Flujos en Networks

Definition 1. *Un Grafo dirigido es un par* $G = (V, E)$ *donde* V *es un conjunto cualquiera (finito para nosotros) y* $E \subseteq V \times V$.

Definition 2. *Denotaremos el lado* (x, y) *como* $\overrightarrow{xy}$.

Si g *es una función definida en los lados y* A *y* B *son subconjuntos de vertices, entonces* $g(A, B)$ *denotará la suma:*

$$g(A,B) = \sum_{\substack{x \in A \\ y \in B \\ \overrightarrow{xy} \in E}} g(\overrightarrow{xy})$$

Dado un vertice x *denotaremos:*

$$\begin{aligned}
\Gamma^+(x) &= \{y \in V | \overrightarrow{xy} \in E\} \\
\Gamma^-(x) &= \{y \in V | \overrightarrow{yx} \in E\} \\
in_g(x) = g(V, \{x\}) = g(\Gamma^-(x), \{x\}) &= \sum_{y \in \Gamma^-(x)} g(\overrightarrow{yx}) \\
out_g(x) = g(\{x\}, V) = g(\{x\}, \Gamma^+(x)) &= \sum_{y \in \Gamma^+(x)} g(\overrightarrow{xy})
\end{aligned}$$

Definition 3. *Un Network será un grafo dirigido con pesos positivos en los lados, es decir, un triple* (V, E, c) *donde* (V, E) *es un grafo dirigido y* $c : E \mapsto I\!\!R_{>0}$.

En este contexto, $c(\overrightarrow{xy})$ *se llamara la "capacidad" del lado* $\overrightarrow{xy}$.

La idea basica es que un network es como una red de cañerias, donde los pesos c es por ejemplo, cuanta agua puede llevar la cañeria. En este contexto, no querremos minimizar un costo, sino maximixar la cantidad de "agua" que se pueda transportar por la cañeria.

Definition 4. *Dado un network* (V, E, c), *y un par de vertices* $s, t \in V$, *un flujo de* s *a* t *es una funcion* $f : E \mapsto I\!\!R$ *con las siguientes propiedades:*

1. $0 \leq f(\overrightarrow{xy}) \leq c(\overrightarrow{xy}) \quad \forall \overrightarrow{xy} \in E$.
2. $in_f(x) = out_f(x) \forall x \in V - \{s, t\}$.
3. $out_f(s) \geq in_f(s)$.

El valor del flujo es $v(f) = out_f(s) - in_f(s)$

Theorem 1. $v(f) = in_f(t) - out_f(t)$.

Prueba:

$$\begin{aligned}
f(V,V) &= \sum_{\substack{x \in V \\ y \in V \\ \overrightarrow{xy} \in E}} f(\overrightarrow{xy}) \\
&= \sum_{x\in V} \sum_{y\in \Gamma^+(x)} f(\overrightarrow{xy}) \\
&= \sum_{x\in V} out_f(x)
\end{aligned}$$

Por otro lado, tambien podemos escribir:

$$\begin{aligned}
f(V,V) &= \sum_{\substack{y \in V \\ x \in V \\ \overrightarrow{yx} \in E}} f(\overrightarrow{yx}) \\
&= \sum_{x\in V} \sum_{y\in \Gamma^-(x)} f(\overrightarrow{yx}) \\
&= \sum_{x\in V} in_f(x)
\end{aligned}$$

Por lo que concluimos que $\sum_{x\in V} out_f(x) = \sum_{x\in V} in_f(x) \quad (\star)$. Por la propiedad 2) de flujos, $out_f(x) = in_f(x)$ para todo $x \neq s, t$, por lo que en $\star$, se cancelan todos los terminos menos los correspondientes a s y t. Con lo cual nos queda la igualdad $out_f(s) + out_f(t) = in_f(s) + in_f(t)$, es decir, $in_f(t) - out_f(t) = out_f(s) - in_f(s) = v(f) \geq 0$. $\square$

Definition 5. *Dado un network N y vertices s, t, un flujo maximal de s a t (o "Max flow") es un flujo f de s a t tal que $v(g) \leq v(f)$ para todo flujo g de s a t.*

