



DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Algoritmos y Estructuras de Datos III

Primer cuatrimestre 2021 (*dictado a distancia*)

Flujo máximo en redes

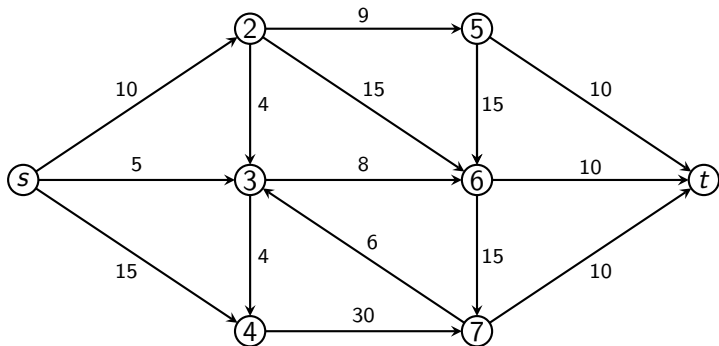
Flujo en redes

- ▶ Las redes se presentan en infinidad de formas en la vida cotidiana: redes eléctricas y telefónicas, redes de transporte, redes de servicios aéreos, redes de distribución de mercadería, redes de computadores, líneas de producción.
- ▶ El objetivo es mover alguna entidad desde un punto a otro de la red de la forma más eficiente posible.
- ▶ Líquido a través de cañería, electricidad en una red eléctrica, personas en una red de rutas, mercadería en una red de abastecimiento, datos en una red de comunicación, partes en una línea de ensamblado

Flujo en redes

- ▶ Una *red* $N = (V, X)$ es un grafo orientado conexo que tiene dos vértices distinguidos, una *fuentes* s , con grado de salida positivo, y un *sumidero* t , con grado de entrada positivo.
- ▶ Una *función de capacidades* en la red es una función $c : X \rightarrow \mathbb{R}^{\geq 0}$.

Flujo en redes



Flujo en redes

- ▶ *Problema de flujo máximo:* Dada una red con capacidades de flujo en los arcos, ¿cómo podemos enviar la máxima cantidad de flujo posible desde un origen a un sumidero determinados respetando las capacidades máximas de los arcos?
- ▶ *Problema de flujo de costo mínimo:* Si, además de la capacidad máxima de flujo que puede atravesar un arco, cada arco tiene asociado un costo por unidad de flujo enviado a través de él, ¿cómo podemos enviar una determinada cantidad de flujo desde un origen a un sumidero determinados respetando las capacidades máximas de los arcos a menor costo?

Flujo en redes

- ▶ En la clase nos vamos a dedicar al problema de flujo máximo.
- ▶ Las entidades que viajan son producidas en una fuente y son consumidas en un sumidero.
- ▶ Los vértices que no son ni la fuente ni el sumidero, no producen ni consumen material.
- ▶ Cada arco en la red tiene una capacidad máxima de producto que puede atravesarlo.

Flujo en redes

- Un *flujo factible* en una red $N = (V, X)$ con función de capacidad c , es una función $f : X \rightarrow \mathbb{R}^{\geq 0}$ que verifica:

1. $0 \leq f(e) \leq c(e)$ para todo arco $e \in X$.
2. Ley de conservación de flujo:

$$\sum_{e \in In(v)} f(e) = \sum_{e \in Out(v)} f(e)$$

para todo vértice $v \in V \setminus \{s, t\}$, donde

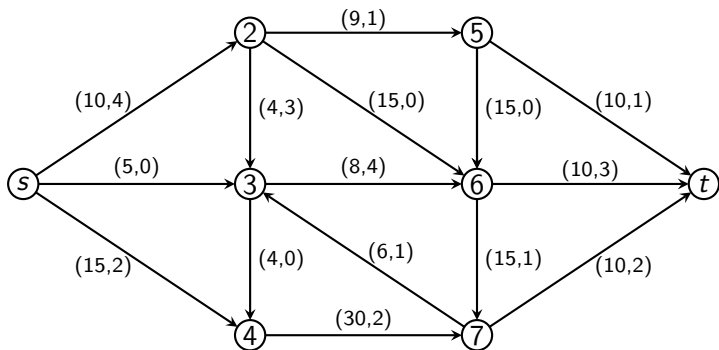
$$In(v) = \{e \in X, e = (w \rightarrow v), w \in V\}$$

