen 2 vértices, el fuente "s" y el destino "t". Cada vértice $v \in N$ esta en algún s~>v~>t.
- El grafo es conexo $\Rightarrow |A| \geq |N|-1$

peso de las aristas representa la capacidad máxima de transportar un flujo.

Algoritmos de Flujos

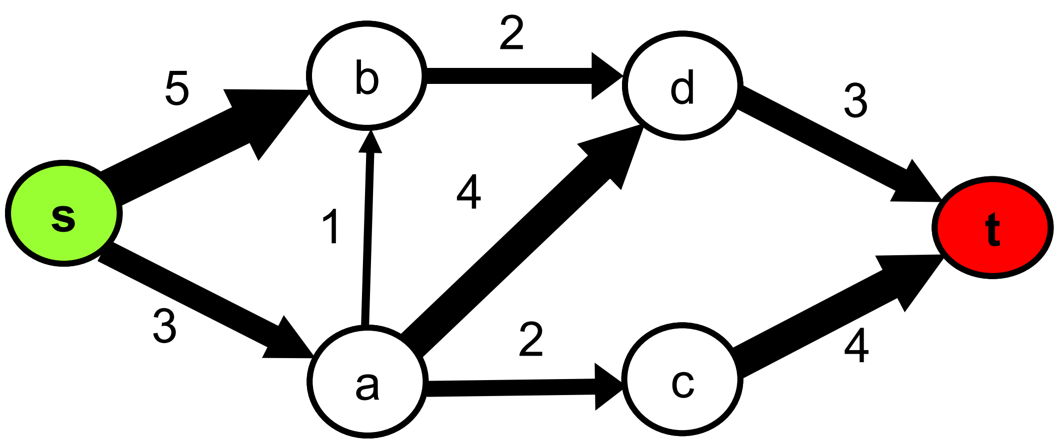
- ***Problema*: Maximizar la cantidad de flujo desde un vértice fuente a otro sumidero, sin superar las restricciones de capacidad.**
  - **Método de Ford-Fulkerson para resolver el problema de máximo flujo.**

Problema del flujo máximo: encontrar la rata máxima a la cual el material puede ser transportado de una fuente (s) a un destino (t), sin violar las restricciones de capacidad de la red.

Flujo Máximo

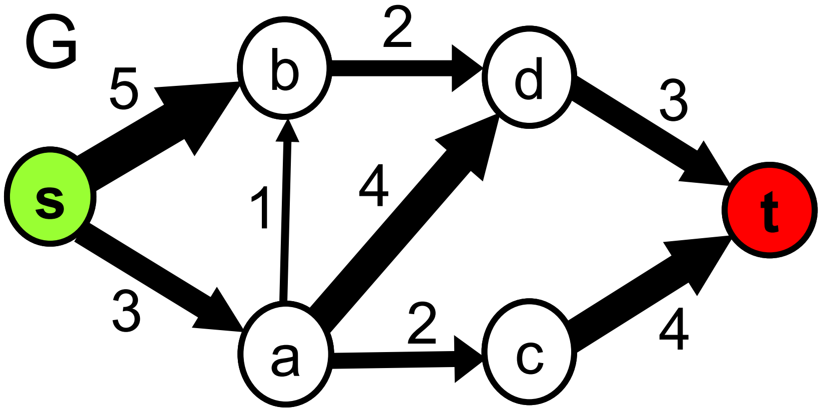
- **Restricciones:**
  - La suma de las entradas de cada nodo interior debe ser igual a la suma de sus salidas.
  - Los valores de flujo en cada arista no pueden superar los valores máximos.

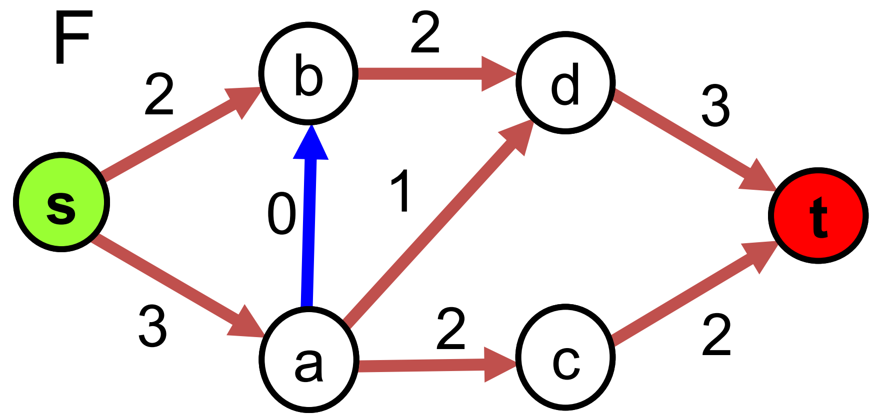
G



Flujo Máximo

- **Solución. G:** grafo del problema. **F:** grafo resultante.





- El problema se puede resolver de forma eficiente.

- **Posible algoritmo:**
  - Encontrar un camino cualquiera desde **s** hasta **t**.
  - El máximo flujo que puede ir por ese camino es el mínimo coste de las aristas que lo forman, **m**.
  - Sumar **m** en el camino en **F**, y restarlo de **G**.
- **Ojo:** este algoritmo no garantiza solución óptima.

Redes de flujo

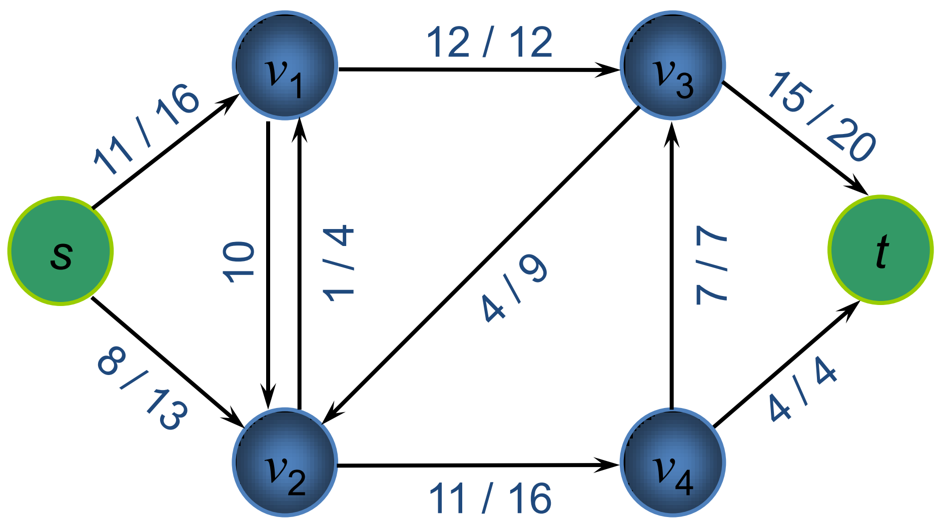
- Digrafo G=(V, E)
- Los pesos de las aristas representan capacidad (c(u, v)> 0). Si no hay aristas la capacidad es cero.
- Vértices especiales:

  fuente *s*, vértice sin aristas de entrada.

  sumidero *t*, vértice sin aristas de salida.
- El grafo es conectado: Hay un camino entre *s* y *t* por algún vértice intermedio del grafo.

Redes de flujo

- Un flujo en G es una función real f : VxV $\rightarrow \mathfrak{R}$ que satisface las siguientes propiedades:
  - Restricción de capacidad: Para todo $u, v \in$ V, $f(u, v) \leq c(u, v)$
  - Antisimetría: Para todo $u, v \in$ V, $f(u, v) = -f(v, u)$
  - Conservación de flujo: Para todo $u \in V - \{s, t\}$, = 0
- Valor del flujo: | f | = =



El problema del flujo máximo

Se define como: dado G, s (fuente) y t (destino)

encontrar el flujo neto total máximo desde s hasta t

Flujo neto positivo entrando a un nodo v es

$$\Sigma_{u \in N} F(u,v)$$

y el **que sale se define simétricamente**.

