Tecnológico de Monterrey.

TC2038. Análisis y diseño de Algoritmos A

M.C. Ramona Fuentes Valdéz

rfuentes@tec.mx

127

Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Aplicaciones

- Redes

Redes	Nodos	Arcos	Flujo *(flow)*
Comunicación	Teléfono, Computadoras, Satélites	Cables, fibra óptica, *microwave relays*	Voz, video, *packets*
Circuitos	Compuertas, Registros, Procesadores	Cables (*wires*)	Corriente (*current*)
Mecánicas	Uniones (*joints*)	Varillas, vigas, resortes	Calor, energía
Hidraúlicas	Estaciones de bombeo, lagos	Oleoductos (pipelines)	Fuido, aceite
Financieras	Acciones, empresas	Transacciones	Dinero
Transporte	Aeropuertos, Ferrocarriles, intersecciones de calles	Carreteras, vías férreas, rutas aéreas	Mercancías, vehículos, pasajeros

128

Grafos

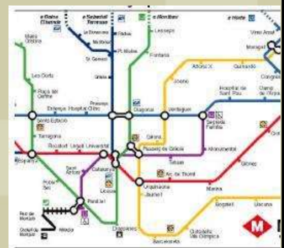
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Grafo

- Un escenario común es usar un grafo para representar una "red de flujo" y usarlo para responder preguntas sobre los flujos de materiales.
- El flujo es la velocidad a la que el material se mueve a través de la red.
- Cada arista dirigida es un conducto para el material con cierta capacidad establecida.
- Los vértices son puntos de conexión pero no recogen material.
  - El flujo hacia un vértice debe ser igual al flujo que sale del vértice, conservación del flujo.

Conceptos

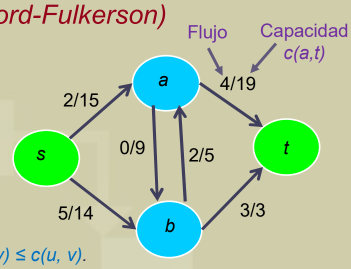
- Origen.- vértice s
  - donde se produce el material.
- Destino.- vértice t
  - donde se consume material.
- Para todos los demás vértices - lo que entra debe salir
  - Conservación de flujo.
- **Objetivo:** determinar la tasa máxima de flujo de material desde la fuente hasta el destino.



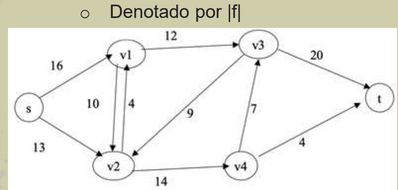
Grafos

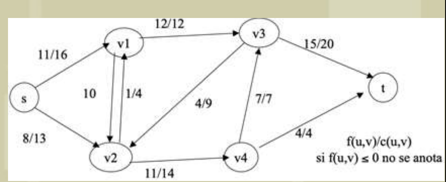
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Problema formal de flujo máximo



- Grafo *G = (V, E)* - una **red de flujo**
  - Dirigido, cada arista tiene capacidad *c(u, v) ≥ 0*.
  - Dos vértices especiales: *fuente s* y *destino t*.
  - Para cualquier otro vértice *v*, hay un camino *s →…→ v →…→ t*.
- **Flujo** - una función *f: V x V → **R***
  - Restricción de capacidad: Para todo *u, v є V*: *f(u, v) ≤ c(u, v)*.
  - Simetría oblicua: Para todo *u, v є V*: *f(u, v) = –f(v, u)*.
  - Conservación del flujo: Para todo *u є V - {s, t}*: $\sum_{v \in V} f(u,v) = f(u,V) = 0$, or $\sum_{v \in V} f(v,u) = f(V,u) = 0$
- **Flujo máximo**
  - Se quiere encontrar un flujo de valor máximo desde la fuente hasta el destino.
    - Denotado por |f|





Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Método de Ford-Fulkerson

- Incrementa iterativamente el valor del flujo.
- Comienza con $f(u, v) = 0$ para todas las aristas, dando un flujo de 0.
- En cada iteración, incrementa el valor de flujo en G encontrando un “camino de aumento" en un "grafo residual" asociado.
- Aumenta repetidamente el flujo hasta que la red residual no tenga más caminos de aumento.

