1. Tabla de Contenidos
Contents

2. Caminos aumentantes

2.1. **Idea y motivación.** En clase vimos un ejemplo donde Greedy puede no obtener un flujo maximal, dependiendo de la elección de caminos.

Imaginemos que nuestro network representa una red de cañerias 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 **su** Γ^+ si le puede mandar flujo,
- etc, hasta llegar a t

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 que dimos ($\overrightarrow{sA}, \overrightarrow{sB}, \overrightarrow{AC}, \overrightarrow{BC}, \overrightarrow{Ct}, \overrightarrow{Dt}$ con capacidad 20 y $\overrightarrow{AD}$ y $\overrightarrow{BD}$ de capacidades 10) se podria pensar que s le pregunta a A si le puede mandar flujo, A se fija que le puede mandar flujo a C, asi que le pregunta, este se fija que le puede mandar flujo a t, asi que le responde que si, A le dice que si a s, y se crea el camino $sACt$.

En clase vimos que luego de $sACt : 20$, $sBDt : 10$, no podiamos seguir pues si bien s le puede mandar flujo a B y este a C, ahi ya no podemos seguir porque el lado $\overrightarrow{BC}$ esta saturado.

Podriamos pensar que el proceso termina asi: s mira que le puede mandar flujo a B y le pregunta a B si le puede mandar flujo. B se fija que $f(\overrightarrow{BC}) = 0 < 10 = c(\overrightarrow{BC})$ y le responde a A: "tengo un candidato, esperá un cacho que le pregunto."

B le pregunta a C si puede mandarle flujo. C mira que sólo le podria mandar flujo a t, pero $f(\overrightarrow{Ct}) = 20 = c(\overrightarrow{Ct})$. Asi que C le dice "no" a B, y como B no tiene mas "candidatos" , (pues el otro lado, BD esta saturado: $f(\overrightarrow{BD}) = 10 = c(\overrightarrow{BD})$) le dice "no" a s

Ford y Fulkerson tuvieron la siguiente idea (bueno, no creo que la hayan tenido exactamente de la forma en que la estoy contando). C no puede mandar mas flujo a t porque tuvo que mandarle flujo a t porque A le mandó flujo a el (C) ¿porqué no puede C decirle a A "che, podes no mandarme tanto flujo, que tengo que hacerle lugar a B?"

Es decir, C le pregunta a A 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 A?

A podria razonar "C no quiere que le mande tanto flujo. Si no le mando a el, a quien le puedo mandar?"y entonces A se fija que tambien tiene a D en $\Gamma^+(A)$, con $f(\overrightarrow{AD}) = 0 < 10 = c(\overrightarrow{AD})$ asi que le pregunta a D si le puede mandar flujo. D se fija que le puede mandar flujo a t, pues $f(\overrightarrow{Dt}) = 10 < 20 = c(\overrightarrow{Ct})$ asi que le responde a A: "si"

Ese "si"se propaga por toda la cadena y entonces se puede aumentar el flujo.

(1) A redirige 10 de los 20 que le mandaba a C hacia D.
(2) D le manda 10 unidades mas de flujo a t.
(3) s le manda 10 mas unidades a B.
(4) B se los manda a C.
(5) y C sigue mandandole 20 a t. (10 que recibe de A y 10 de B)

Esa es la base para la idea de Ford y Fulkerson: Al pararnos en un vértice x, en vez de limitar la busqueda a vértices en $\Gamma^+(x)$ a los cuales x les pueda "mandar" flujo, **permitir que x tambien busque vértices en $\Gamma^-(x)$ a los cuales les pueda pedir que no le manden mas flujo, o al menos no tanto flujo como le estan mandando.**

Claro que para pedirle a alguien que no te mande mas flujo, ese "alguien" te tiene que haber mandado flujo.

