

Ford-Fulkerson

Daniel Penazzi

14 de abril de 2021

Tabla de Contenidos

Caminos aumentantes

- Idea y motivación.

- Definición y discusión

Algoritmo de Ford-Fulkerson

- Algoritmo de FF y comparación con Greedy

- Primera propiedad que necesitamos probar

Max Flow Min Cut Theorem

- Enunciado del teorema

- Prueba parte A

- Prueba Partes B y C

Consecuencias

- Correctitud del algoritmo de Ford-Fulkerson

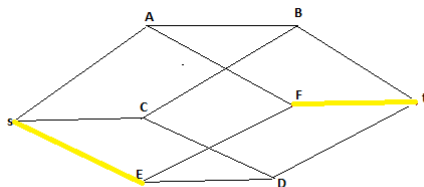
- Teorema de la Integralidad

Repaso

- Al final de la clase anterior habíamos dicho que se podía modificar Greedy para que “se diera cuenta” que se equivocó en la elección de los caminos, y que podía autocorregirse.
- También dijimos que una de las cosas que íbamos a necesitar era:
- una generalización del concepto de camino dirigido no saturado, que permita al algoritmo autocorregirse.
- Antes de introducir la generalización, analicemos un poco lo que tenemos.

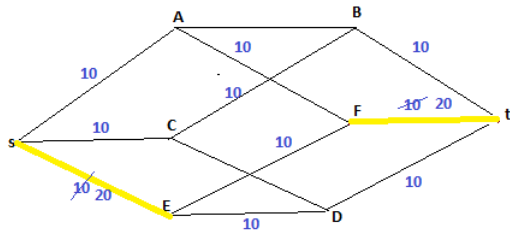
Caminos dirigidos no saturados

- Recordemos uno de los ejemplos que vimos.

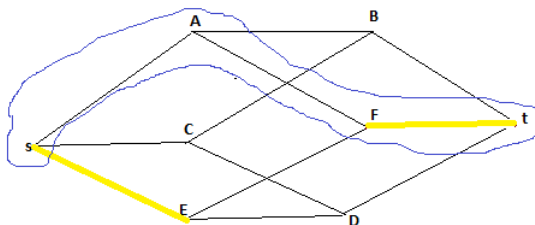


- Capacidades todas 10 excepto los lados marcados de distinto color, de capacidad 20.

- tenemos un flujo de valor 40 en vez de 30.



- En este ejemplo chico podemos ver que el problema se origina cuando elegimos
- En vez de



- En este caso podemos ver “a ojo” lo que habría que corregir pero en un caso cuando hayamos creado miles de caminos aumentantes no es posible examinar cada uno a ver si estaba bien o mal.
- Así que la idea no es esa.
- Imaginemos que nuestro network representa una red de cañerías de agua por la cual estamos queriendo mandar agua desde s a t
- y que en cada vértice $x \neq s, t$ hay un operador, que controla cuanto flujo de agua puede mandar desde x hacia $\Gamma^+(x)$.
- y que cuando estamos queriendo aumentar el flujo, lo que va a ocurrir es lo siguiente:
 - s le pregunta a alguno de sus vecinos de $\Gamma^+(s)$ si le puede mandar flujo.
 - Ese vértice le pregunta a alguno de sus vecinos de Γ^+ si le puede mandar flujo,
 - etc, hasta llegar a t

Caminos dirigidos no saturados

- Cada vértice debe “preguntarle” al siguiente si le puede mandar flujo, pues un vértice $\neq s, t$ no puede aceptar que le manden flujo sin saber si se lo puede “sacar” de encima.
- En el ejemplo podría pensar que s le pregunta a A si le puede mandar flujo, A se fija que le puede mandar flujo a B , así que le pregunta, este se fija que le puede mandar flujo a t , así que le responde que sí, A le dice que sí a s , y se crea el camino $sABt$.

Caminos dirigidos no saturados

- Idem con los otros hasta llegar a $sABt:10$, $sCDt:10$, $sEFt:10$
- Podríamos pensar que el proceso termina así:
- s mira que le puede mandar flujo a E y le pregunta a E si le puede mandar flujo.
- E se fija que $f(\overrightarrow{ED}) = 0 < 10 = c(\overrightarrow{ED})$
- y responde tengo un candidato, esperará un cacho que le pregunto.
- Le pregunta a D que mira que sólo le podría mandar flujo a t , pero $f(\overrightarrow{Dt}) = 10 = c(\overrightarrow{Dt})$.
- D le dice “no” a E , y como E no tiene mas “candidatos”, le dice “no” a s

Idea de FF

- Ford y Fulkerson tuvieron la siguiente idea (bueno, no creo que la hayan tenido exactamente de la forma en que la estoy contando).
- D no puede mandar mas flujo a t porque tuvo que mandarle flujo a t porque C le mandó flujo a el (D)
- ¿porqué no puede D decirle a C “che, podes no mandarme tanto flujo, que tengo que hacerle lugar a E ?”
- Es decir, D le pregunta a C si no puede “devolverle” flujo.
- (no es que realmente le “devuelva” flujo, simplemente le pide a C que no le mande, o al menos no le mande tanto).
- ¿Que hace C ?

