Flujo en Redes

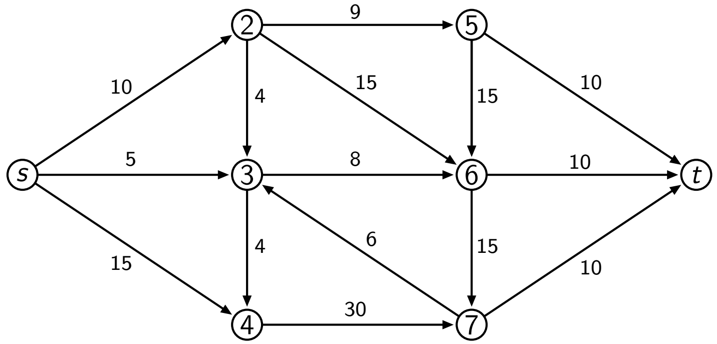
Algoritmos y Estructuras de Datos III

Flujo en Redes

Definiciones:

- Una red $N = (V, X)$ es un grafo orientado conexo que tiene dos nodos distinguidos una fuente 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 $c : X \to \mathbb{R}^{\geq 0}$.

Flujo en Redes



Flujo en Redes

Definiciones:

- Un **flujo factible** en una red $N = (V, X)$ con función de capacidad c, es una función $f : X \to \mathbb{R}^{\geq 0}$ que verifica:
  1. $0 \leq f(e) \leq c(e)$ para todo arco $e \in X$.
  2. Ley de conservación de flujo:

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

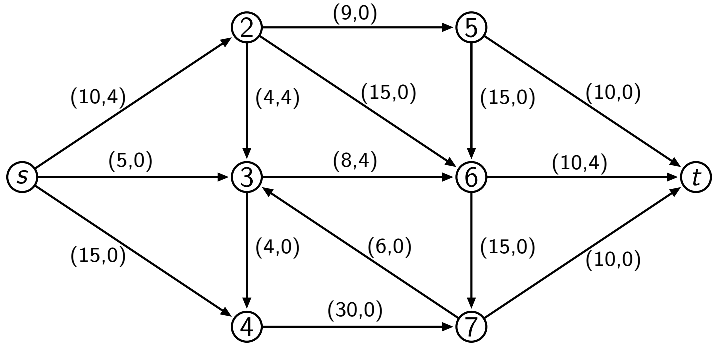
  para todo nodo $v \in V \setminus \{s, t\}$, donde

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

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

Flujo en Redes

$F = 4$



Flujo en Redes

Problema: Determinar el flujo de valor máximo F que se puede definir en una red $N = (V, X)$.

Definiciones:

- Un **corte** en la red $N = (V, X)$ es un subconjunto $S \subseteq V \setminus \{t\}$, tal que $s \in S$.
- Dados $S, T \subseteq V$, $ST = \{(u \to v) \in X : u \in S \text{ y } v \in T\}$

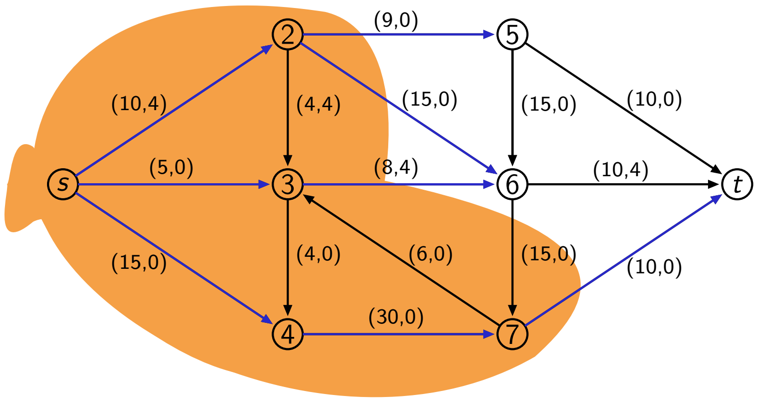
Proposición: Sea f un flujo definido en una red $N = (V, X)$ y sea S un corte, entonces