Entonces, técnicamente, lo que Ford y Fulkerson hacen es que: en vez de limitar la busqueda a $y \in \Gamma^+(x)$ con $f(\overrightarrow{xy}) < c(\overrightarrow{xy})$ permiten ademas buscar $y \in \Gamma^-(x)$ con $f(\overrightarrow{yx}) > 0$

2.1.1. *Definición:* ***Un camino aumentante (o f-camino aumentante si necesitamos especificar f) o camino de Ford-Fulkerson, entre x y z es una sucesión de vértices $x_0, x_1, ..., x_r$ tales que:***

- $x_0 = x$, $x_r = z$.
- *Para cada $i = 0, ..., r-1$ ocurre una de las dos cosas siguientes:*
  I $x_i\overrightarrow{x_{i+1}} \in E$ *y* $f(x_i\overrightarrow{x_{i+1}}) < c(x_i\overrightarrow{x_{i+1}})$ *o:*
  II $x_{i+1}\overrightarrow{x_i} \in E$ *y* $f(x_{i+1}\overrightarrow{x_i}) > 0$.

Si $x = s$ y $z = t$ diremos directamente un camino aumentante o f-camino aumentante. ("augmenting path"en ingles). Es decir, si no especificamos los vertices de salida y llegada, se asumen que son la fuente y el resumidero.

A los lados en I los llamaremos "lados forward"y a los lados en II los llamaremos "lados backward"

Observemos que el orden en que se listan los vértices en el camino aumentante es $...x_i, x_{i+1}...$ 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 $x_{i+1}\overrightarrow{x_i}$ en el caso de los lados backward.

En el ejemplo que habiamos dado, s, B, C, A, D, t es un camino aumentante.

Denotaremos la acción de mandar 10 unidades de flujo (u otra cantidad) por ese camino asi:

$$s\overleftarrow{BCA}Dt : 10$$

Las flechas de derecha a izquierda indicando cuales son los lados "backward". Ademas con esa notación sabemos que esos lados deben "devolver" flujo.

En general, "mandar" ε unidades de flujo por un camino aumentante $x_0, x_1, ..., x_r$ significará:

Cambiar f por medio de:

- $f(x_i\overrightarrow{x_{i+1}}) += \varepsilon$ en los lados forward.
- $f(\overrightarrow{x_{i+1}x_i}) -= \varepsilon$ en los lados backwards.

Podemos dar el nuevo algoritmo:

3. Algoritmo de Ford-Fulkerson

3.1. **Algoritmo de FF y comparación con Greedy.** .

Pseudocodigo:

3.1.1. *Algoritmo de Ford-Fulkerson para hallar flujo maximal*

(1) $f = 0$ *(es decir,* $f(\overrightarrow{xy}) = 0 \forall \overrightarrow{xy} \in E$*).*
(2) *While(existe* f*-camino aumentante entre* s *y* t*)*
   (a) *Tomar un* f*-camino aumentante* $s = x_0, x_1, ..., x_r = t$*.*
   (b) *Definir* ε_i *de la siguiente forma:*
      - $\varepsilon_i = c(x_i\overrightarrow{x_{i+1}}) - f(x_i\overrightarrow{x_{i+1}})$ *en los lados forward.*
      - $\varepsilon_i = f(\overrightarrow{x_{i+1}x_i})$ *en los lados backward*
   (c) *Tomar* $\varepsilon = \min\{\varepsilon_i\}$*.*
   (d) *Aumentar* f *a lo largo del camino en* ε*.*
(3) *EndWhile*
(4) *Return(*f*)*

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.($O(m^2)$) La clave de la complejidad era que una vez que un lado se saturaba, nunca se desaturaba, y como en cada iteracion del While de Greedy se satura un lado, hay $O(m)$ iteraciones, cada una de las cuales es $O(m)$. Pero como en Ford-Fulkerson los lados se des-saturan para luego quizas volverse a saturar, en principio Ford-Fulkerson podria no terminar nunca. (luego veremos ejemplos donde demora mucho o no termina nunca)

¿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 que tiene Ford-Fulkerson que no tiene Greedy es el siguiente: Con un camino **dirigido** 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.

3.1.2. *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 $s = x_0, x_1, ..., x_r = t$ 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 $0 \leq f^* \leq c$.

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:

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

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

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:

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

Por lo tanto $f^* \geq 0$ tambien vale en los lados backward.

Ahora tenemos que ver la ley de conservación para f^*.

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:

$$in_{f^*}(x) = in_f(x), \qquad out_{f^*}(x) = out_f(x)$$

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}. Como no sabemos si son forward o backward, tendremos que analizar cuatro casos, dependiendo de las combinaciones posibles de forward y backward entre ellos.

(1) Los dos son forward.