Idea de FF

- C podría razonar “ D no quiere que le mande flujo. Si no le mando a el, a quien le puedo mandar?”
- y entonces C se fija que también tiene a B en $\Gamma^+(C)$, con
$$f(\overrightarrow{CB}) < c(\overrightarrow{CB})$$
- así que le pregunta a el si le puede mandar flujo.
- B se fija que no le puede mandar a t , pues
$$f(\overrightarrow{Bt}) = 10 = c(\overrightarrow{Bt}).$$
- Así que está por responderle “no” a Cpero se da cuenta de algo parecido a D :
- B le tuvo que mandar flujo a t porque A le había mandado a EL .
- Así que B le pregunta a A lo mismo que D le preguntó a E .

Idea de FF

- B : “che, A , puedes mandarme menos flujo que C quiere mandarme flujo y no sé que hacer?”
- A se fija que tiene a F en $\Gamma^+(A)$ y que $f(\overrightarrow{AF}) = 0 < 10 = c(\overrightarrow{AF})$.
- así que le pregunta a F si le puede mandar flujo.
- F observa que $t \in \Gamma^+(F)$ y que $f(\overrightarrow{Ft}) = 10 < 20 = c(\overrightarrow{Ft})$ y dice “sí”
- Ese “sí” se propaga por toda la cadena y entonces se puede aumentar el flujo.

Idea de FF

- A redirige los 10 que le mandaba a B a F .
- F le manda 10 unidades mas de flujo a t .
- C le manda a B los 10 que le mandaba a D , pues ahora B no tiene problemas en aceptarlos.
- B sigue mandandole 10 a t .
- s le manda 10 a E .
- E se los manda a D .
- y D sigue mandandole 10 a t .
- lo que queda es equivalente a haber hecho los caminos $sAft$, $sCBt$, $sEft$, $sEDt$ de entrada.

Idea de FF

- Esa es la base para la idea de Ford y Fulkerson:
- Al pararnos en un vértice x , en vez de limitar la búsqueda a vértices a los cuales x les pueda “mandar” flujo.
 - Permitir que x también busque vértices a los cuales les pueda pedir que no le manden más flujo, o al menos no tanto flujo como le están mandando.
- Claro que para pedirle a alguien que no te mande más flujo, ese “alguien” te tiene que haber mandado flujo.
- Entonces, técnicamente, lo que Ford y Fulkerson hacen es que:
 - en vez de limitar la búsqueda a $y \in \Gamma^+(x)$ con $f(\overrightarrow{xy}) < c(\overrightarrow{xy})$
 - permiten además buscar $y \in \Gamma^-(x)$ con $f(\overrightarrow{yx}) > 0$

Camino aumentante

Definición:

Un camino aumentante (o f -camino aumentante si necesitamos especificar f) o camino de Ford-Fulkerson, es una sucesión de vértices x_0, x_1, \dots, x_r tales que:

- $x_0 = s, x_r = t$.
- Para cada $i = 0, \dots, r - 1$ ocurre una de las dos cosas siguientes:

1 $\overrightarrow{x_i x_{i+1}} \in E$ y $f(\overrightarrow{x_i x_{i+1}}) < c(\overrightarrow{x_i x_{i+1}})$ o:

2 $\overrightarrow{x_{i+1} x_i} \in E$ y $f(\overrightarrow{x_{i+1} x_i}) > 0$.

Si en vez de comenzar en s y terminar t el camino es como arriba pero con $x_0 = x, x_r = z$ diremos que es un camino aumentante **desde x a z**

Lados forward y backward

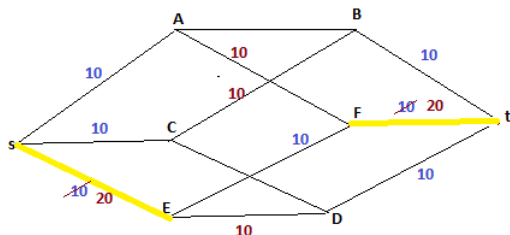
- A los lados en 1) los llamaremos
 - “lados de tipo I” o “lados forward”
- A los lados en 2) los llamaremos
 - “lados de tipo II” o “lados backward”
- Observemos que el orden en que se listan los vértices en el camino aumentante es $\dots x_i, x_{i+1} \dots$ pero como JUSTAMENTE estamos generalizando la noción de camino dirigido, no siempre (x_i, x_{i+1}) será un lado, sino que el lado será $\overrightarrow{x_{i+1} x_i}$ en el caso de los lados backward.