$$Out(v) = \{e \in X, e = (v \rightarrow w), w \in V\}$$

- El *valor del flujo* es $F = \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e)$.

Flujo en redes

$$F = 6$$



Flujo en redes

Es indistinto definir el valor del flujo sobre la fuente o sobre el sumidero:

Lema: Dada una red $N = (V, X)$ con función de capacidad c , fuente s y sumidero t :

$$\sum_{e \in Out(s)} f(e) - \sum_{e \in In(s)} f(e) = F$$

Problema de flujo máximo

Problema: El *problema de flujo máximo* consiste en, dada una red $N = (V, X)$ con capacidades en los arcos y dos vértices específicos, una fuente s , y un sumidero t , determinar el flujo de valor máximo F que se puede enviar de s a t .

- ▶ Aplicaciones directas: redes de envío de combustibles, autos en una red de rutas, mensajes en una red de computadoras, electricidad en una red eléctrica.
- ▶ Otras aplicaciones: *machine scheduling*, asignación de módulos de un programa a procesadores, conectividad en grafos.
- ▶ Como subproblema de problemas en redes más complicados, como en el caso del problema de flujo de costo mínimo.

Problema de corte mínimo

Un **corte** en la red $N = (V, X)$ es un subconjunto $S \subseteq V \setminus \{t\}$, tal que $s \in S$.

Notación: Dados $S, T \subseteq V$, llamaremos

$$ST = \{(u \rightarrow v) \in X : u \in S \text{ y } v \in T\}.$$

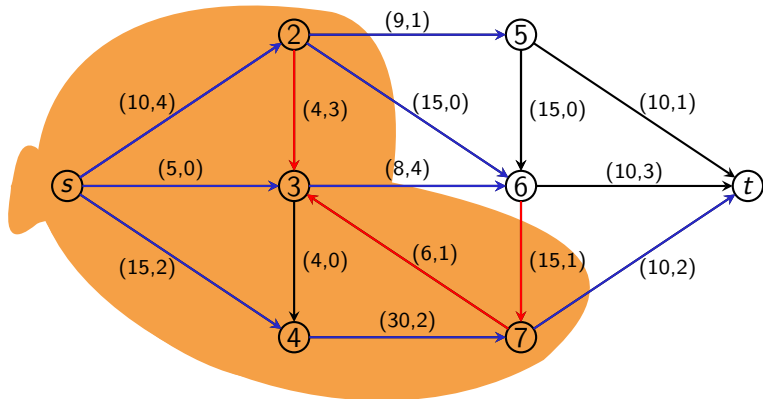
Proposición: Sea f un flujo definido en una red N y sea S un corte, entonces

$$F = \sum_{e \in S\bar{S}} f(e) - \sum_{e \in \bar{S}S} f(e)$$

donde $\bar{S} = V \setminus S$.

Problema de corte mínimo

$$F = 6$$



Problema de corte mínimo

Dada una red $N = (V, X)$ con función de capacidad c , la capacidad de un corte S se define como

$$c(S) = \sum_{e \in S\bar{S}} c(e).$$

El *problema de corte mínimo* consiste en, dada una red $N = (V, X)$ con función de capacidades en los arcos c , determinar un corte capacidad mínima. Es decir, encontrar S corte de N tal que:

$$c(S) = \min\{c(\bar{S}) | \bar{S} \text{ es un corte de } N\}.$$

Problema de flujo máximo - corte mínimo

Los siguientes resultados relacionan ambos problemas de una forma dual.