El flujo maximal puede no ser único: Sea N el network dado por vertices s, A, B, C, t, y lados $\overrightarrow{sA}, \overrightarrow{AB}, \overrightarrow{AC}, \overrightarrow{Bt}, \overrightarrow{Ct}$, todos con capacidades iguales a 10. Entonces es claro que tanto el flujo que vale 10 en $\overrightarrow{sA}, \overrightarrow{AB}, \overrightarrow{Bt}$ y cero en lo otros lados, com el que vale 10 en $\overrightarrow{sA}, \overrightarrow{Ac}, \overrightarrow{Ct}$ y cero en los otros, son ambos flujos maximales.

Si bien puede no ser único, un mayor problema es que no es claro de las definiciones que haya al menos uno.

Si agregamos la condición de que las capacidades y el flujo en cada lado deben ser naturales, (esto se llama un flujo "entero"), entonces, como hay una cantidad finita de flujos enteros, es claro que existe un flujo entero maximal, i.e., un flujo que es maximal entre todos los flujos enteros. Lo que no es claro es que el flujo entero maximal sea maximal entre TODOS los flujos. En algunos libros (p.ej. el Biggs) esto se ignora, porque facilita las pruebas.

Ademas del problema de si existe o no flujo maximal, el problema que tenemos es, aun si existe uno, ¿como hallarlo? (con un algoritmo polinomial).

¿como construir un flujo, no hablemos ya de un flujo maximal? Si colocamos numeros al azar en los lados, lo mas probable es que la función resultante NO sea unflujo. Ahora bien, la funcion que vale 0 en todos los lados es claramente un flujo. Si tomamos un camino (dirigido) desde s a t e incrementamos el valor del flujo a lo largo de ese camino de forma tal que en ninguno de los lados superemos la capacidad, lo que nos queda seguirá siendo un flujo. Por lo tanto, tenemos el siguiente algoritmo, basado en una estrategia Greedy, que parece a primera vista bueno.

Algoritmo Greedy para hallar flujo maximal

1. $f := 0$.

2. Buscar un camino dirigido $s = x_0, x_1, ..., x_r = t$, con $\overrightarrow{x_i x_{i+1}} \in E$ tal que $f(\overrightarrow{x_i x_{i+1}}) < c(\overrightarrow{x_i x_{i+1}})$ para todo i.

3. Aumentar el flujo a lo largo del camino de 2. en ϵ unidades, donde

$$\epsilon = \min\{c(\overrightarrow{x_i x_{i+1}} - f(x_i x_{i+1})\}$$

4. Repetir 2. hasta que no se puedan hallar mas caminos.

Veamos un ejemplo: en el network dado en el ejemplo en el cual el flujo no era único, siguiendo este algoritmo, hallamos el camino $sABt$, aumentamos en 10 a lo largo de este camino, y se acaba el algoritmo, porque ya no podemos hallar ningun otro camino. En este caso, el greedy encontró un flujo maximal, pero hagamos otro ejemplo:

Sea N el network con:

vertices s, A, B, C, D, E, F, t,

lados $\overrightarrow{sA}, \overrightarrow{sC}, \overrightarrow{sE}, \overrightarrow{AB}, \overrightarrow{AF}, \overrightarrow{CB}, \overrightarrow{CD}, \overrightarrow{ED}, \overrightarrow{EF}, \overrightarrow{Bt}, \overrightarrow{Dt}, \overrightarrow{Ft}$,

todos de capacidad 10, excepto los lados $\overrightarrow{sE}$ y $\overrightarrow{Ft}$ de capacidad 20. Entonces, Greedy obtiene lo siguiente (notación : por $sx_1...x_{r-1}t : \epsilon$ indicaremos el paso de hallar el camino y aumentar el flujo por ϵ).

$$sABt : 10$$
$$sCDt : 10$$
$$sEFt : 10$$

y luego no puede hacer mas, porque si busca un camino, solo puede salir desde s a E, luego ir a D, y ahi no puede seguir mas. Por lo tanto, Greedy obtiene un flujo de valor igual a 30.