Ejemplo y notación

- En el ejemplo que habíamos dado, s, E, D, C, B, A, F, t es un camino aumentante.
- Denotaremos la acción de mandar 10 unidades de flujo (u otra cantidad) por ese camino así:
- $s \overset{\leftarrow}{E} \overset{\leftarrow}{D} C B A \vec{F} t : 10$
- Las flechas de derecha a izquierda indicando cuales son los lados “backward”
- pues $\overset{\rightarrow}{DC}$ y $\overset{\rightarrow}{BA}$ no existen en el network.
- Los que existen son $\overset{\rightarrow}{CD}$ y $\overset{\rightarrow}{AB}$
- pero el orden en que listamos los vértices es D primero, luego C
- y B primero, luego A .

Notación (cont.)

- además con esa notación sabemos que esos lados deben “devolver” flujo.
- Es decir, el flujo f cambia de la siguiente forma, usando la notación de C:
- $f+ = 10$ en los lados $\overrightarrow{sE}, \overrightarrow{ED}, \overrightarrow{CB}, \overrightarrow{AF}, \overrightarrow{Ft}$.
- $f- = 10$ en los lados $\overrightarrow{CD}, \overrightarrow{AB}$.



Notación (cont.)

- En general, “mandar” ε unidades de flujo por un camino aumentante x_0, x_1, \dots, x_r significará:
- Cambiar f por medio de:
 - $f(\overrightarrow{x_i x_{i+1}}) + = \varepsilon$ en los lados forward.
 - $f(\overrightarrow{x_{i+1} x_i}) - = \varepsilon$ en los lados backwards.
- Podemos dar el nuevo algoritmo:

Ford-Fulkerson

Algoritmo de Ford-Fulkerson para hallar flujo maximal

- 1 Comenzar con $f = 0$ (es decir, $f(\overrightarrow{xy}) = 0 \forall \overrightarrow{xy} \in E$).
- 2 Buscar un f -camino aumentante $s = x_0, x_1, \dots, x_r = t$.
- 3 Definamos ε_i de la siguiente manera:
 - $\varepsilon_i = c(\overrightarrow{x_i x_{i+1}}) - f(\overrightarrow{x_i x_{i+1}})$ en los lados forward.
 - $\varepsilon_i = f(\overrightarrow{x_{i+1} x_i})$ en los lados backward
- 4 Calcular $\varepsilon = \min\{\varepsilon_i\}$.
- 5 Aumentar f a lo largo del camino de 2. en ε , como se explicó antes.
- 6 Repetir 2 hasta que no se puedan hallar mas caminos aumentantes.

Ford-Fulkerson vs Greedy

- El algoritmo de Ford-Fulkerson es muy parecido al Greedy, sólo que en vez de usar caminos dirigidos no saturados, usa caminos aumentantes.
- Ahora bien, si bien vimos que el Greedy no siempre daba con un flujo maximal, al menos era obvio que:
 - cuando el algoritmo cambiaba el flujo, lo que quedaba también era flujo, y:
 - Greedy siempre terminaba, con complejidad polinomial.
- ¿Porqué Greedy siempre termina, y la complejidad es polinomial?
- Lo habia dicho la clase pasada pero no lo deje escrito, asi que repitamoslo:

Complejidad de Greedy vs Ford-Fulkerson

- 1 En cada camino dirigido no saturado que Greedy usa, al menos un lado se satura.
 - 2 Como en Greedy los lados nunca se des-saturan, entonces Greedy puede hacer a lo sumo $O(m)$ incrementos de flujo antes de que forzosamente deba terminar si o si.
 - 3 Encontrar un camino dirigido no saturado es $O(m)$
 - 4 Asi que la complejidad total de Greedy es $O(m^2)$.
- En Ford-Fulkerson vale el [3] arriba, pues encontrar un camino aumentante tambien es $O(m)$.
 - Y tambien vale algo similar al [1] pues en todo camino que use Ford-Fulkerson va a haber al menos un lado que se sature, o un lado que se vacie.
 - El problema es que **no vale [2]**.

Complejidad Ford-Fulkerson

- Justamente como puedes “devolver” flujo a través de los lados backward, ahora los lados **pueden des-saturarse**, como vimos en el ejemplo. (o vaciarse y volverse a llenar)
- Así que no queda claro que la complejidad sea $O(m^2)$ o siquiera si es polinomial.
- De hecho no lo es: veremos un ejemplo con $n = 4$, $m = 5$ en donde Ford-Fulkerson puede hacer dos millones de iteraciones.
- O cuatro mil millones.
- Mas aún veremos que hay ejemplos en donde Ford-Fulkerson **no termina nunca**.