- Contiene varios algoritmos:
  - ✓ Grafo/red residual
  - ✓ Camino de aumento
    - Encontrar un camino p de s a t (camino de aumento), tal que hay algún valor $x > 0$, y para cada arista (u, v) en p se puedan agregar x unidades de flujo:

$$f(u, v) + x \leq c(u, v)$$

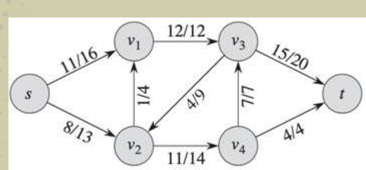
```
FORD-FULKERSON-METHOD(G, s, t)
initialize flow f to 0
while there exists an augmenting path p
    do augment flow f along p
return f
```


Grafos

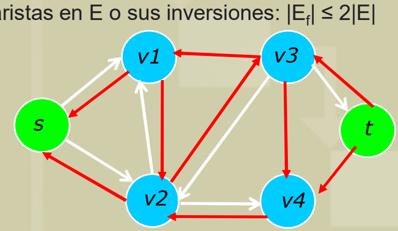
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Red residual

- Para encontrar una ruta de aumento, se puede encontrar cualquier ruta en la red residual:
  - Capacidades residuales: $c_f(u, v) = c(u, v) - f(u, v)$
    - es decir, la capacidad real menos el flujo de u a v.
    - El flujo puede ser negativo.
  - Red residual: $G_f = (V, E_f)$, donde: $E_f = \{(u, v) \in V \times V: c_f(u, v) > 0\}$
  - Nota: las aristas en E_f pueden ser aristas en E o sus inversiones: $|E_f| \leq 2|E|$



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



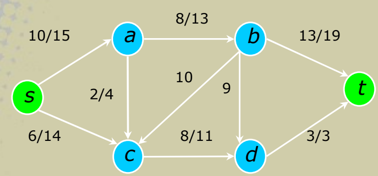
Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Grafo residual

Ejemplo

- Calcula el grafo residual del grafo con el siguiente flujo:



Capacidad residual y Camino en aumento

- Encontrar un camino de aumento
  - Encuentra una ruta de *s* a *t* en el grafo residual
  - La *capacidad residual* de un camino *p* en G_f: $c_f(p)$ = min{$c_f(u, v)$: (u, v) está en *p*}
    - es decir, encuentra la capacidad mínima a lo largo de *p*
  - Para hacer aumentos: para toda *(u, v)* en *p*, se suma este $c_f(p)$ a *f(u, v)* (y se resta de *f(v, u)*)
  - El flujo resultante es un flujo válido con un valor mayor.

Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

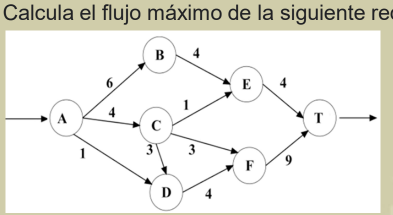
Grafo residual

Capacidad residual y Camino en aumento

- Encontrar un camino de aumento
  - Encuentra una ruta de *s* a *t* en el grafo residual
  - La *capacidad residual* de un camino *p* en G_f: $c_f(p)$ = min{$c_f(u, v)$: (u, v) está en *p*}
    - es decir, encuentra la capacidad mínima a lo largo de *p*
  - Para hacer aumentos: para toda *(u, v)* en *p*, se suma este $c_f(p)$ a *f(u, v)* (y se resta de *f(v, u)*)
  - El flujo resultante es un flujo válido con un valor mayor.

Ejemplo

- Calcula el flujo máximo de la siguiente red:



Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

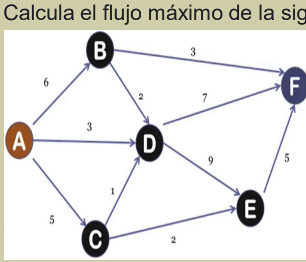
Grafo residual

Capacidad residual y Camino en aumento

- Encontrar un camino de aumento
  - Encuentra una ruta de *s* a *t* en el grafo residual
  - La *capacidad residual* de un camino *p* en G_f: $c_f(p) = \min\{c_f(u, v): (u, v) \text{ está en } p\}$
    - es decir, encuentra la capacidad mínima a lo largo de *p*
  - Para hacer aumentos: para toda *(u, v)* en *p*, se suma este $c_f(p)$ a $f(u, v)$ (y se resta de $f(v, u)$)
  - El flujo resultante es un flujo válido con un valor mayor.