Algoritmo de Ford-Fulkerson

Es un método iterativo que depende de tres ideas importantes:

- Red residual
- Aumento de camino
- Cortes

Usa el teorema max-flow min-cut que caracteriza el flujo máximo en términos de cortes de la red de flujo.

En cada iteración se va consiguiendo un valor de flujo que aumenta el camino, es decir, podemos aumentar el flujo en un camino de *s* a *t*. Este proceso se repite hasta que no haya más posibilidad de aumentar.

Flujo residual

Es el flujo disponible en una determinada arista una vez que se ha enviado flujo por ella (en ningún caso el flujo neto residual debe ser mayor a la capacidad de dicha arista ni menor que cero).

flujo residual = capacidad – flujo_actual,

Dada una red de flujo G=(N,A) con fuente s y destino t Sea f el flujo en G, y considere un par de vertices u,v $\in$N.

La cantidad de flujo adicional que se puede verter sobre u,v es la **capacidad residual**.

$$c_f(u,v) = c(u,v) - f(u,v)$$

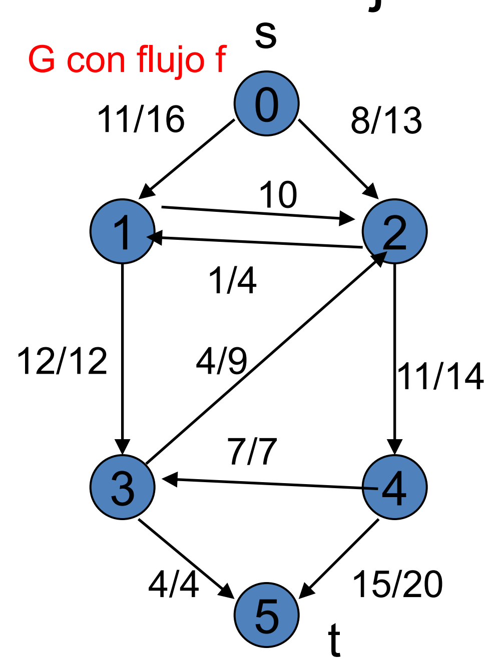
Red residual

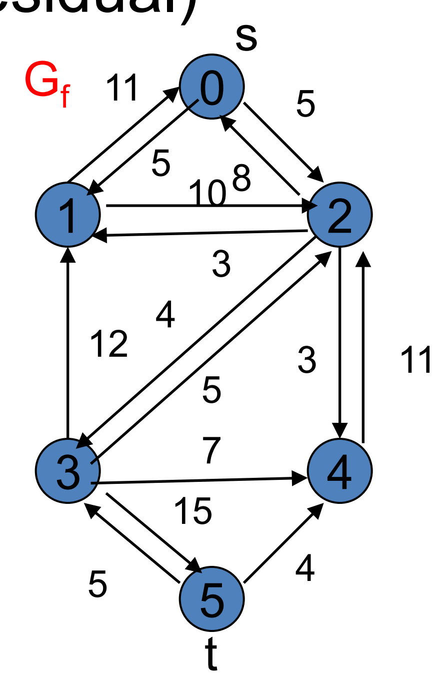
Un **camino de flujo residual** es aquel camino de la fuente al sumidero donde todas las aristas en el camino tienen un flujo residual mayor a cero

red residual consiste en arcos que admiten más flujo (en ella aparecen los caminos de flujo residual)

Un arco (u, v) aparece en G_f solo si (u, v)∈A y si hay una flujo neto positivo o flujo residual de u a v o (v, u)∈A y esta pasando un flujo actual entre (v, u)

$$| A_f | \leq 2|A|$$





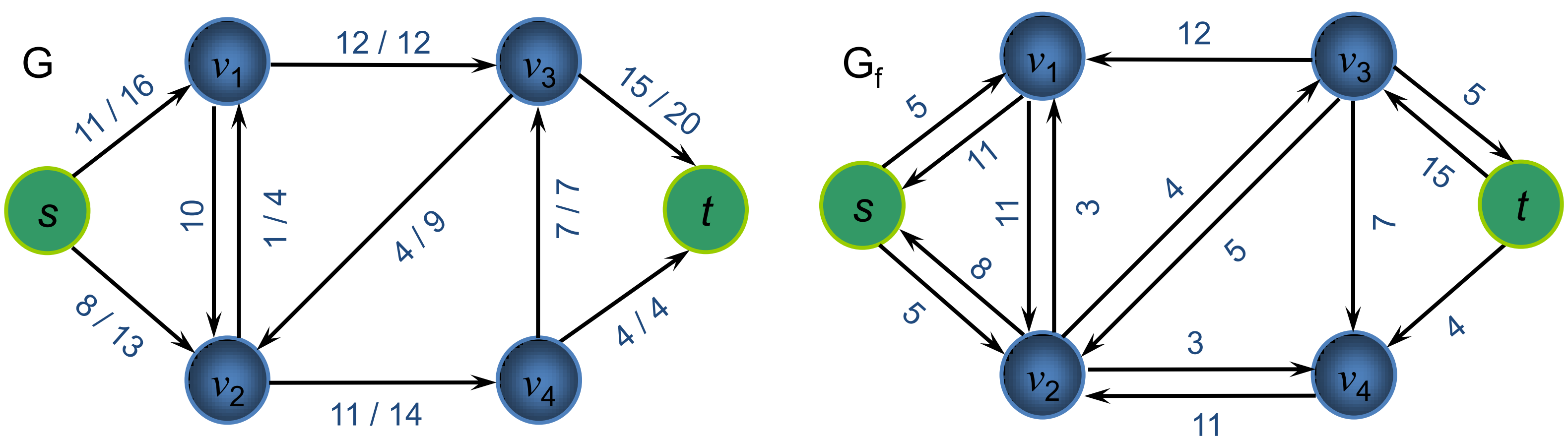
Redes residuales

- Para una red de flujo y un flujo, la red residual es el conjunto de aristas que pueden admitir más flujo.
- Sea una red de flujo G=(V, E) con fuente *s* y sumidero *t*. Sea *f* un flujo en G y un par de vértices $u, v \in$ V. El flujo neto adicional desde *u* a *v* sin exceder la capacidad c(*u*, *v*) es la ***capacidad residual*** de (*u*, *v*), definida por:

$$c_f(u,v) = c(u,v) - f(u,v)$$

- La ***red residual*** de G inducida por *f* es G_f = (V, E_f) donde

$$E_f = \{(u,v) \in V \times V : c_f(u, v) > 0\}$$



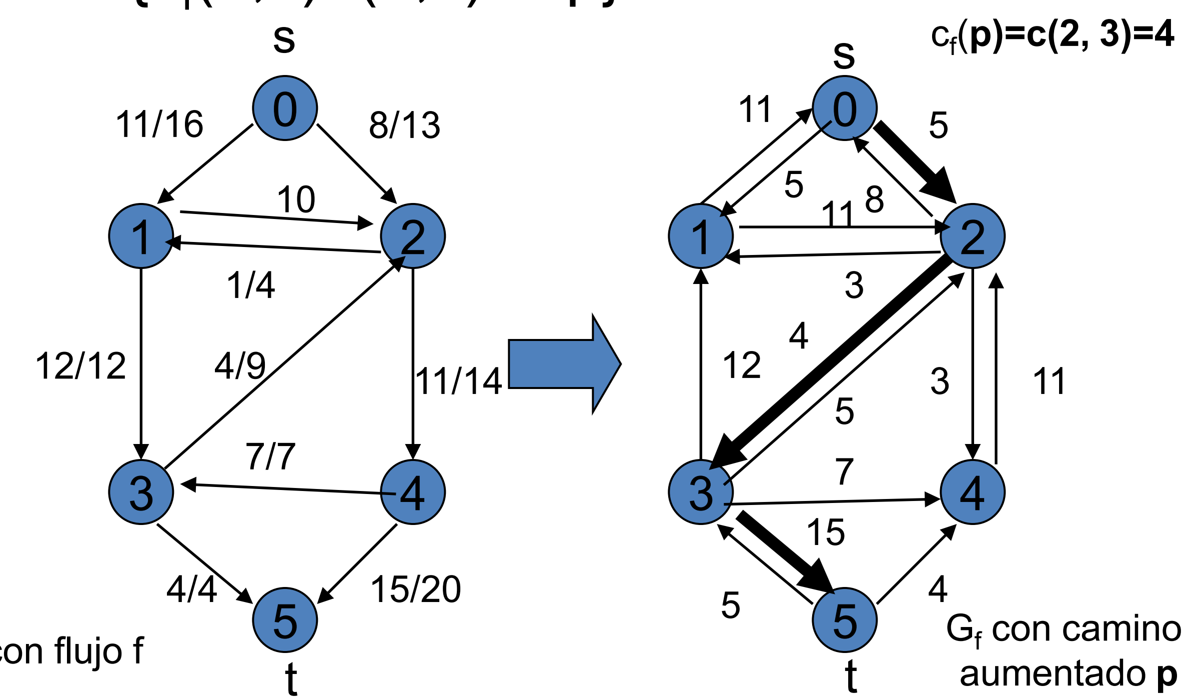
Aumento de camino

Dada una red de flujo G=(N,A), un camino aumentado *p* es un camino simple de *s* a *t* en la red residual G_f.

La cantidad máxima de flujo que puede llevarse por los arcos en un camino aumentado *p* se denomina **capacidad residual** de *p*, y está dado por:

$$c_f(p) = \min \{c_f(u,v) : (u,v) \in p\}$$

El método busca repetidamente aumentar el flujo a través de los caminos de aumento hasta alcanzar el máximo



G con flujo f

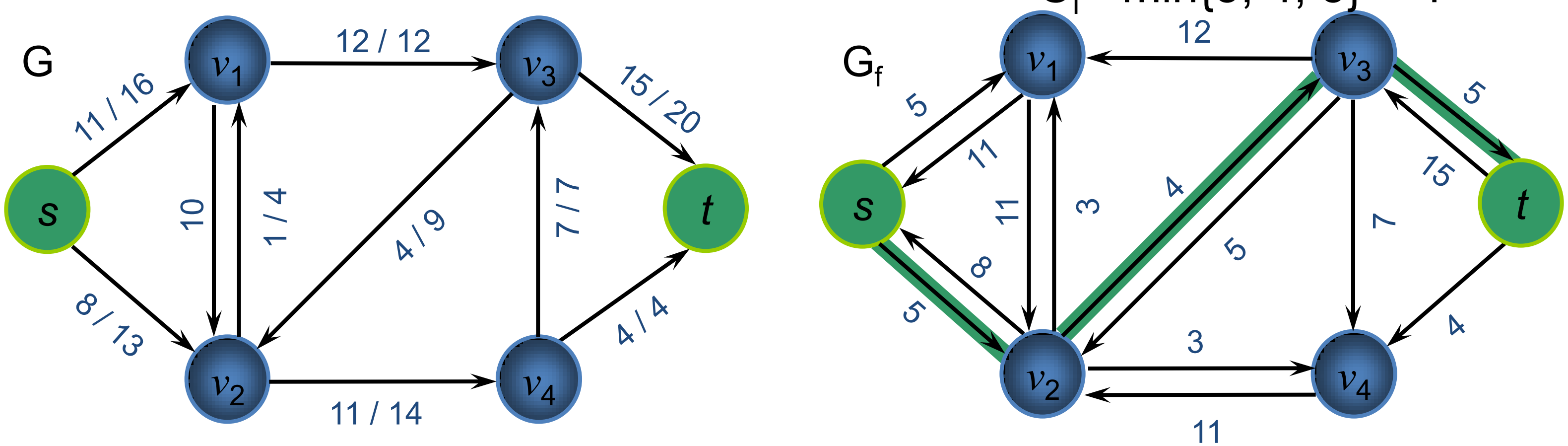
G_f con camino aumentado **p**

Caminos aumentantes

- Un camino aumentante p en una red de flujo G=(V, E) y flujo *f*, es un camino simple de *s* a *t* en la red residual G_f.
- Cada arista (*u*, *v*) del camino aumentante admite un flujo neto positivo adicional de *u* a *v* sin violar la restricción de capacidad de la arista.
- Capacidad residual: es la máxima cantidad de flujo neto que se puede enviar por las aristas de un camino aumentante. Se calcula por:

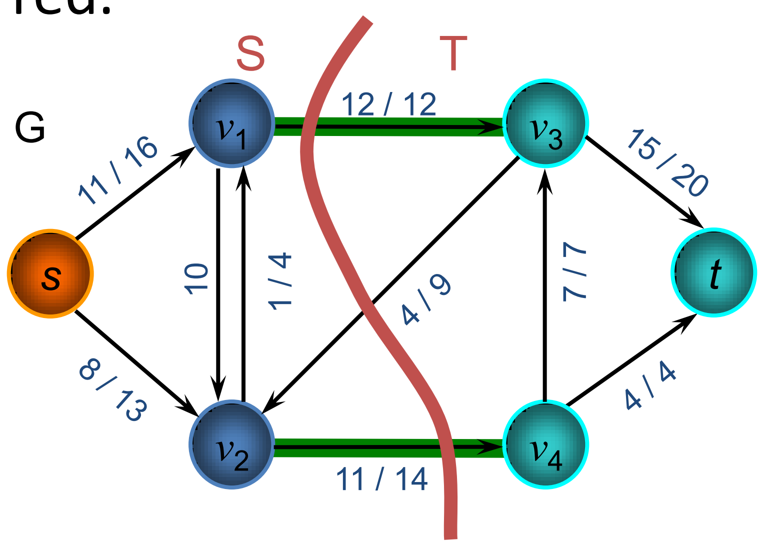
$$c_f(p) = \min\{c_f(u,v) \ / \ (u,v) \in p\}$$

$C_f = \min\{5, 4, 5\} = 4$



Cortes en redes de flujo

- Un corte (S, T) de una red de flujo G=(V, E) es una partición del conjunto de vértices V en dos subconjuntos S y T = V–S tal que $s \in$ S y $t \in$ T.
- Si *f* es un flujo:
  - *f*(S, T) es el flujo neto a través del corte (S,T).
  - c(S, T) es la capacidad del corte (S,T).
- Flujo en una red = flujo neto a través de cualquier corte de la red.



Corte = ({s, v_1, v_2}, {s, v_1, v_2})

$f(s, t) = f(v_1, v_3) + f(v_2, v_3) + f(v_2, v_4) =$
$12 + (-4) + 11 = 19$

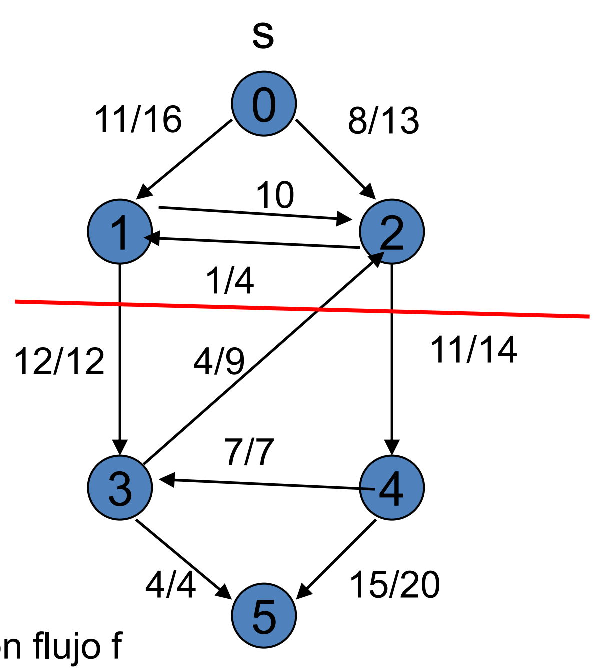
$c(s, t) = c(v_1, v_3) + c(v_2, v_4) = 12 + 14 = 26$

Teorema flujo-máximo mínimo-corte

- Si f es un flujo en una red de flujo G = (V, E) con fuente s y sumidero t, entonces las siguientes condiciones son equivalentes:
  - f es un flujo máximo en G.
  - La red residual G_f no contiene caminos aumentantes.
  - $| f | = c(S, T)$ para algún corte (S, T) de G.