Complejidad Ford-Fulkerson

- Pero ,¿para que estudiamos un algoritmo que no sólo no es polinomial sino que puede NO TERMINAR?!!!!
- Porque vamos a demostrar que si Ford-Fulkerson termina, entonces termina con un flujo maximal.
- y que Ford-Fulkerson es la base de una serie de algoritmos que terminan, y en tiempo polinomial.
- La correctitud de todos ellos se apoya en la correctitud de Ford-Fulkerson, dando todos flujos maximales por la propiedad base de Ford-Fulkerson.

Otro problema con Ford-Fulkerson

- Otro problema que tiene Ford-Fulkerson que no tiene Greedy es el siguiente:
- Con un camino aumentante no saturado, era obvio que al mandar ε unidades de flujo por el camino, si el ε no “reventaba” ningún caño, lo que quedaba era flujo.
- Con Ford-Fulkerson no es completamente obvio que lo que quede luego de aumentar el flujo siga siendo un flujo.
- Hemos dado en el ejemplo introductorio una racionalidad por la cual esto también debería pasar en un camino de Ford-Fulkerson, pero debemos probarlo.

FordFulkerson mantiene “flujicidad”

Teorema:

Si f es un flujo de valor v y aumentamos f con un f -camino aumentante con ε calculado como se explica en el algoritmo de Ford-Fulkerson, entonces lo que queda sigue siendo flujo y el valor del nuevo flujo es $v + \varepsilon$

- Prueba: Sea $s = x_0, x_1, \dots, x_r = t$ el camino aumentante.
- Para diferenciar entre ambos, llamaremos f al flujo antes de aumentar y f^* a lo que queda luego de aumentar.
- Veamos primero que f^* satisface la primera condición de flujo.
- Es decir, tenemos que ver que $0 \leq f^* \leq c$.

Continuación prueba: $f \leq c$

- En los lados backward le restamos algo a f , por lo tanto $f^* < f \leq c$ en ellos.
- Mientras que en los lados forward, tenemos:

$$\begin{aligned}
 f^*(\overrightarrow{x_i x_{i+1}}) &= f(\overrightarrow{x_i x_{i+1}}) + \varepsilon \\
 &\leq f(\overrightarrow{x_i x_{i+1}}) + \varepsilon_i \quad \text{pues } \varepsilon \text{ es el min de los } \varepsilon_i \\
 &= f(\overrightarrow{x_i x_{i+1}}) + c(\overrightarrow{x_i x_{i+1}}) - f(\overrightarrow{x_i x_{i+1}}) \\
 &= c(\overrightarrow{x_i x_{i+1}})
 \end{aligned}$$

Por lo tanto $f^* \leq c$ también vale en los lados forward.

Continuación prueba: $0 \leq f$

- En los lados forward le sumamos algo a f , por lo tanto $0 \leq f < f^*$ en ellos.
- Mientras que en los lados backward, tenemos:

$$\begin{aligned}
 f^*(x_{i+1} \overset{\rightarrow}{x_i}) &= f(x_{i+1} \overset{\rightarrow}{x_i}) - \varepsilon \\
 &\geq f(x_{i+1} \overset{\rightarrow}{x_i}) - \varepsilon_i \quad \text{pues } \varepsilon \leq \varepsilon_i \Rightarrow -\varepsilon \geq -\varepsilon_i \\
 &= 0 \quad \text{pues } \varepsilon_i = f(x_{i+1} \overset{\rightarrow}{x_i}) \text{ en los lados backward}
 \end{aligned}$$

Por lo tanto $f^* \geq 0$ también vale en los lados backward.
Ahora tenemos que ver la ley de conservación para f^* .

Continuación prueba: $in = out$

- Sea $x \neq s, t$.
- Si x no es ningún x_i , entonces como no se cambia ningún lado que entre o salga de x , tenemos:
 - $in_{f^*}(x) = in_f(x)$ y $out_{f^*}(x) = out_f(x)$
 - Por lo tanto $in_{f^*}(x) = out_{f^*}(x)$.
- Ahora tomemos un $x = x_i$.
- Como $x \neq s, t$, entonces $0 < i < r$.
- Y entonces los vértices x_{i-1} y x_{i+1} existen.
- Los lados que cambian son los lados entre x_{i-1} y x_i y entre x_i y x_{i+1} .
- Pero no sabemos si son forward o backward.