Los lados que existen son entonces $x_{i-1}\overrightarrow{x_i}$ y $x_i\overrightarrow{x_{i+1}}$ $f^* = f + \varepsilon$ en ambos lados. Como $x_{i-1}\overrightarrow{x_i}$ "entra" a x_i, entonces: $in_{f^*}(x) = in_f(x) + \varepsilon$. (*)

Y como $x_i\overrightarrow{x_{i+1}}$ "sale"de x_i, entonces: $out_{f^*}(x) = out_f(x) + \varepsilon$ (**)

Como sabemos que $in_f(x) = out_f(x)$ entonces (*) y (**) implican $in_{f^*}(x) = out_{f^*}(x)$.

(2) Los dos son backward. En este caso la prueba es muy similar a la anterior, reemplazando + por −:

Los lados que existen son $x_i\overrightarrow{x_{i-1}}$ y $x_{i+1}\overrightarrow{x_i}$. $f^* = f - \varepsilon$ en ambos lados. Como $x_i\overrightarrow{x_{i-1}}$ "sale"de x_i, entonces $out_{f^*}(x) = out_f(x) - \varepsilon$ (*) y como $x_{i+1}\overrightarrow{x_i}$ "entra" a x_i, entonces $in_{f^*}(x) = in_f(x) - \varepsilon$. (**) por lo tanto (*) y (**) implican $in_{f^*}(x) = out_{f^*}(x)$

(3) El primero es forward, el segundo backward.

Los lados que existen son $x_{i-1}\overrightarrow{x_i}$ y $x_{i+1}\overrightarrow{x_i}$

(4) $f^* = f + \varepsilon$ en el primero y $f^* = f - \varepsilon$ en el segundo. Como $x_{i-1}\overrightarrow{x_i}$ "entra" a x_i, entonces: este lado contribuye con un "$+\varepsilon$" a $in_{f^*}(x)$. Pero como $x_{i+1}\overrightarrow{x_i}$ también entra a x_i, este último lado contribuye con "$-\varepsilon$" a $in_{f^*}(x)$. Por lo tanto $in_{f^*}(x) = in_f(x) + \varepsilon - \varepsilon = in_f(x)$. (*)

Como ningún lado que sale de x_i es cambiado, entonces $out_{f^*}(x) = out_f(x)$ Esto último y (*) implican $in_{f^*}(x) = out_{f^*}(x)$

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

Por lo tanto hemos demostrado que f^* satisface la propiedad de conservación

Para ver la tercera propiedad de flujos y calcular el valor de f^* simultaneamente observemos que como $in_f(s) = 0$ entonces el lado s, x_1 solo puede ser forward. Es decir, el lado que existe es $s\overrightarrow{x_1}$ y $f^* = f + \varepsilon$ en ese lado. Por lo tanto $in_{f^*}(s) = in_f(s) = 0$ y $out_{f^*}(s) = out_f(s) + \varepsilon$. Asi que hemos probado la tercer propiedad de flujo y $v(f^*) = out_{f^*}(s) = out_f(s) + \varepsilon = v(f) + \varepsilon$.

Fin del teorema.

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 antes, un par de cosas.

4. Max Flow Min Cut Theorem

4.1. **Preliminares.** En la ultima clase habiamos probado lo siguiente pero lo agregamos aca por completitud:

4.1.1. *Lema*

Si f es un flujo y S es un corte, entonces $v(f) = f(S, \overline{S}) - f(\overline{S}, S)$.

Prueba:

$out_f(x) - in_f(x) = 0$ para todo $x \neq s, t$.

Y para $x = s$, tenemos $out_f(s) - in_f(s) = v(f) - 0 = v(f)$.

$x = t$ no nos interesa pues miraremos sólo $x \in S$, y S es un corte.

Para no escribir sumatorias con grandes subcondiciones abajo, lo cual complica el display en un pdf, usaremos la notacion [P]= 1 si P es verdadera y 0 si P es falsa.