Teorema max-flow min-cut

Corte (S, T)
S={s, 1, 2}
T={3, 4, t}
f(S, T) = 19
c(S, T) = 27



G con flujo f

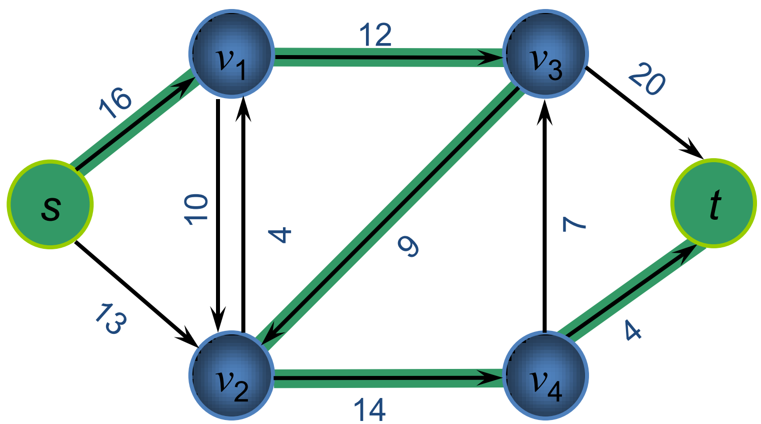
Teorema: El máximo valor de entre todos los flujos en una red es igual a la capacidad mínima de entre todos los cortes.

Algoritmo de Ford-Fulkerson

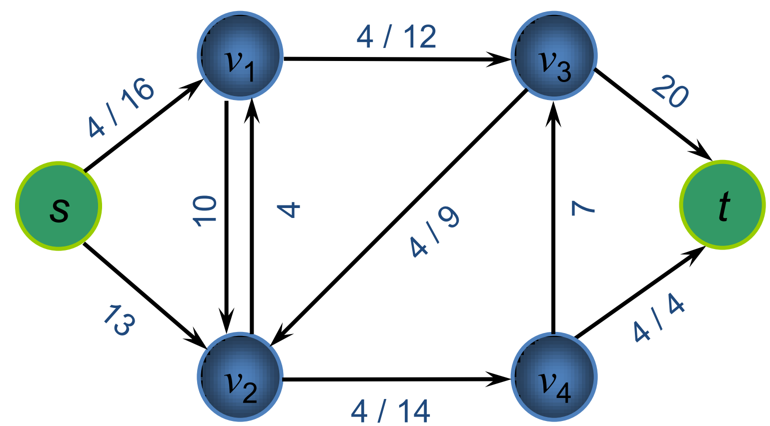
```
FORD-FULKERSON( f, s)
Para cada arista (u, v) en el grafo
        f[u][v] = 0
        f[v][u] = 0
Mientras exista un camino de flujo residual entre f y s
        incremento = min(cap(u,v) tal que (u,v) está en el
            camino p)
        para cada arista (u,v) en el camino
                f[u][v] = f[u][v] + incremento
                f[v][u] = -f[u][v]
```