Continuación prueba: $in = out$

- Caso 2: Los dos son backward.
- En este caso la prueba es muy similar a la anterior, reemplazando $+$ por $-$:
 - Los lados que existen son $\overrightarrow{x_i x_{i-1}}$ y $\overrightarrow{x_{i+1} x_i}$
 - $f^* = f - \varepsilon$ en ambos lados.
 - Como $\overrightarrow{x_i x_{i-1}}$ “sale” de x_i , entonces:
 - $out_{f^*}(x) = out_f(x) - \varepsilon$ (*)
 - Como $\overrightarrow{x_{i+1} x_i}$ “entra” a x_i , entonces:
 - $in_{f^*}(x) = in_f(x) - \varepsilon$. (**)
 - (*) y (**) implican $in_{f^*}(x) = out_{f^*}(x)$

Continuación prueba: $in = out$

- Caso 4: El primero es backward, el segundo forward.
- Similar al anterior
 - Los lados que existen son $\overrightarrow{x_i x_{i-1}}$ y $\overrightarrow{x_i x_{i+1}}$
 - $f^* = f - \varepsilon$ en el primero y $f^* = f + \varepsilon$ en el segundo.
 - En este caso, como ningún lado que entra a x_i es cambiado, entonces $in_{f^*}(x) = in_f(x)$ (*)
 - Como $\overrightarrow{x_i x_{i-1}}$ sale de x_i y contribuye con $-\varepsilon$
 - y $\overrightarrow{x_i x_{i+1}}$ también sale de x_i y contribuye con $+\varepsilon$
 - entonces $out_{f^*}(x) = out_f(x) + \varepsilon - \varepsilon = out_f(x)$.
 - Esto último y (*) implican $in_{f^*}(x) = out_{f^*}(x)$
- Con lo que f^* satisface la propiedad de conservación

Continuación prueba: $v(f^*) = v + \varepsilon$

- Para ver la tercera propiedad de flujos y calcular el valor de f^* simultaneamente tenemos que analizar dos casos:

- Caso 1: s, x_1 es forward.

- En este caso el lado que existe es $\overrightarrow{s x_1}$ y $f^* = f + \varepsilon$ en ese lado.
- Por lo tanto $in_{f^*}(s) = in_f(s)$ y $out_{f^*}(s) = out_f(s) + \varepsilon$. Asi que:
- $v(f^*) = out_{f^*}(s) - in_{f^*}(s) = out_f(s) + \varepsilon - in_f(s) = v(f) + \varepsilon$.

- Caso 1: s, x_1 es backward.

- En realidad en ninguno de los ejemplos que daremos o los algoritmos que usaremos se puede producir este caso.
- Pero probemoslo igual, por completitud.
- El lado que existe es $\overrightarrow{x_1 s}$ y $f^* = f - \varepsilon$
- $in_{f^*}(s) = in_f(s) - \varepsilon$ y $out_{f^*}(s) = out_f(s)$.
- $v(f^*) = out_{f^*}(s) - in_{f^*}(s) = out_f(s) - (in_f(s) - \varepsilon) = v(f) + \varepsilon$.

- Fin.

Correctitud de Ford-Fulkerson

- Bien, hemos visto la primera parte de la correctitud de Ford-Fulkerson: en todo momento, las funciones intermedias que produce son flujos.
- Por lo tanto, si termina, lo que devuelve es un flujo.
- Lo que queremos ver a continuación es que, si termina, ese flujo que devuelve es maximal.
- Pero para eso necesitaremos un concepto nuevo y otro teorema.

MFMC

- Ya estamos en condiciones de probar que el algoritmo de Ford-Fulkerson, si termina, termina con un flujo maximal.
- La prueba tambien involucra probar el famoso “Max Flow Min Cut Theorem” : el teorema del flujo maximal y corte minimal.
- Este teorema relaciona, como el nombre lo indica, flujos maximales y cortes minimales.
- Pero ademas, permite modificar el algoritmo de Ford-Fulkerson para que si termina, no sólo termine con un flujo maximal sino con un CERTIFICADO de que el flujo es maximal.
- Es decir, con algo que permite verificar que el flujo es maximal, independientemente del algoritmo.

Max Flow Min Cut

Teorema de Ford-Fulkerson(Max Flow Min Cut):

- A** Si f es un flujo y S es un corte, entonces

$$v(f) = f(S, \bar{S}) - f(\bar{S}, S).$$
- B** El valor de todo flujo es menor o igual que la capacidad de todo corte.
- C** Si f es un flujo, las siguientes afirmaciones son equivalentes:
 - 1** Existe un corte S tal que $v(f) = \text{cap}(S)$.
 - 2** f es maximal.
 - 3** No existen f -caminos aumentantes.
- Y si se cumplen, el S de [1] es minimal.
- En realidad suele llamarse Max Flow Min Cut a la parte [C] específicamente.
- o incluso mas restrictivo, a la implicación $2 \Rightarrow 1$.

- Prueba: veamos A:
- Sea f un flujo cualquiera y S un corte cualquiera.
- Por la propiedad de la conservación, tenemos que
- $out_f(x) - in_f(x) = 0$ para todo $x \neq s, t$.
- Y para $x = s$, tenemos $out_f(s) - in_f(s) = v(f)$.
- $x = t$ no nos interesa pues miraremos sólo $x \in S$, y S es un corte.