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

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

Flujo en Redes

$F = 4$



Flujo en Redes

Definición: La capacidad de un corte S se define como

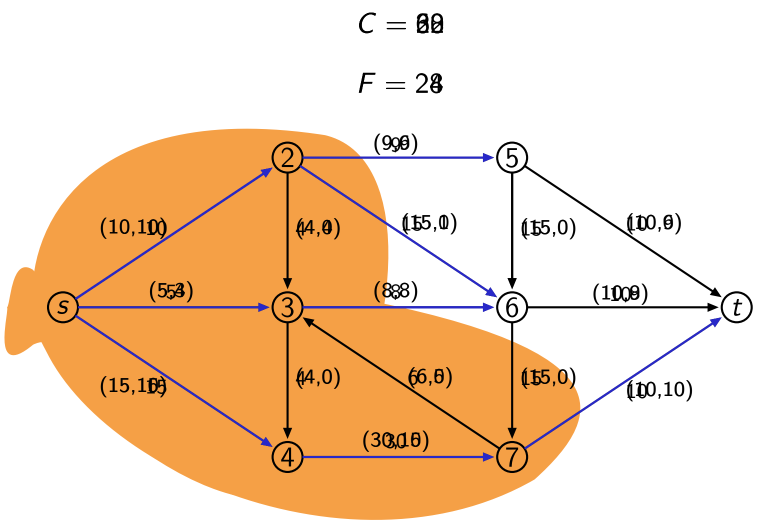
$$c(S) = \sum_{e \in S\bar{S}} c(e).$$

Lema: Si f es una función de flujo con valor F y S es un corte en N, entonces

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

Corolario (certificado de optimalidad): Si F es el valor de un flujo f y S un corte en N tal que $F = c(S)$ entonces f define un flujo máximo y S un corte de capacidad mínima.

Flujo en Redes



Flujo en Redes - Camino de Aumento

Definiciones: 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

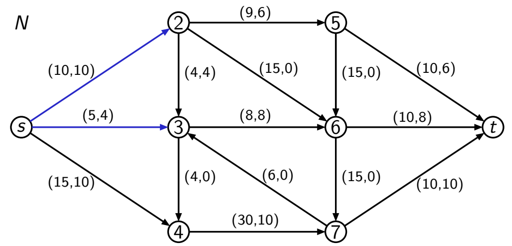
$$\forall (v \to w) \in X,$$

  - $(v \to w) \in X_R$ si $f((v \to w)) < c((v \to w))$
  - $(w \to v) \in X_R$ si $f((v \to w)) > 0.$

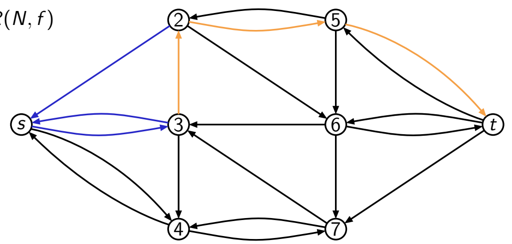
- Un **camino de aumento** es un camino orientado P de s a t en $R(N, f)$.

Flujo en Redes - Camino de Aumento

N



$R(N, f)$



Flujo en Redes - Camino de Aumento

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

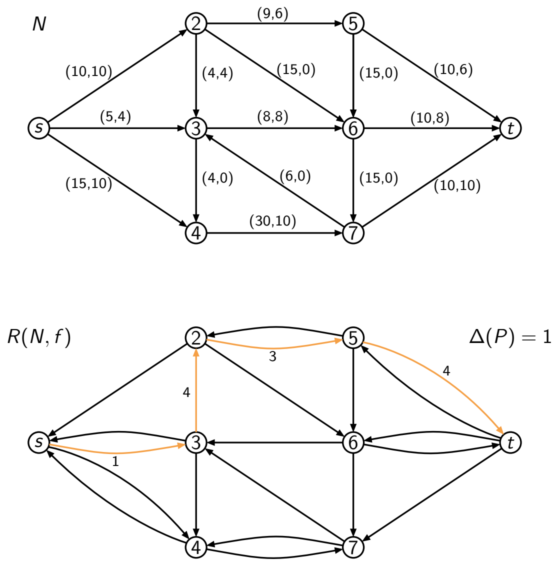
- Para cada arco $(v \to w)$ en el camino de aumento P, definimos

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

- Y

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

Flujo en Redes - Camino de Aumento



Flujo en Redes - Algoritmo de camino de aumento

Entrada: Dada una red N con función de flujo f, la red residual $R(N,f) = (V, X_R)$.