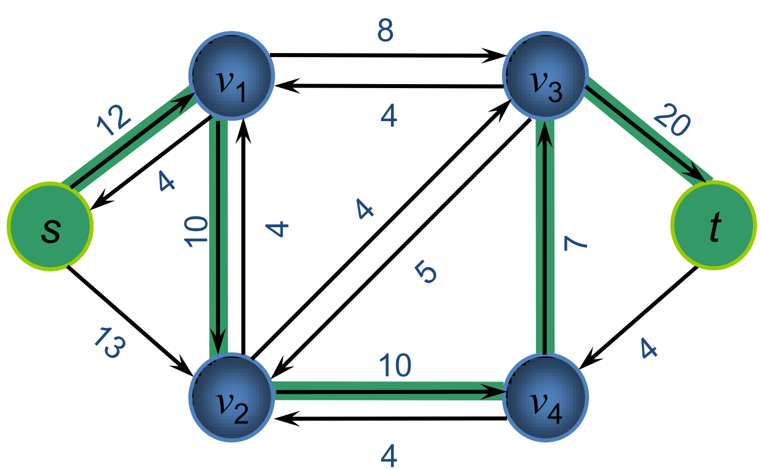
Ejemplo

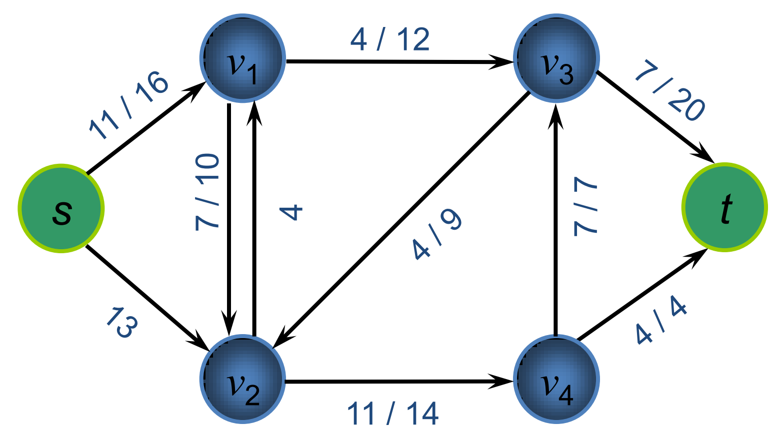
Grafo Residual



Flujo

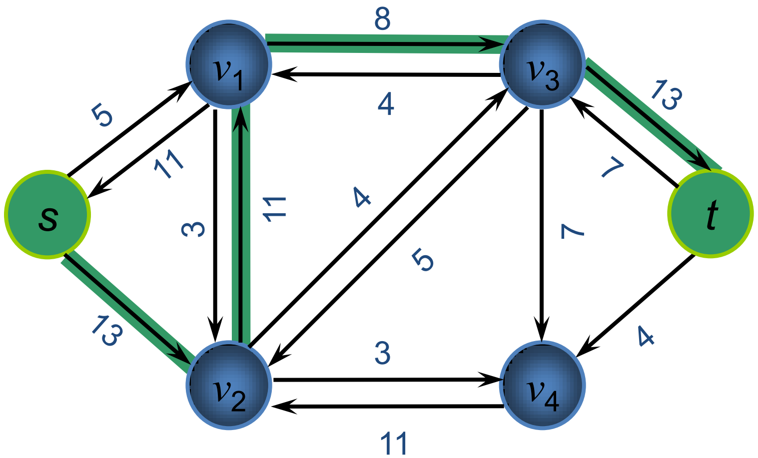


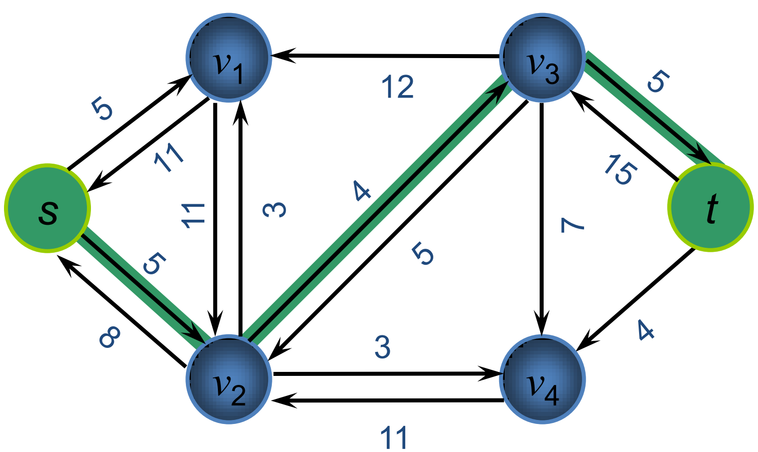




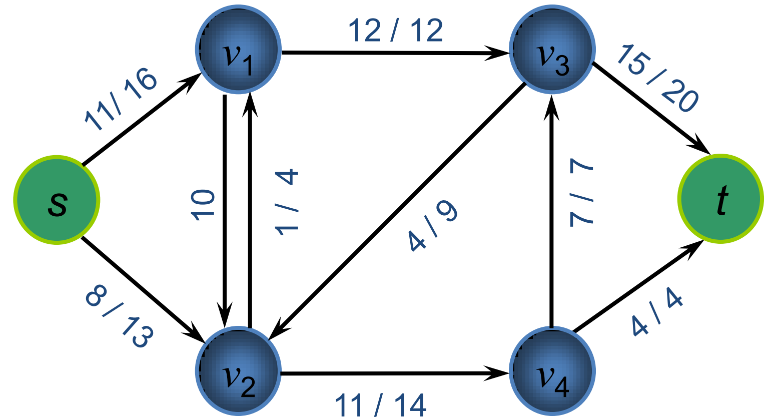
Ejemplo

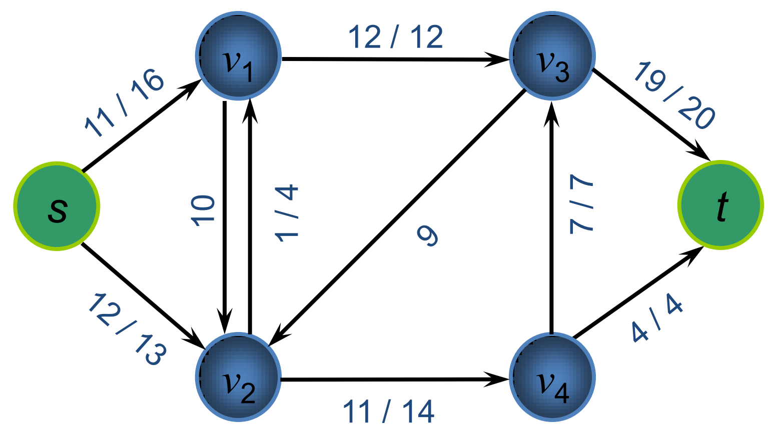
Grafo Residual





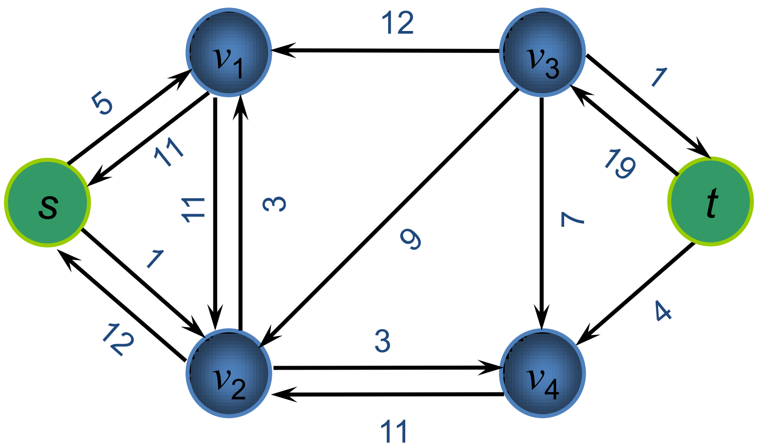
Flujo



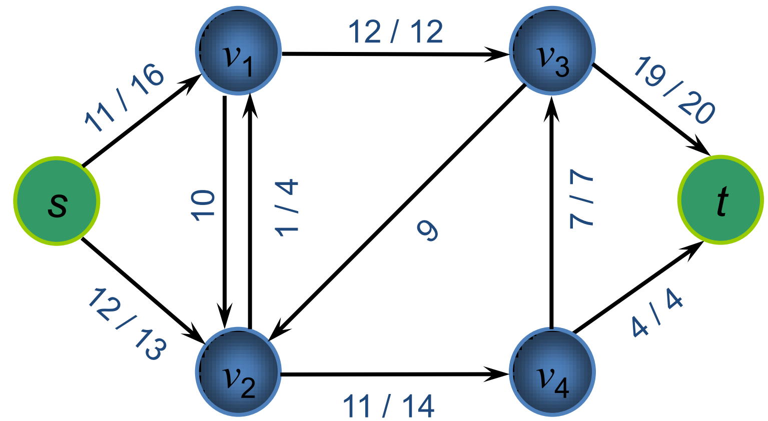


Ejemplo y complejidad

Grafo Residual

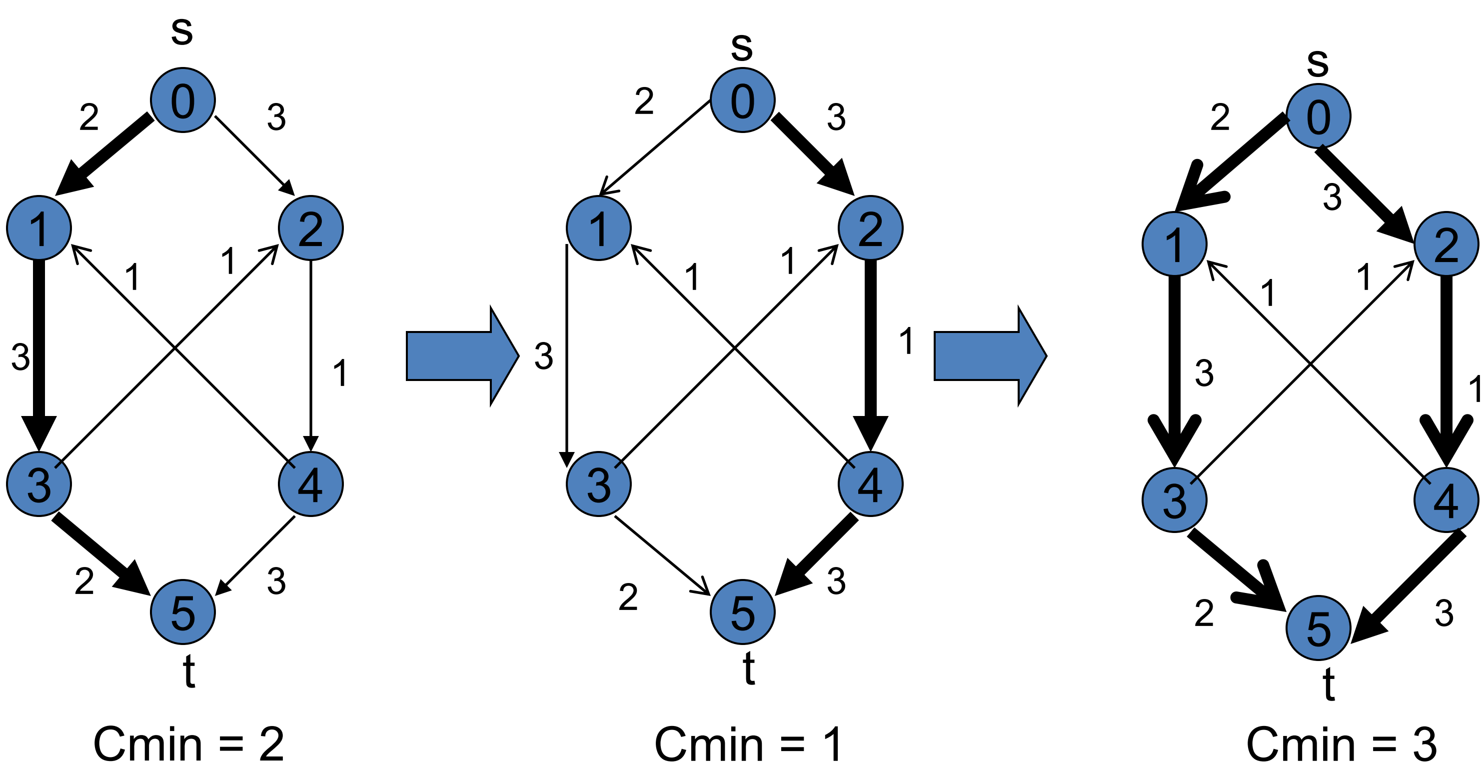


Flujo



- Para hallar el camino aumentante se puede usar cualquier tipo de recorrido (BPA o BPP).
- La capacidad de cada arista se puede multiplicar por un factor de escala para conseguir que sea entera.
- Bajo estas condiciones el algoritmo tiene una complejidad de $O(E|f^*|)$, donde f^* es el máximo flujo obtenido por el algoritmo.