Ejemplo

- Calcula el flujo máximo de la siguiente red:



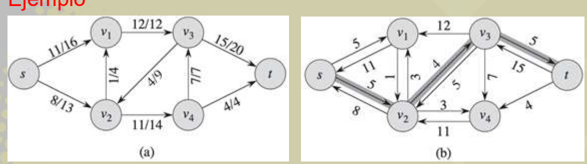
135

Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

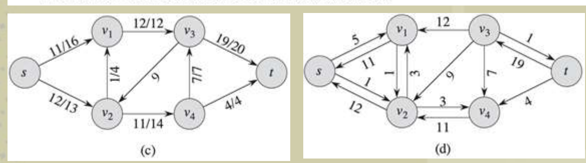
Red residual y camino en aumento

Ejemplo



(a) The flow network G and flow f of Figure 26.1(b).

(b) The residual network G_f with augmenting path p shaded; its residual capacity is $c_f(p) = c_f(v_2, v_3) = 4$.
Edges with residual capacity equal to 0, such as (v_1, v_3), are not shown, a convention we follow in the remainder of this section.



(c) The flow in G that results from augmenting along path p by its residual capacity 4. Edges carrying no flow, such as (v_3, v_2), are labeled only by their capacity, another convention we follow throughout.

(d) The residual network induced by the flow in (c).

136

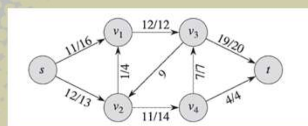
Grafos

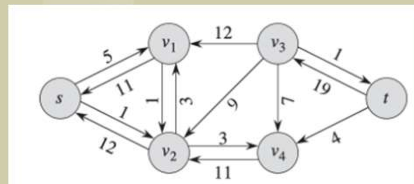
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Red residual y camino en aumento

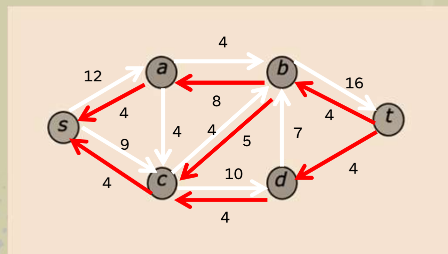
Ejemplo

- Obtener la red residual de la red de flujo:





- Obtener un camino en aumento de la red residual resultante e indica el valor que se agregará al flujo.



s→a→b→t. min{12,4,16} = 4

s→a→c→b→t. min{12,4,4,16} = 4

s→c→b→t. min{9,4,16} = 4

s→c→d→b→t. min{9,10,7,16} = 7

Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Algoritmo

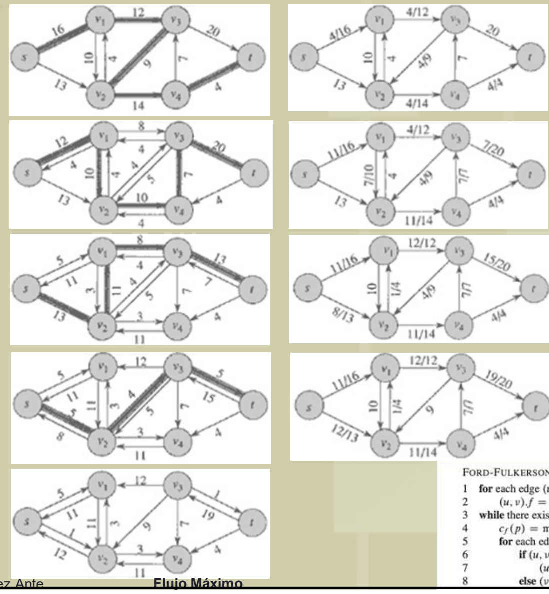
```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ G.E
    (u, v).f = 0
while there exists a path p from s to t in the residual network G_f
    c_f(p) = min {c_f(u, v) : (u, v) is in p}
    for each edge (u, v) in p
        if (u, v) ∈ E
            (u, v).f = (u, v).f + c_f(p)
        else (v, u).f = (v, u).f − c_f(p)
```

Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Ejemplo:

Grafo residual | Flujo en aumento



```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ G.E
    (u, v).f = 0
while there exists a path p from s to t in the residual network G_f
    c_f(p) = min{c_f(u, v) : (u, v) is in p}
    for each edge (u, v) in p
        if (u, v) ∈ E
            (u, v).f = (u, v).f + c_f(p)
        else (v, u).f = (v, u).f − c_f(p)
```


Grafos

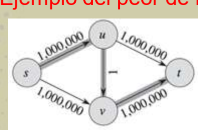
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Tiempo de ejecución

En el peor de los casos…

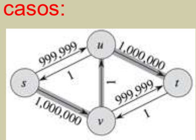
- Suponiendo un flujo de enteros.
- Cada aumento incrementa el valor del flujo en una cantidad positiva.
- El aumento se puede realizar en O(E).
- El tiempo total de ejecución en el peor de los casos O(E|f*|), donde f* es el flujo máximo encontrado por el algoritmo.

Ejemplo del peor de los casos:



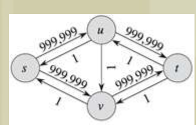
(a)

Camino aumentado en 1



(b)

Red Residual Resultante



(c)

Red Residual Resultante

Grafos

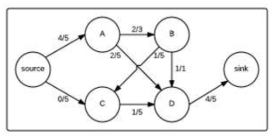
Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Edmonds-Karp

- El algoritmo de Edmonds-Karp es una implementación específica del algoritmo de Ford-Fulkerson.
- Toma el camino más corto (*en términos de número de aristas*) como un camino de aumento.
- Se especifica que la búsqueda primero en amplitud debe usarse para encontrar las rutas más cortas durante las etapas intermedias del programa.
- El tiempo de ejecución $O(VE^2)$, porque el número de aumentos es $O(VE)$.
- Esta mejora es importante porque hace que el tiempo de ejecución de Edmonds-Karp sea independiente del flujo máximo de la red, f*.

In the Edmonds-Karp algorithm, the set of augmenting paths to choose from is well defined. Which of the following options will be the next augmenting path chosen by Edmonds-Karp?

The following graph shows a set of vertices and edges. Each edge shows two numbers: its current flow divided by its capacity. In this implementation, vertices are processed in alphabetical order for search.



Graph in the middle of Edmonds-Karp

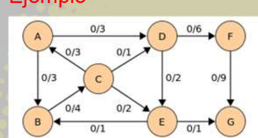
- ◯ {source, A, D, sink}
- ◯ {source, C, B, D, sink}
- ◯ {source, C, D, sink}
- ◯ {source, A, B, C, D, sink}

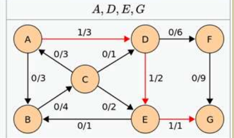
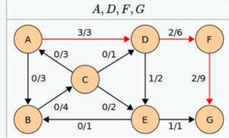
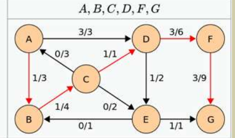
Grafos

Flujo máximo en un grafo *(Max Flow: Ford-Fulkerson)*

Edmonds-Karp

Ejemplo



Capacidad	Camino Network resultante
$\min(c_f(A,D), c_f(D,E), c_f(E,G)) =$ $\min(3-0, 2-0, 1-0) =$ $\min(3,2,1) = 1$	A, D, E, G 
$\min(c_f(A,D), c_f(D,F), c_f(F,G)) =$ $\min(3-1, 6-0, 9-0) =$ $\min(2,6,9) = 2$	A, D, F, G 
$\min(c_f(A,B), c_f(B,C), c_f(C,D), c_f(D,F), c_f(F,G)) =$ $\min(3-0, 4-0, 1-0, 6-2, 9-2) =$ $\min(3,4,1,4,7) = 1$	A, B, C, D, F, G 
$\min(c_f(A,B), c_f(B,C), c_f(C,E), c_f(E,D), c_f(D,F), c_f(F,G)) =$ $\min(3-1, 4-1, 2-0, 0--1, 6-3, 9-3) =$ $\min(2,3,2,1,3,6) = 1$	A, B, C, E, D, F, G