Entonces:

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

Calculemos $\sum_x (out_f(x) - in_f(x))[x \in S]$ de otra forma. 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 $f(B, A \cup C) = f(B, A) + f(B, C)$:

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

Volviendo a la prueba, habiamos visto que $\sum_x (out_f(x) - in_f(x))[x \in S] = v(f)$, asi que:

$$\begin{aligned} v(f) &= \sum_x (out_f(x) - in_f(x))[x \in S] \\ &= \sum_x out_f(x)[x \in S] - \sum_x in_f(x)[x \in S] \\ &= \sum_x \sum_y f(x, y)[y \in \Gamma^+(x)][x \in S] - \sum_x \sum_y f(y, x)[y \in \Gamma^-(x)][x \in S] \\ &= \sum_x \sum_y f(x, y)[\overrightarrow{xy} \in E][x \in S] - \sum_x \sum_y f(y, x)[\overrightarrow{yx} \in E][x \in S] \\ &= f(S, V) - f(V, S) \\ &= f(S, S \cup \overline{S}) - f(S \cup \overline{S}, S) \\ &= f(S, S) + f(S, \overline{S}) - f(S, S) - f(\overline{S}, S) \\ &= f(S, \overline{S}) - f(\overline{S}, S) \qquad \text{Fin.} \end{aligned}$$

4.1.2. *Definición La capacidad de un corte es* $cap(S) = c(S, \overline{S})$.

4.1.3. *Teorema El valor de todo flujo es menor o igual que la capacidad de todo corte.*

Prueba: Como $0 \leq f(\overrightarrow{xy})$ para todo lado, entonces $0 \leq f(A, B)$ para cualesquiera subconjuntos A, B pues $f(A, B)$ es una suma de $f(\overrightarrow{xy})$'s.

En particular, $0 \leq f(\overline{S}, S)$ y por lo tanto $-f(\overline{S}, S) \leq -0$.

Ademas, como $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$ entonces $f(A, B) \leq c(A, B)$ para cualesquiera subconjuntos A, B.

Entonces, usando 4.1.1:

$$v(f) = f(S, \overline{S}) - f(\overline{S}, S) \leq c(S, \overline{S}) - 0 = cap(S)$$

Fin.

4.1.4. *Teorema Si f es un flujo tal que existe un corte S con $v(f) = cap(S)$. entonces f es maximal. (y S es "minimal", es decir, su capacidad es la menor de todas las capacidades de cortes)*

Prueba: Sea g un flujo cualquiera. Entonces, por 4.1.3 $v(g) \leq cap(S)$. Como $v(f) = cap(S)$ tenemos $v(g) \leq cap(S) = v(f)$ es decir $v(g) \leq v(f)$ lo cual implica que f es maximal.

Si T es un corte, entonces $cap(T) \geq v(f) = cap(S)$ demuestra que S es minimal.

Fin.

4.2. **Enunciado del teorema.** Probaremos que si el algoritmo de Ford-Fulkerson termina, entonces 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.

Recien vimos que si $v(f) = cap(S)$ entonces f es maximal. El Max Flow Min Cut dice que vale la vuelta:

4.2.1. *Teorema de Ford-Fulkerson(Max Flow Min Cut):*
Si f es maximal, entonces existe un corte S tal que $v(f) = cap(S)$.

Prueba:
Definamos:

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

Como f es maximal entonces no existen f-caminos aumentantes desde s a t pues si existiese un tal f-camino aumentante, podriamos mandar un $\varepsilon > 0$ a través de el, obteniendo un flujo f^* tal que $v(f^*) = v(f) + \varepsilon > v(f)$ lo cual contradice que f sea maximal. Por lo tanto, como no existen f-caminos aumentantes desde s a t entonces $t \notin S$. Como $s \in S$, entonces S es un corte.

Observemos que en el resto de la prueba no usaremos que f es maximal, solo que es flujo y que S es corte. Es decir, la prueba valdria para cualquier f tal que el S definido sea un corte. Esto será importante luego.

Como f es flujo y S es corte, entonces $v(f) = f(S,\overline{S}) - f(\overline{S},S)$. Calculemos $f(S,\overline{S})$ y $f(\overline{S},S)$.

(1) $f(S,\overline{S})$:

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

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 $s = x_0, x_1, ..., x_r = x$. Pero como $y \notin S$, entonces no existe ningún f-camino aumentante entre s e y. En particular

$$s = x_0, x_1, ..., x_r = x, y$$

NO ES un f-camino aumentante. Pero $\overrightarrow{xy} \in E$, asi que PODRIA serlo.

