Redes de Flujo: Acelerando Ford-Fulkerson

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski

vpodberezski@fi.uba.ar

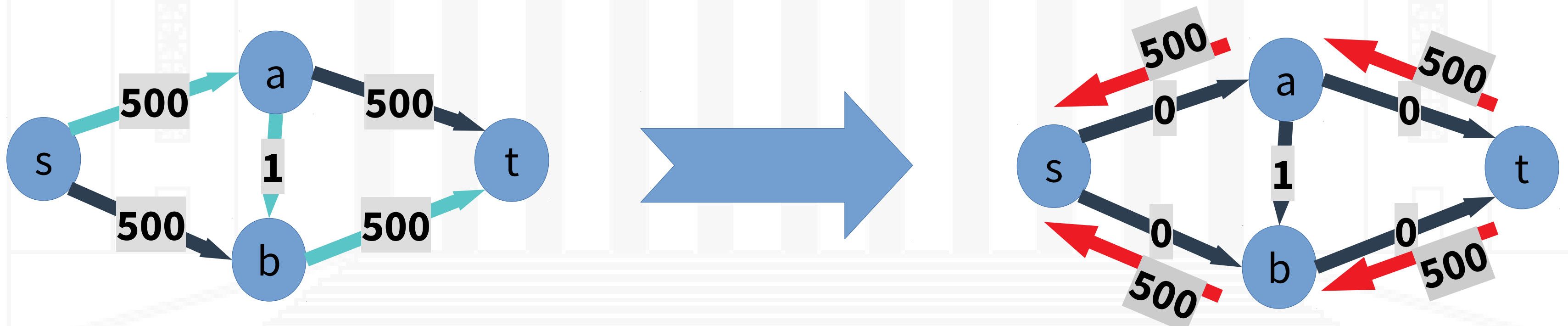
Un simple ejemplo ...

Sea G un grafo.

Podemos resolver el problema del flujo máximo

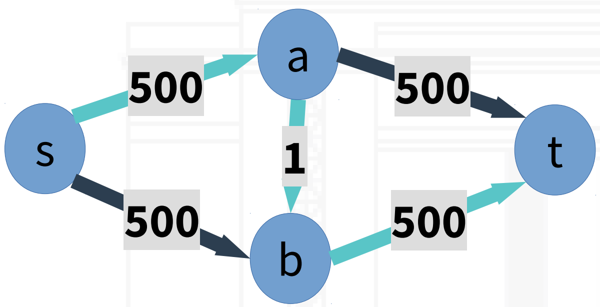
Vemos que el flujo máximo es de 1000

(se puede resolver con 2 caminos de aumento)

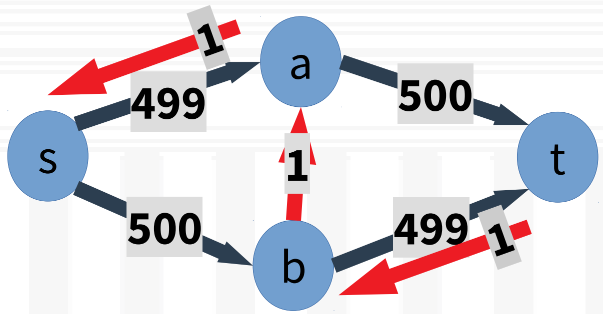


… que puede derivar en patológico

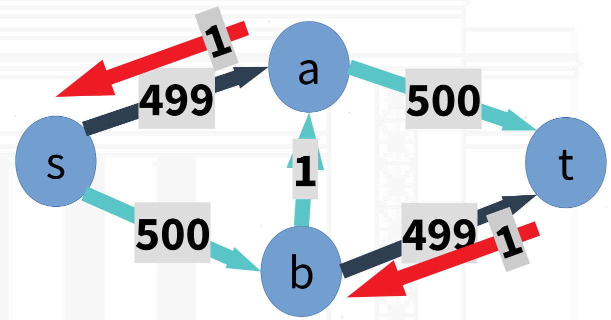
La selección de caminos s-t malos puede relentizar mucho la finalización del algoritmo



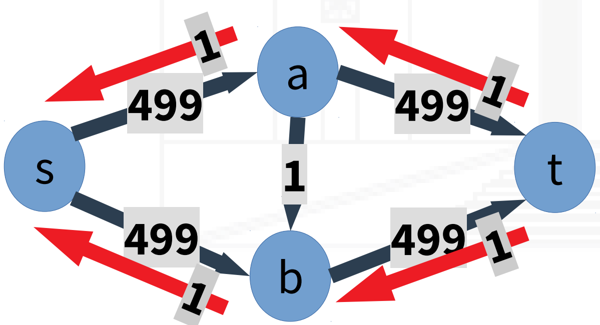
Grafo Gf, Camino = s,a,b,t



Grafo Gf



Grafo Gf, Camino = s,b,a,t



Grafo Gf

Y repetimos hasta que
no haya caminos s-t
X 1000

Selección eficiente de camino s-t

Intentar elegir aquellos caminos con mayor cuellos de botella

- Idealmente seleccionar de mayor cuello de botella para cada paso.
- Mantener esta información actualizada puede ser costoso.

Existen varios métodos para intentar realizar selecciones convenientes.

Método: Parámetro de escalado Δ

Se utilizará un parámetro Δ

solo se buscarán aquellos P caminos s-t con bottleneck(P,f)> Δ

Inicialmente Δ tomara

la potencia de 2 mas grande que sea no mayor a a la suma capacidad de los ejes que salen de s.

Repetir hasta hallar el flujo máximo

Armar grafo residual unicamente con los ejes que superen Δ.

Buscar caminos s-t en ese subgrafo y proceder igual que en Ford-Fulkerson

De no existir más caminos se reduce Δ a la mitad.

Pseudocódigo

```
Scaling Max-Flow

    Inicialmente f(e)=0 para todo e en G
    Definicar delta como la mas grande potencia de 2 que no sea mayor a la capacidad
maxima que sale de s

    Mientras delta ≥ 1
        Mientras exista un camino s-t en el grado Gf(delta)
            Sea P camno simple eh Gf(delta)

            f’ = augment(f,P)
            Actualizar f para ser f’ y actualizar Gf(delta)

        delta = delta / 2

    retornar f
```

Análisis del algoritmo

Al ser una variante de Ford-Fulkerson

se pueden utilizar varias de las demostraciones que vimos antes.

Si las capacidades son enteras, mediante el escalado

el flujo y las capacidades residuales siguen siendo enteras.

Cuando $\Delta=1$, $G_f(\Delta)=G_f$,

por lo que cuando el algoritmo termina el flujo es máximo

El numero de iteraciones definido por el parámetro Δ

esta limitado por la $C=\sum c_{out}$ de los ejes que salen de s. : $1+\lfloor \log_2 C \rfloor$

En cada iteración hay a lo sumo $2|E|$ caminos de aumento

El tiempo de ejecución es $O(|E|^2 \log_2 C)$

Si C es grande es mejor que $O(|E|C)$

Presentación realizada en Mayo de 2020