$S := \{s\}$

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

  $ant(w) := v$

  $S := S \cup \{w\}$

fin mientras

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

  `retornar` S `corte de` V

si no

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

  `retornar` P `camino de aumento`

fin si

Flujo en Redes - Algoritmo de camino de aumento

Proposición: El algoritmo de camino de aumento determina un camino de aumento si existe, y si no llega a incorporar a t en S es porque no hay camino de aumento.

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

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

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

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 $\Longleftrightarrow$ no existe camino de aumento en $R(N,f)$.

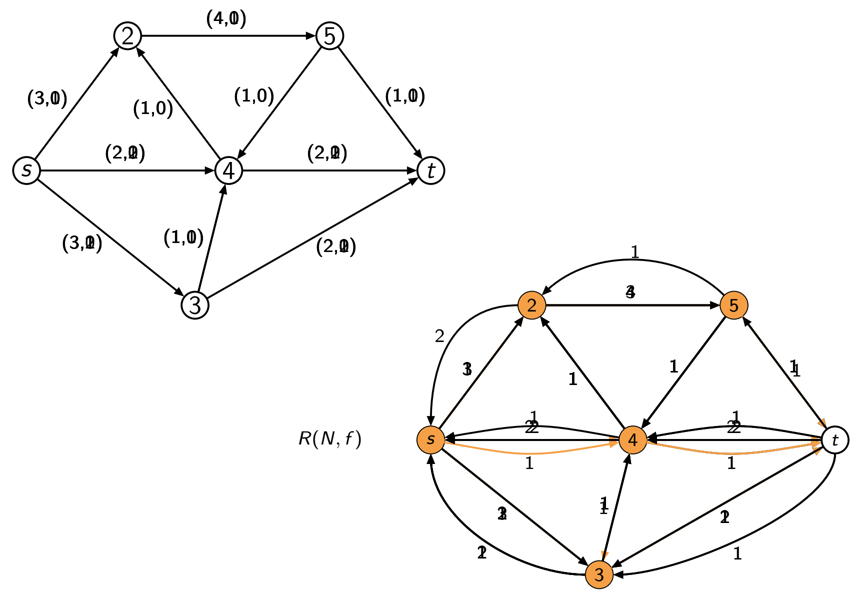
Teorema: Dada una red N, el valor del flujo máximo es igual a la capacidad del corte mínimo.

Flujo en Redes - Algoritmo de Ford y Fulkerson

Entrada: Red $N = (X, V)$ con función de capacidad $c : X \to \mathbb{R}^+$.

`definir un flujo inicial en` N
(por ejemplo $f(e) := 0$ para todo $e \in X$)
mientras `exista` $P :=$ `camino de aumento en` $R(N, f)$ **hacer**
  para `cada arco` $(v \to w)$ `de` P **hacer**
    si $(v \to w) \in X$ **entonces**
      $f((v \to w)) := f((v \to w)) + \Delta(P)$
    si no $((w \to v) \in X)$
      $f((w \to v)) := f((w \to v)) - \Delta(P)$
    fin si
  fin para
fin mientras

Flujo en Redes - Algoritmo de Ford y Fulkerson



$R(N, f)$

Flujo en Redes - Algoritmo de Ford y Fulkerson

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

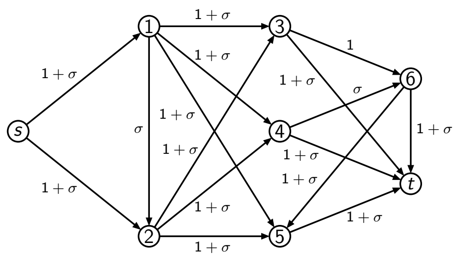
Teorema: Si los valores del flujo inicial y las capacidades de los arcos de la red son enteras el método de Ford y Fulkerson realiza a lo sumo nU iteraciones, siendo entonces $\mathcal{O}(nmU)$, donde U es una cota superior finita para el valor de las capacidades.

Si las capacidades o el flujo inicial son números irracionales, el método de Ford y Fulkerson puede no parar (realizar un número infinito de pasos).

Si no se especifica el orden en el que se eligen los arcos y nodos a marcar en el algoritmo de camino de aumento, el número de iteraciones puede ser no polinomial respecto del tamaño del problema.

Flujo en Redes - Algoritmo de Ford y Fulkerson

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



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

Flujo en Redes - Modificación de Edmonds y Karp

- Usa BFS en el algoritmo de camino de aumento para marcar nodos.
- La complejidad del algoritmo es $\mathcal{O}(m^2 n)$.
- Hay otros algoritmos más eficientes (más complicados).