¿Porqué $s = x_0, x_1, ..., x_r = x, y$ no es un f-camino aumentante a pesar de que $s = x_0, x_1, ..., x_r = x$ 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:

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

(2) $f(\overline{S},S)$:

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

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 $s = x_0, x_1, ..., x_r = y$. Pero como $x \notin S$, entonces no existe un f-camino aumentante entre s y x. En particular

$$s = x_0, x_1, ..., x_r = y, x$$

NO ES un f-camino aumentante. Pero $\overrightarrow{xy} \in E$, asi que PODRIA serlo, usando y, x como lado backward.

¿Porqué $s = x_0, x_1, ..., x_r = y, x$ no es un f-camino aumentante a pesar de que $s = x_0, x_1, ..., x_r = y$ si lo es y $\overrightarrow{xy}$ existe?

La única razón es que no podemos usarlo como lado backward por estar vacio, es decir, que $f(\overrightarrow{xy}) = 0$.

Esto es cierto para cualesquiera x, y que aparezcan en esa suma. Entonces:

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

Entonces hemos probado que para este S:

(1) $f(S,\overline{S}) = cap(S)$
(2) $f(\overline{S},S) = 0$

Por lo tanto:

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

Fin

4.3. **Correctitud del algoritmo de Ford-Fulkerson.**

4.3.1. *Corolario (de la prueba): Si el algoritmo de Ford-Fulkerson termina,entonces 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. Si vemos la demostración del MaxFlowMinCut vemos que el unico lugar en donde se usa que f es maximal es para demostrar que no hay f-caminos aumentantes. El resto de la prueba sigue de esto. Por lo tanto, si Ford-Fulkerson termina, termina con un flujo f para el cual vale toda la prueba a partir del momento en que demostramos que S es corte. Por lo tanto, para ese flujo f existe un corte S con $v(f) = cap(S)$ Por lo tanto f será maximal por 4.1.4 Fin

Observación: mientras se corre Ford-Fulkerson, durante la busqueda 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 servira como "certificado" de que f es maximal.

Pues dados f y S, se pueden calcular $v(f)$ y $cap(S)$ en forma independiente, y verificar si son iguales.

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.

4.4. **Teorema de la Integralidad.** Aún con las limitaciones que tiene Ford-Fulkerson, es suficiente para probar un teorema importante.

Un flujo f es "entero" si $f(\overrightarrow{xy})$ es un entero para todo lado.

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**.

Pero no necesariamente serán maximales entre **todos** los flujos. Pej si tenemos un sólo lado st de capacidad π entonces el flujo entero maximal tiene valor 3, pero

el flujo maximal tiene valor π. 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:

4.4.1. *Teorema de la integralidad. En un network con capacidades enteras, existen flujos maximales enteros.*

(por lo tanto todo flujo entero que sea maximal entre los flujos enteros es un flujo maximal entre todos los flujos).

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

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

Prueba: Este teorema demuestra el Teorema de la Integralidad porque ya vimos que si Ford-Fulkerson termina, entonces termina con un flujo maximal.

Para demostrar este teorema demostraremos que todos los flujos intermedios que se producen en el algoritmo de Ford-Fulkerson son enteros. Asi que obviamente el flujo final será entero, si es que Ford-Fulkerson termina. Pero ademas en la misma prueba veremos que debe terminar.

Como empezamos con el flujo $f = 0$, que es entero, basta probar que si f es entero y lo aumentamos con un camino aumentante de Ford-Fulkerson, entonces el flujo resultante sigue siendo entero.

Asumiendo f entero, tomemos entonces un f-camino aumentante $s = x_0, x_1, ..., x_r = t$.

Los ε_i que calculamos son:

(1) $\varepsilon_i = c(\overrightarrow{x_i x_{i+1}}) - f(\overrightarrow{x_i x_{i+1}})$ en los lados forward.
Como tanto c como f son enteros, entonces ese ε_i es entero en los casos de los lados forward.

(2) o bien $\varepsilon_i = f(\overrightarrow{x_{i+1} x_i})$ en los lados backward.
Como f es entero por hipotesis, entonces ε_i también es entero en los lados backward

Concluimos que todos los ε_i son enteros. 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 tambien será 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. Pero 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 un entero positivo, es decir, aumenta en AL MENOS UNO.

Como el valor de todo flujo es menor o igual que la capacidad de cualquier corte,(4.1.3) 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.