Lema: Dados una red $N = (V, X)$ con función de capacidad c , una función de flujo con valor F y un corte S , se cumple que:

$$F \leq c(S).$$

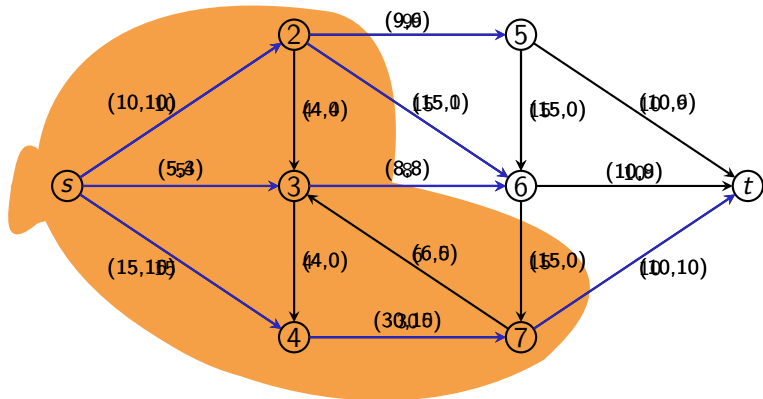
Corolario (certificado de optimalidad): Si F es el valor de un flujo f y S un corte en una red N tal que $F = c(S)$ entonces:

- ▶ f define un flujo máximo y
- ▶ S es un corte de capacidad mínima.

Problema de flujo máximo - corte mínimo

$$c(S) = 00$$

$$F = 28$$

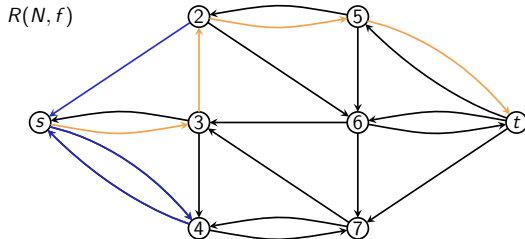
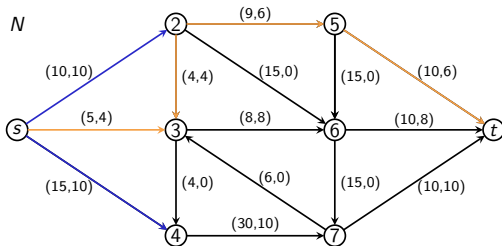


Camino de aumento

Dada una red $N = (V, X)$ con función de capacidad c y un flujo factible f :

- ▶ Definimos la *red residual*, $R(N, f) = (V, X_R)$ donde $\forall (v \rightarrow w) \in X$,
 - ▶ $(v \rightarrow w) \in X_R$ si $f((v \rightarrow w)) < c((v \rightarrow w))$
 - ▶ $(w \rightarrow v) \in X_R$ si $f((v \rightarrow w)) > 0$.
- ▶ Un *camino de aumento* es un camino orientado de s a t en $R(N, f)$.

Camino de aumento



Algoritmo de camino de aumento

caminoAumento(N, f, R)

entrada: $N = (V, X)$, flujo f , red residual $R(N, f) = (V, X_R)$

salida: P camino de aumento o S corte (que sera minimo)

$S \leftarrow \{s\}$

mientras $t \notin S$ **y** $\exists (v \rightarrow w) \in X_R$ **y** $v \in S$ **y** $w \notin S$ **hacer**

$ant[w] \leftarrow v$

$S \leftarrow S \cup \{w\}$

fin mientras

si $t \notin S$ **entonces**

retornar S corte de N

si no

 reconstruir P entre s y t usando ant a partir de t

retornar P camino de aumento

fin si

Algoritmo de camino de aumento

Proposición: Dada una red N , un flujo f y su residual $R(N, f) = (V, X_R)$, el algoritmo de camino de aumento determina un camino de aumento, P , si existe y, en caso contrario, determina un corte S de N .