Calculando $v(f)$ por medio de un corte

Entonces:

$$\begin{aligned}
 & \sum_x (out_f(x) - in_f(x)) [x \in S] = \\
 &= out_f(s) - in_f(s) + \sum_x (out_f(x) - in_f(x)) [x \in S][x \neq s] \\
 &= v(f) + \sum_x 0 [x \in S][x \neq s] \\
 &= v(f)
 \end{aligned}$$

Usemos esto que hemos probado, y calculemos

$\sum_x (out_f(x) - in_f(x)) [x \in S]$ de otra forma.

Observación trivial

- Pero antes, veamos una observación trivial: sea f definida en lados y $A, B, C \subseteq V$ tales que $A \cap C = \emptyset$.
- Entonces es trivial ver que $f(A \cup C, B) = f(A, B) + f(C, B)$.
- y similarmente $f(B, A \cup C) = f(B, A) + f(B, C)$. pej la primera:

$$\begin{aligned}
 f(A \cup C, B) &= \sum_{x,y} [x \in A \cup C][y \in B][\overrightarrow{xy} \in E] f(\overrightarrow{xy}) \\
 &= \sum_{x,y} ([x \in A] + [x \in C])[y \in B][\overrightarrow{xy} \in E] f(\overrightarrow{xy}) \\
 &= \sum_{x,y} [x \in A][y \in B][\overrightarrow{xy} \in E] f(\overrightarrow{xy}) + \\
 &\quad + \sum_{x,y} [x \in C][y \in B][\overrightarrow{xy} \in E] f(\overrightarrow{xy}) \\
 &= f(A, B) + f(C, B)
 \end{aligned}$$

Calculando $v(f)$ por medio de un corte

Volviendo a la prueba, habíamos visto que

$\sum_x (out_f(x) - in_f(x)) [x \in S] = v(f)$, así que:

$$\begin{aligned}
 v(f) &= \sum_x (out_f(x) - in_f(x)) [x \in S] \\
 &= \sum_x (f(\{x\}, V) - f(V, \{x\})) [x \in S] \\
 &= \sum_x f(\{x\}, V) [x \in S] - \sum_x f(V, \{x\}) [x \in S] \\
 &= f(S, V) - f(V, S) \\
 &= f(S, S \cup \bar{S}) - f(S \cup \bar{S}, S) \\
 &= f(S, S) + f(S, \bar{S}) - f(S, S) - f(\bar{S}, S) \\
 &= f(S, \bar{S}) - f(\bar{S}, S)
 \end{aligned}$$

Fin parte [A]

Observación secundaria

- Esta observación no es parte de la prueba pero es un buen lugar para hacerla.
- Recordemos que $v(f)$ esta definido como $out_f(s) - in_f(s) = f(\{s\}, V) - f(V, \{s\})$.
- Eso tambien es igual a $v(f) = f(\{s\}, V - \{s\}) - f(V - \{s\}, \{s\})$.
- Puesto que $S = \{s\}$ es un corte, vemos que la definición de $v(f)$ es un caso particular de lo que acabamos de probar que $v(f) = f(S, \bar{S}) - f(\bar{S}, S)$.
- Tambien habiamos probado que $v(f) = in_f(t) - out_f(t) = f(V - \{t\}, \{t\}) - f(\{t\}, V - \{t\})$.
- Como $V - \{t\}$ es corte y $V - (V - \{t\}) = \{t\}$, esto tambien es un caso particular.

Parte [B]

Prueba de [B]:

$$\begin{aligned} v(f) &= f(S, \bar{S}) - f(\bar{S}, S) \\ &\leq f(S, \bar{S}) \\ &\leq c(S, \bar{S}) = \text{cap}(S) \end{aligned}$$

Fin B. Ahora probemos C.

Max Flow Min Cut, prueba.

- Para probar la equivalencia de 1,2,3 probaremos que $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.
- Las 2 primeras son faciles.

Max Flow Min Cut, prueba.

- Para probar la equivalencia de 1,2,3 probaremos que $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.
- Las 2 primeras son faciles.
- $1 \Rightarrow 2$
 - Sean S, f como en [1] y sea g un flujo cualquiera.
 - Por la parte [B], $v(g) \leq \text{cap}(S)$.
 - Pero por hipotesis tenemos que $\text{cap}(S) = v(f)$.
 - Concluimos que $v(g) \leq v(f)$ y por lo tanto f es maximal
 - Ademas, si T es un corte, $\text{cap}(T) \geq v(f) = \text{cap}(S)$, es decir, S es minimal.

Max Flow Min Cut, prueba.