Algoritmo de Ford-Fulkerson

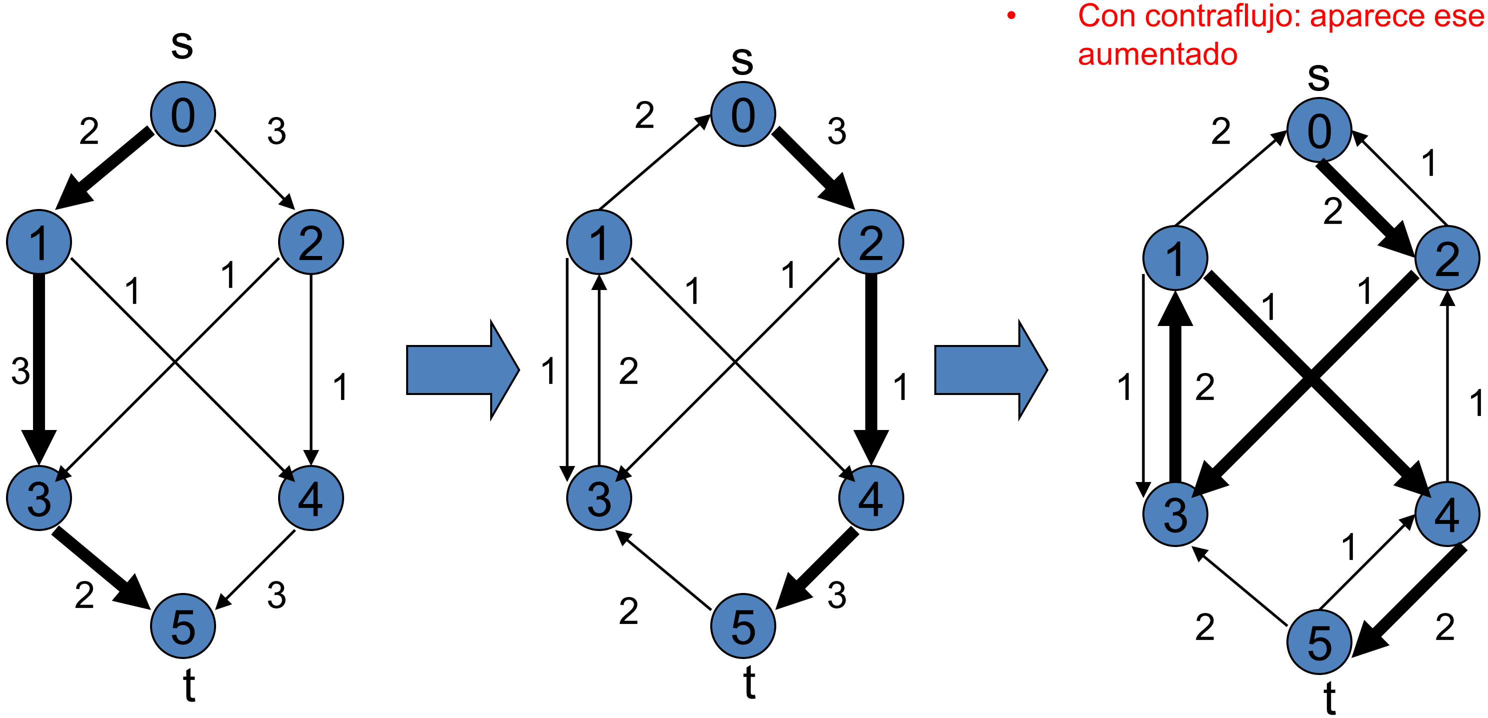


¿Qué pasa si las aristas (4,1) o (3,2) cambian de signo?

Algoritmo de Ford-Fulkerson

Dos casos

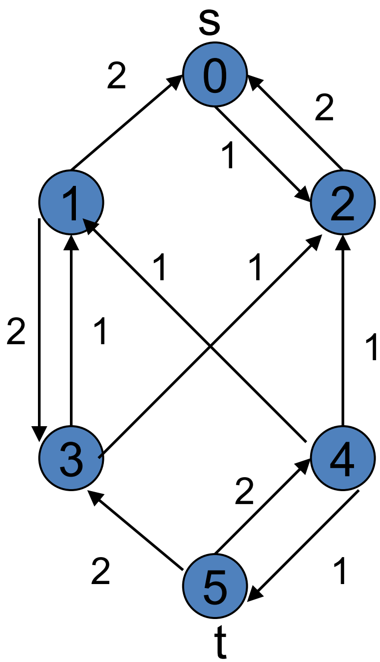
- Sin contraflujo: se detiene el algoritmo en esta iteración=> Fllujo Max.=3
- Con contraflujo: aparece ese camino aumentado

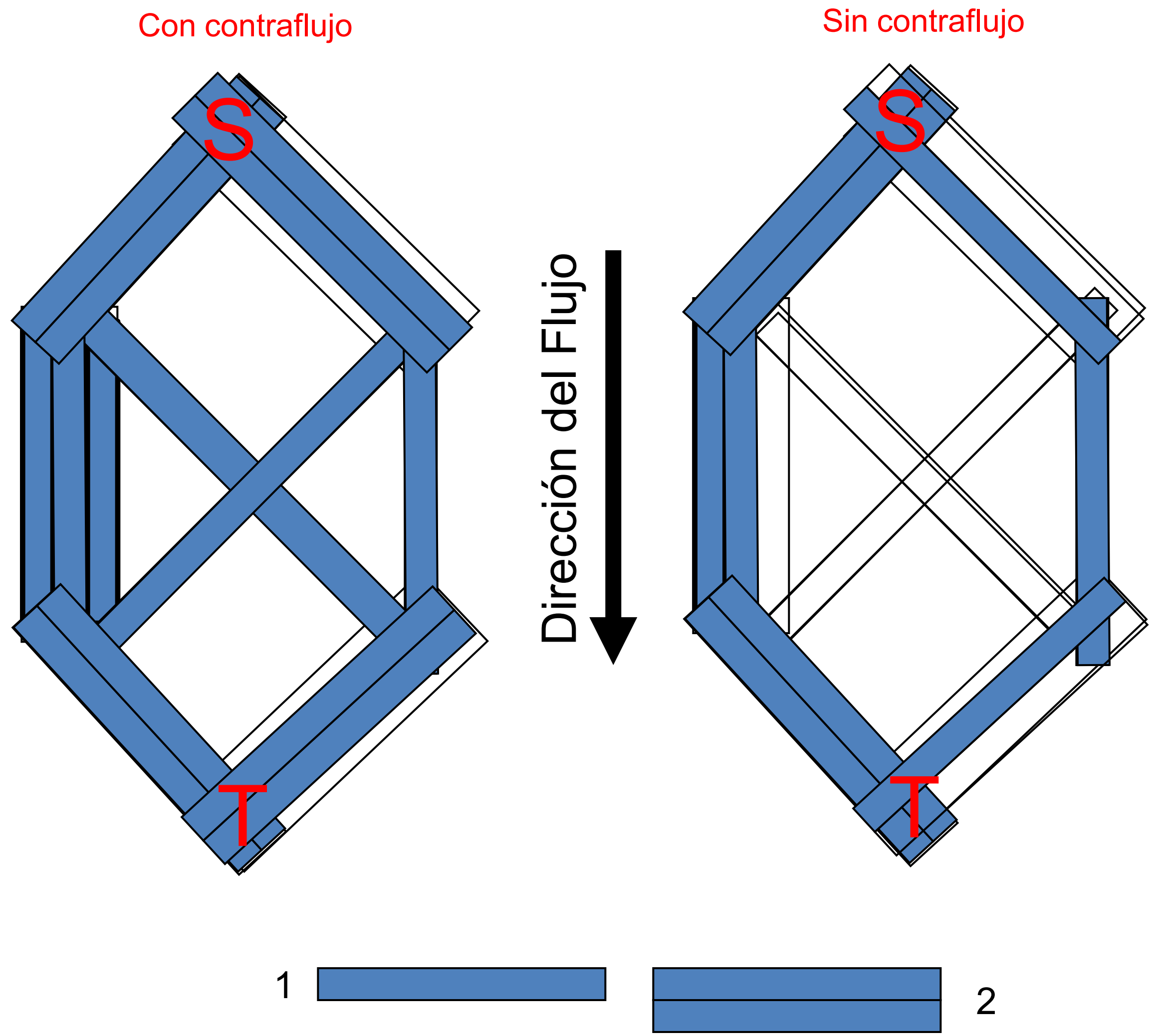


Cmin = 2 Cmin = 1 Cmin = 1

Algoritmo de Ford-Fulkerson

Flujo Máximo





Algoritmo de Ford-Fulkerson

fordFulkerson(Nodo: s, t)