El algoritmo de camino de aumento no dice en que orden deben incorporarse los vértices a S .

Flujo en redes - Camino de aumento

Dada una red $N = (V, X)$ con función de capacidad c , un flujo factible f y un camino de aumento P en $R(N, f)$:

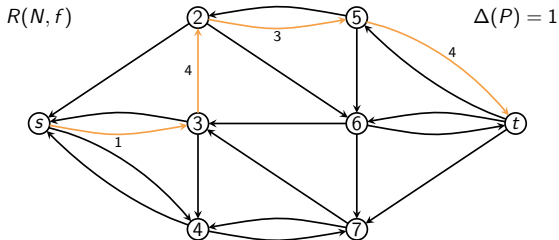
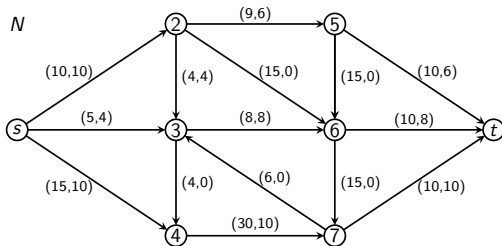
- Para cada arco $(v \rightarrow w)$ de P , definimos

$$\Delta((v \rightarrow w)) = \begin{cases} c((v \rightarrow w)) - f((v \rightarrow w)) & \text{si } (v \rightarrow w) \in X \\ f((w \rightarrow v)) & \text{si } (w \rightarrow v) \in X \end{cases}$$

- Y

$$\Delta(P) = \min_{e \in P} \{\Delta(e)\}$$

Camino de Aumento



Flujo en redes

Proposición: Sea f un flujo definido sobre una red N con valor F y sea P un camino de aumento en $R(N, f)$. Entonces el flujo \bar{f} , definido por

$$\bar{f}((v \rightarrow w)) = \begin{cases} f((v \rightarrow w)) & \text{si } (v \rightarrow w) \notin P \\ f((v \rightarrow w)) + \Delta(P) & \text{si } (v \rightarrow w) \in P \\ f((v \rightarrow w)) - \Delta(P) & \text{si } (w \rightarrow v) \in P \end{cases}$$

es un flujo factible sobre N con valor $\bar{F} = F + \Delta(P)$.

Teorema: Sea f un flujo definido sobre una red N . Entonces f es un flujo máximo \iff no existe camino de aumento en $R(N, f)$.

Teorema: Dada una red N , el valor del flujo máximo es igual a la capacidad del corte mínimo.

Algoritmo de Ford y Fulkerson, F&F

Ford&Fulkerson(N)

entrada: $N = (V, X)$ con función de capacidad c

salida: f flujo máximo

definir un flujo inicial en N

(por ejemplo $f(e) \leftarrow 0$ para todo $e \in X$)

mientras exista P camino de aumento en $R(N, f)$ **hacer**

para cada arco $(v \rightarrow w)$ de P **hacer**

si $(v \rightarrow w) \in X$ **entonces**

$f((v \rightarrow w)) \leftarrow f((v \rightarrow w)) + \Delta(P)$

si no $((w \rightarrow v) \in X)$

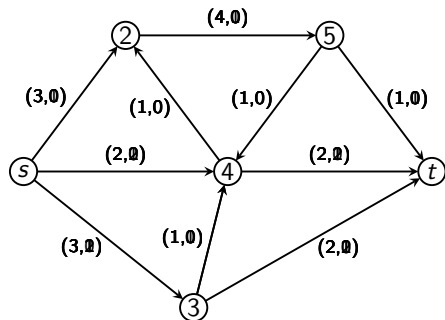
$f((w \rightarrow v)) \leftarrow f((w \rightarrow v)) - \Delta(P)$