- $2 \Rightarrow 3$: (es decir ver que si f es maximal entonces no existen f -caminos aumentantes)

Max Flow Min Cut, prueba.

■ $2 \Rightarrow 3$:

- Si existiese un f -camino aumentante, podríamos mandar un $\varepsilon > 0$ a través de el,
- Obtendríamos un flujo f^* tal que $v(f^*) = v(f) + \varepsilon$.
- Esto diría que $v(f^*) > v(f)$
- lo cual contradice que f sea maximal.

Max Flow Min Cut, prueba.

- $3 \Rightarrow 1$: Es decir, probar que si no existen f -caminos aumentantes entonces existe un corte S tal que $v(f) = \text{cap}(S)$

Max Flow Min Cut, prueba.

- $3 \Rightarrow 1$:
- (esta es la parte principal del teorema, las otras dos implicaciones eran muy fáciles, como vimos.
- Necesitamos construir un S que sea corte con $cap(S) = v(f)$. Primero lo definimos y luego probamos sus propiedades:

$$S = \{s\} \cup \{x \in V : \text{exista un } f\text{-camino aumentante desde } s \text{ a } x\}$$

Max Flow Min Cut, prueba.

- Como estamos suponiendo que vale (3), entonces $t \notin S$.
- Como $s \in S$ por definición, y $t \notin S$, entonces S es un corte.
- Por lo tanto, por la cuenta que hicimos en la parte [A], tenemos que $v(f) = f(S, \bar{S}) - f(\bar{S}, S)$
- Calculemos $f(S, \bar{S})$ y $f(\bar{S}, S)$.

Max Flow Min Cut, continuación prueba 3 \Rightarrow 1

- $f(S, \bar{S}) = \sum_{x,y} f(\overrightarrow{xy})[x \in S][y \notin S][\overrightarrow{xy} \in E]$
- Consideremos un par x, y de los que aparecen en esa suma.
- Como $x \in S$, entonces existe un f -camino aumentante entre s y x , digamos $s = x_0, x_1, \dots, x_r = x$.
- Pero como $y \notin S$, entonces no existe ningún f -camino aumentante entre s e y .
- En particular

$$s = x_0, x_1, \dots, x_r = x, y$$

NO ES un f -camino aumentante.

- Pero $\overrightarrow{xy} \in E$, así que PODRIA serlo.

Max Flow Min Cut, continuación prueba 3 \Rightarrow 1

- ¿Porqué $s = x_0, x_1, \dots, x_r = x, y$ no es un f -camino aumentante a pesar de que $s = x_0, x_1, \dots, x_r = x$ si lo es y \overrightarrow{xy} existe?
- La única razón por la cual no es un f -camino aumentante es porque no podemos usar el lado \overrightarrow{xy} por estar saturado, es decir: $f(\overrightarrow{xy}) = c(\overrightarrow{xy})$.
- Esto es cierto para cualesquiera x, y que aparezcan en esa suma.
- Entonces:

$$\begin{aligned}
 f(S, \bar{S}) &= \sum_{x,y} f(\overrightarrow{xy}) [x \in S] [y \notin S] [\overrightarrow{xy} \in E] \\
 &= \sum_{x,y} c(\overrightarrow{xy}) [x \in S] [y \notin S] [\overrightarrow{xy} \in E] \\
 &= c(S, \bar{S}) = \text{cap}(S)
 \end{aligned}$$

Max Flow Min Cut, continuación prueba 3 \Rightarrow 1

- Veamos ahora $f(\bar{S}, S)$:
- $f(\bar{S}, S) = \sum_{x,y} f(\overrightarrow{xy}) [x \notin S] [y \in S] [\overrightarrow{xy} \in E]$
- Consideremos un par x, y de los que aparecen en esa suma.
- Como $y \in S$, entonces existe un f -camino aumentante entre s e y , digamos $s = x_0, x_1, \dots, x_r = y$.
- Pero como $x \notin S$, entonces no existe un f -camino aumentante entre s y x .
- En particular

$$s = x_0, x_1, \dots, x_r = y, x$$

NO ES un f -camino aumentante.

- Pero $\overrightarrow{xy} \in E$, así que PODRIA serlo, usando y, x como lado backward.