{pre: n > 0 } {pos: n > 0 }

[f(u, v), f(v, u) = 0, 0] (u, v) ∈ A // Inicia f (∃ un p de s a t en Gf) [cf(p) = min(cf(u, v) : (u, v) está en p [f(u, v) = f(u, v) + cf(p) f(v, u) = - f(u, v)] (u, v) en p] // Aumenta el flujo a lo largo de p // por cf (p) hasta que no hayan mas p	-u, v: Nodo. Nodos del grafo. -Gf: Grafo. Grafo residual. -cf: Entero.Capacidad residual -p: Camino aumentado de Gf. -f: Arreglo(n x n)De [Entero]. Flujo del arco.

Otra idea basada en Corte de la red de flujo

Un corte (S,T) de la red de flujo G=(N,A) es una partición de N en S y T=N-S tal que $s \in S$ y $t \in T$.

Si f es el flujo, entonces

- El flujo de red a través del corte (S,T) se define f(S,T)
- La capacidad del corte (S,T) es c(S,T)
- El flujo neto a través del corte (S, T) es f(S, T) = | f |
- El valor de cualquier flujo f en una red de flujos G está limitado superiormente por la capacidad de cualquier corte de G.

Extensiones

Redes con varios nodos fuente y varios nodos destino:

Se reduce al anterior colocando dos nodos ficticios denominados:

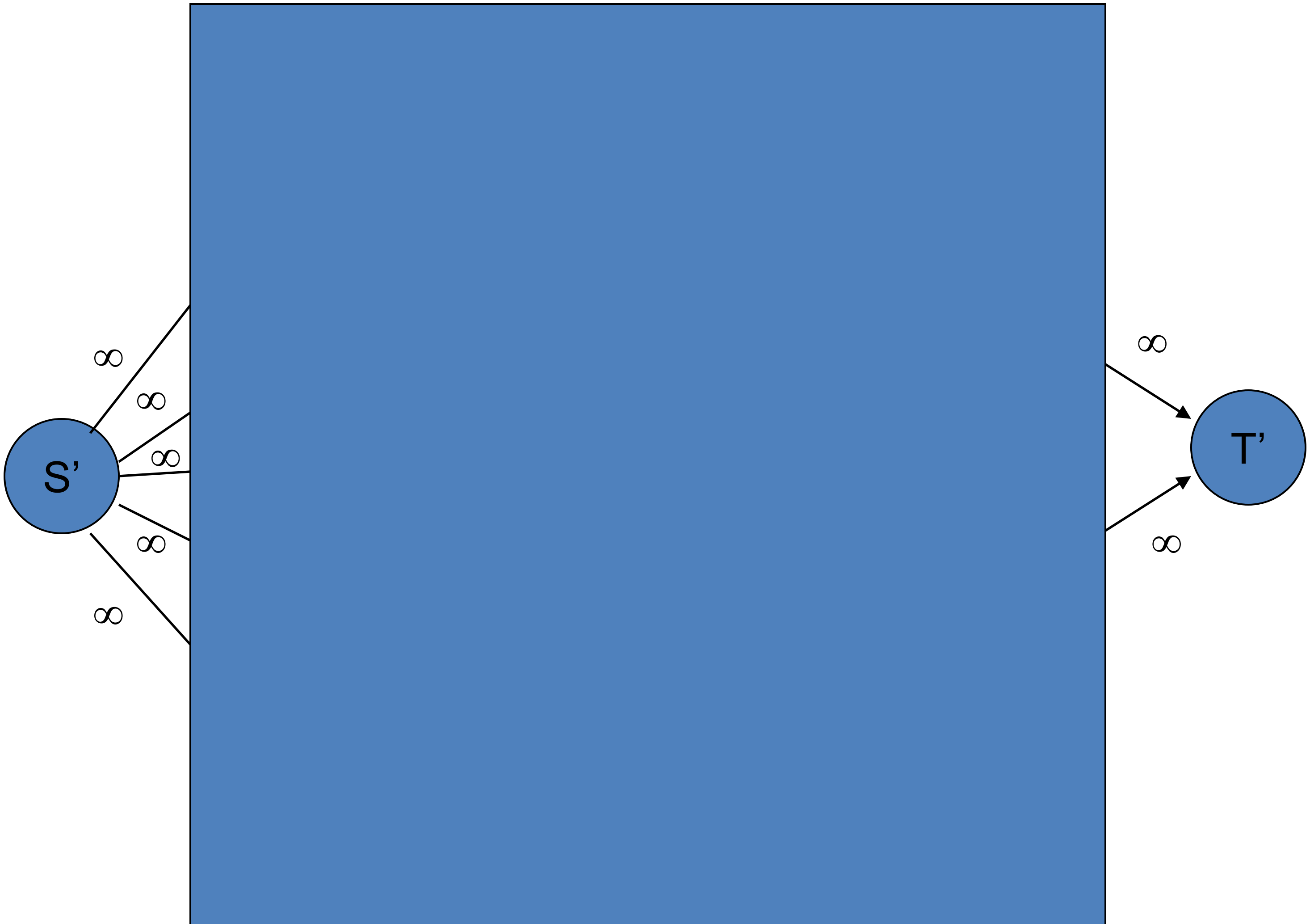
- nodo superfuente s y
- nodo superdestino t,

Anexar

- un arco (s, si) por cada nodo fuente de G para i = 1, 2, ..., m y
- un arco (tj, t) por cada nodo destino de G para j = 1, 2, ..., n.

Ambos nodos se conectan a los otros nodos con arcos de capacidad ∞.

Extensiones



Grafo bipartido

Problema interesante que se puede reducir a un problema de flujo máximo es el del **aparejamiento ("matching") bipartito máximo**.

- **Bipartido significa que se pueden identificar en el grafo dos subconjuntos de vértices** L y R de tal manera que uno de los extremos de las aristas está en R y el otro L.

Las aristas (u,v) donde u y v pertenecen al mismo conjunto no están permitidas.

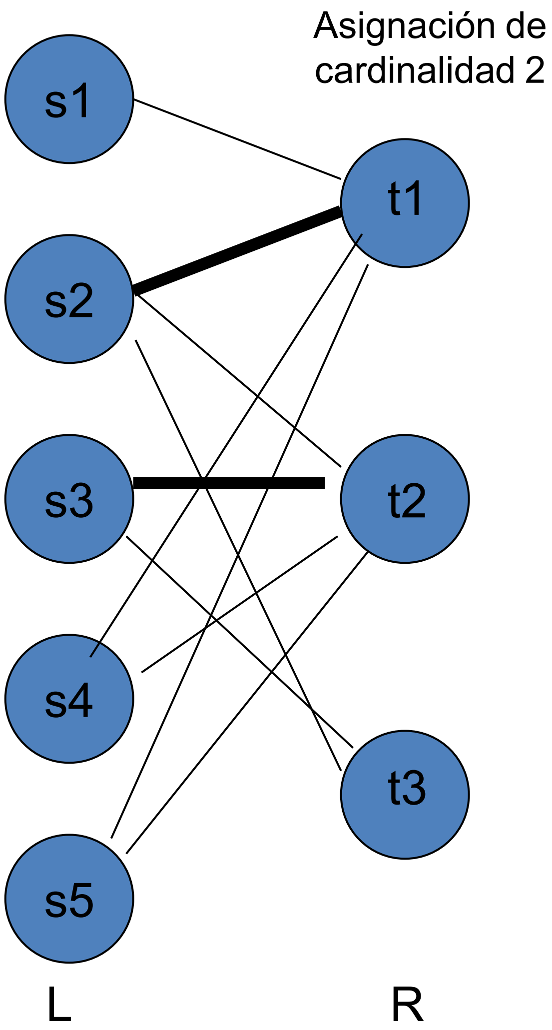
N puede ser dividido en N = L U R, L ∩ R = ∅ y todos las aristas en A van de los nodos de L a los de R.