fin si

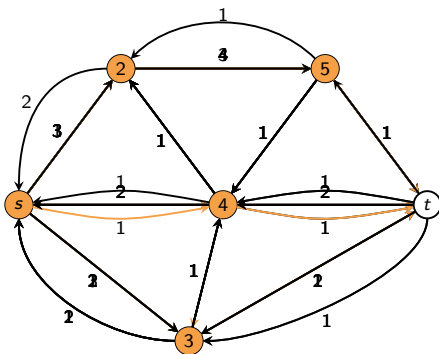
fin para

fin mientras

Algoritmo de Ford y Fulkerson, F&F



$R(N, f)$

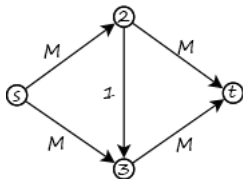


Algoritmo de Ford y Fulkerson, F&F

Proposición: Si las capacidades de los arcos de la red son enteras, el problema de flujo máximo tiene un flujo máximo entero.

Proposición: Si los valores del flujo inicial y las capacidades de los arcos de la red son enteros, F&F realiza a lo sumo F iteraciones, siendo F el valor del flujo máximo.

En esta red el flujo máximo es $F = 2M$. Si los caminos de aumento P de todas las iteraciones incluyen el arco $(v_2 \rightarrow v_3)$, $\Delta(P) = 1$ en toda iteración, debiendo hacer $F = 2M$ iteraciones.



Algoritmo de Ford y Fulkerson, F&F

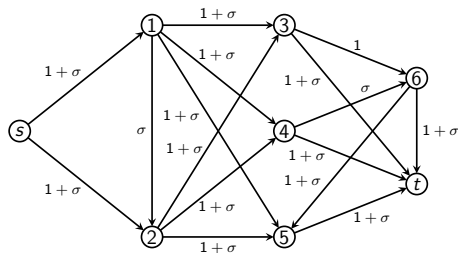
Proposición: Si los valores del flujo inicial y las capacidades de los arcos de la red son enteros, F&F es $\mathcal{O}(nmU)$, donde U es una cota superior para el valor de las capacidades.

Si no se especifica el orden en el que se eligen los arcos y vértices a marcar en el algoritmo de camino de aumento, el número de iteraciones de F&F, aún con capacidades enteras, puede ser no polinomial respecto del tamaño del problema.

Si las capacidades o el flujo inicial son números irracionales, F&F puede no parar (realizar un número infinito de pasos).

Algoritmo de Ford y Fulkerson, F&F

$$\sigma = (\sqrt{5} - 1)/2$$



Iteración	Camino de aumento
$6k + 1$	$s, 1, 2, 3, 6, t$
$6k + 2$	$s, 2, 1, 3, 6, 5, t$
$6k + 3$	$s, 1, 2, 4, 6, t$
$6k + 4$	$s, 2, 1, 4, 6, 3, t$
$6k + 5$	$s, 1, 2, 5, 6, t$
$6k + 6$	$s, 2, 1, 5, 6, 4, t$

Implementación de Edmonds y Karp

- ▶ Usa BFS en el algoritmo de camino de aumento para marcar vértices.
- ▶ E&K realiza a lo sumo mn iteraciones.
- ▶ La complejidad del algoritmo es $\mathcal{O}(m^2n)$.
- ▶ Hay otros algoritmos más eficientes (más complicados).

Implementación de Edmonds y Karp

Teorema: Dada una red $N = (V, X)$ con n vértices y m arcos, el algoritmo de E&K realiza, a lo sumo, nm iteraciones.

Lema: Sea f_i el flujo de la iteración i de algoritmo de E&K. Entonces, para todo $v \in V$, $d_{R(N, f_i)}(s, v) \leq d_{R(N, f_{i+1})}(s, v)$ para toda iteración i del algoritmo.

Variantes del problema de flujo máximo

- ▶ Múltiples fuentes y sumideros
- ▶ Matching máximo en grafos bipartitos
- ▶ Capacidades en los vértices
- ▶ Flujo en redes no dirigidas
- ▶ Flujo máximo con costo mínimo o acotado