Max Flow Min Cut, continuación prueba 3 \Rightarrow 1

- ¿Porqué $s = x_0, x_1, \dots, x_r = y$, x no es un f -camino aumentante a pesar de que $s = x_0, x_1, \dots, x_r = y$ si lo es y \overrightarrow{xy} existe?
- La única razón es que no podemos usarlo como lado backward, es decir, que $f(\overrightarrow{xy}) = 0$.
- Esto es cierto para cualesquiera x, y que aparezcan en esa suma.
- Entonces:

$$\begin{aligned}
 f(\overline{S}, S) &= \sum_{x,y} f(\overrightarrow{xy})[x \notin S][y \in S][\overrightarrow{xy} \in E] \\
 &= \sum_{x,y} 0[x \notin S][y \in S][\overrightarrow{xy} \in E] \\
 &= 0
 \end{aligned}$$

Max Flow Min Cut, conclusión prueba 3 \Rightarrow 1

- Entonces hemos probado que para este S :

- 1 $f(S, \bar{S}) = \text{cap}(S)$

- 2 $f(\bar{S}, S) = 0$

- Por lo tanto:

$$\begin{aligned} v(f) &= f(S, \bar{S}) - f(\bar{S}, S) \\ &= \text{cap}(S) - 0 \\ &= \text{cap}(S) \end{aligned}$$

- Con lo cual hemos probado (1). Fin

Correctitud de Ford-Fulkerson

Corolario:

Si el algoritmo de Ford-Fulkerson termina, termina con un flujo maximal

- Prueba: si Ford-Fulkerson termina, entonces termina con un flujo f para el cual no hay mas f -caminos aumentantes.
- Por lo tanto f es maximal, por el Max Flow Min Cut Theorem. Fin

S como certificado

- Observación: mientras se corre Ford-Fulkerson, durante la búsqueda de un camino aumentante, se puede ir construyendo S .
- Si $t \in S$, existe un f -camino aumentante y podemos seguir.
- Si $t \notin S$, el algoritmo se detendrá, el flujo será maximal, y el último S construido servirá como “certificado” de que f es maximal.
- Pues dado el f y el S , se pueden calcular $v(f)$ y $cap(S)$ en forma independiente, y verificar si son iguales.
- Por la parte $1 \Rightarrow 2$ del teorema, si son iguales sabemos que f es maximal.
- En los ejercicios prácticos esto sirve para chequear que uno no haya cometido algún error tal que el flujo calculado no sea en realidad un flujo maximal.

Flujos enteros maximales

- Aún con las limitaciones que tiene Ford-Fulkerson, es suficiente para probar un teorema importante.
- Ya habíamos hablado antes sobre lo de abajo, pero lo vuelvo a recordar para plantear el teorema.
- En la definición de flujo, no se requiere que $f(\overrightarrow{xy})$ sea un entero para todo lado. Llamabamos “entero” a tales flujos.
- Algunos libros, como el Biggs, si requieren eso.
- Si uno limita los flujos a flujos enteros, entonces como hay una cantidad finita de ellos es obvio que hay flujos enteros maximales, es decir, flujos que son maximales **entre los flujos enteros**.

Teorema de la Integralidad

- Pero no necesariamente serán maximales entre **todos** los flujos.
- Pej si tenemos un sólo lado st de capacidad 10,4 entonces el flujo entero maximal tiene valor 10, pero el flujo maximal tiene valor 10,4.
- Esto es consecuencia directa de que hay lados con capacidades no enteras.
- Pero ¿qué pasa si todas las capacidades son enteras?
- Ese es el Teorema de la Integralidad:

Teorema de la integralidad.

En un network con capacidades enteras, todo flujo entero maximal es un flujo maximal.

Teorema de la Integralidad

- El Teorema de la Integralidad sale directo del siguiente Teorema:

Teorema

En un network donde todas las capacidades sean enteros, Ford-Fulkerson siempre termina y el flujo maximal resultante es un flujo entero.

- Este teorema demuestra el Teorema de la Integralidad porque ya vimos que si Ford-Fulkerson termina, entonces termina con un flujo maximal.
- Para demostrarlo demostraremos primero que todos los flujos intermedios que se producen en el algoritmo de Ford-Fulkerson son enteros.
- Asi que obviamente el flujo final será entero.

- A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

Prueba del Teorema de la Integralidad

- Como ε es el mínimo de los ε_i y estos son todos enteros, entonces ε es entero.
- Como f es cambiado sumando o restando ε en los lados, entonces el nuevo f también es entero.
- Esto concluye con la primera parte de la prueba. Para poder terminar la prueba debemos ver que efectivamente Ford-Fulkerson siempre termina si las capacidades son enteras.
- Vimos que el ε es entero, y sabemos que es positivo, porque se calcula en un camino aumentante.
- Por lo tanto, en cada aumento de flujo, el valor del flujo aumenta en AL MENOS UNO.

Prueba del Teorema de la Integralidad

- Como el valor de todo flujo es menor o igual que la capacidad de cualquier corte, todo flujo debe tener un valor acotado por pej, $cap(\{s\})$.
- Como en cada calculo de un nuevo flujo el valor aumenta en al menos uno,
- concluimos que Ford-Fulkerson puede calcular a lo sumo $cap(\{s\})$ flujos intermedios, y luego si o si debe terminar en el caso de capacidades enteras.
- Fin.