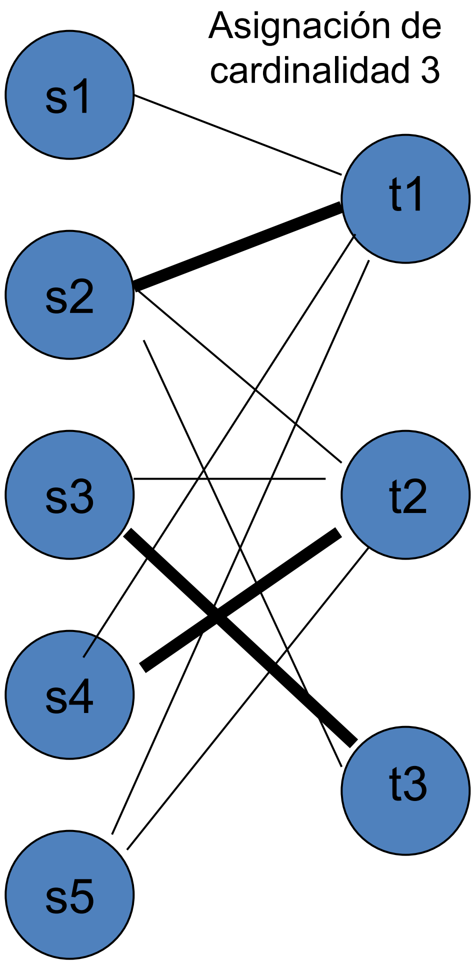
- **Este tipo de grafos sirve para resolver problemas como el de asignación de tareas**.
  - Si suponemos que el conjunto R representa a un grupo de trabajadores y que L corresponde a un conjunto de tareas,
  - Las aristas representan la relación de que un trabajador puede realizar determinada tarea.
  - El problema consiste en asignar la mayor cantidad de tareas para que sean realizadas por los trabajadores.

Grafo bipartido

Dado un grafo no dirigido G = (N, A), un "matching" es un subconjunto de aristas $M \subseteq A$, tal que para todos los nodos $v \in N$, a lo sumo una arista de M incide en v.

Un "**matching" M es máximo** si tiene una cardinalidad máxima tal que para cualquier M' se tiene $|M'| \geq |M|$.





Grafo bipartido

Se puede usar el algoritmo de Ford-Fulkerson para resolver dicho problema.

Si tenemos el grafo bipartido G = (N, A) se construye el grafo de flujos G’ = (N’, A’) de la siguiente manera

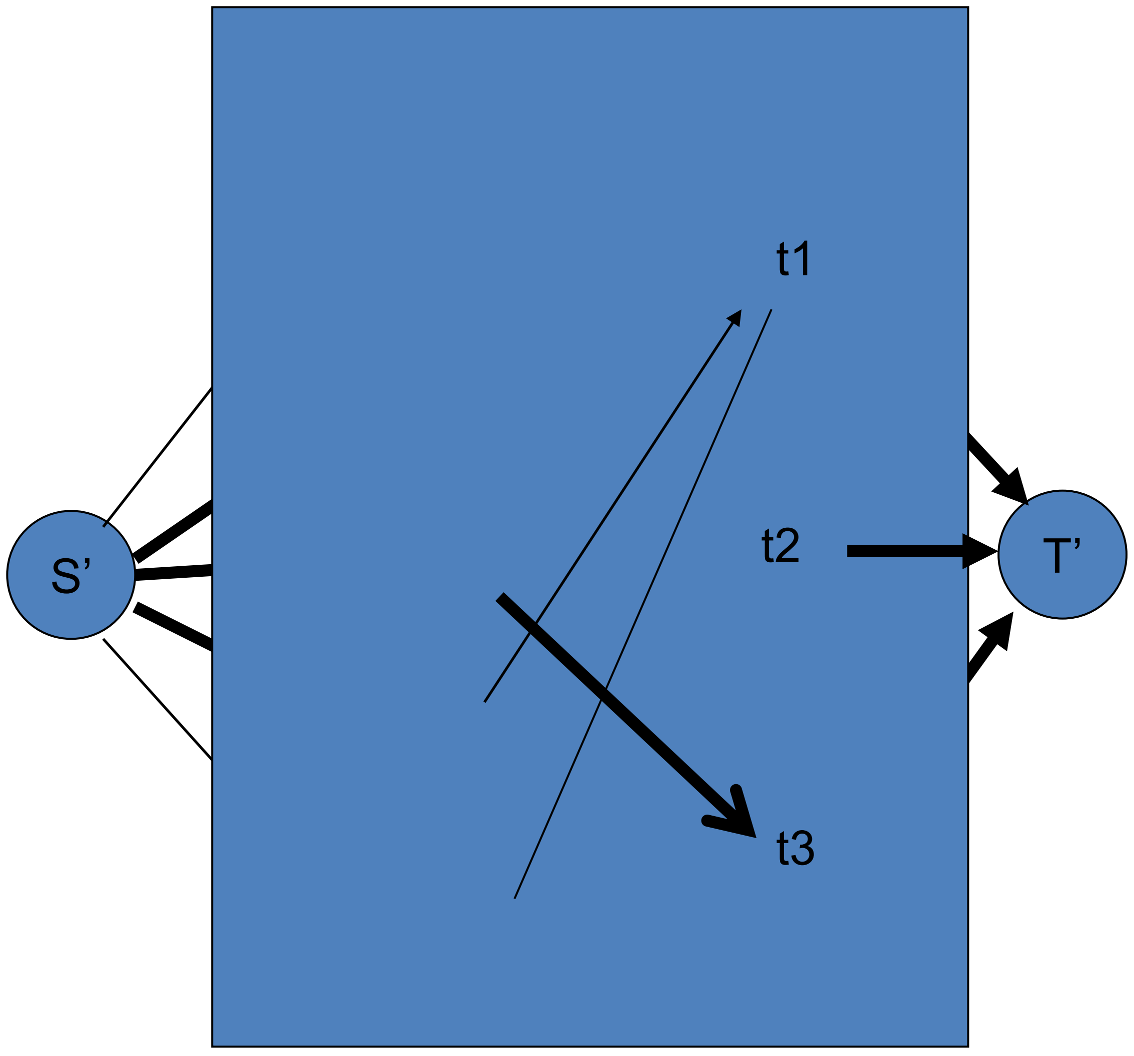
- Agregar dos nodos ficticios s y t, N' = N U {s, t}
- Los arcos de G' son las aristas de A dirigidas de L a R, más las aristas de/a los nodos ficticios.

$$A' = \{(s,u) : u \in L\} U$$
$$\{(u,v) : u \in L, v \in R, (u,v) \in A\} U$$
$$\{(v,t) : v \in R\}$$

- Asignar capacidad de 1 a los arcos de A'.

.

Grafo bipartido



Grafo bipartido

Si M es una asignación ("matching") en G, entonces hay un flujo de valor entero en G' cuyo valor | f | = | M | y viceversa.

La cardinalidad de la asignación máxima en un grafo bipartito G es el valor del flujo máximo en su correspondiente red de flujo G'.