Pero Greedy podria haber hecho tambien la siguiente corrida:

$$sAFt : 10$$
$$sCBt : 10$$
$$sEFt : 10$$
$$sEDt : 10$$

y tenemos un flujo de valor 40. Lo cual muestra que no siempre Greedy obtiene un flujo de valor maximal. (mas adelante veremos que este ultimo flujo si es maximal).

En principio entonces, el Greedy de flujos se comporta como el Greedy de coloreo: es un algoritmo polinomial (ejercicio: probar que la complejidad de Greedy es $O(m^2)$) pero que no siempre nos provee con un flujo maximal. Sin embargo, hay una diferencia fundamental con coloreo. En flujos, veremos que hay un "certificado" que nos permite comprobar si un flujo es o no maximal. Este certificado permite hacerle una corrección a Greedy, obteniendo otro algoritmo que sigue siendo polinomial pero si obtiene un flujo maximal. Basicamente, el algoritmo chequea su flujo parcial contra el certificado, y si todavia no obtuvo un flujo maximal, el mismo certificado le sugiere que cosas tiene que cambiar en el flujo para obtener uno mas grande.

Ese certificado es el concepto de "corte":

Definition 6. *Un Corte es un subconjunto de los vertices que tiene a s pero no tiene a t.*

Por ejemplo, $S = \{s\}$ es un corte. Tambien es corte el conjunto $S = V - \{t\}$.

Definiremos la *capacidad* de un corte como : $cap(S) = c(S, \overline{S})$, donde $\overline{S} = V - S$. Un corte es MINIMAL si su capacidad es la menor de las capacidades de todos los cortes. Cortes minimales existen porque hay una cantidad finita de cortes (ejercicio: probar que hay 2^{n-2} cortes).

Theorem 2. *Sea f flujo y S corte.*

1. $v(f) = f(S, \overline{S}) - f(\overline{S}, S)$.
2. $v(f) \leq cap(S)$.
3. *Si $v(f) = cap(S)$, entonces f es maximal y S es minimal.*

Prueba: (1): Puesto que $out_f(s) - in_f(s) = v(f)$, pero $out_f(x) - in_f(x) = 0$ para todo otro $x \in S$ (pues al ser S corte, $t \notin S$), entonces tenemos que:

$$\begin{aligned}
v(f) &= \sum_{x \in S} out_f(x) - in_f(x) \\
&= \sum_{x \in S} \sum_{y \in \Gamma^+(x)} f(\overrightarrow{xy}) - \sum_{y \in \Gamma^-(x)} f(\overrightarrow{yx}) \\
&= \sum_{\substack{x \in S \\ y \in S \\ \overrightarrow{xy} \in E}} f(\overrightarrow{xy}) + \sum_{\substack{x \in S \\ y \notin S \\ \overrightarrow{xy} \in E}} f(\overrightarrow{xy}) - \sum_{\substack{x \in S \\ y \in S \\ \overrightarrow{yx} \in E}} f(\overrightarrow{yx}) - \sum_{\substack{x \in S \\ y \notin S \\ \overrightarrow{yx} \in E}} f(\overrightarrow{yx}) \\
&= f(S, S) + f(S, \overline{S}) - f(S, S) - f(\overline{S}, S) \\
&= f(S, \overline{S}) - f(\overline{S}, S)
\end{aligned}$$

(2): Observemos que $0 \leq f(A, B) \leq c(A, B)$ para cualesquiera subconjuntos A y B. Por lo tanto:

$$\begin{aligned}
v(f) &= f(S, \overline{S}) - f(\overline{S}, S) \\
&\leq f(S, \overline{S}) \qquad (\text{pues } f(\overline{S}, S) \geq 0) \\
&\leq c(S, \overline{S}) = cap(S)
\end{aligned}$$

(3): Sea g otro flujo. Entonces, por (2), aplicado a g y S, tenemos $v(g) \leq cap(S) = v(f)$ y por lo tanto f es maximal.

Sea T otro corte. Por (2) aplicado a T y f, tenemos: $cap(T) \geq v(f) = cap(S)$ y por lo tanto S es minimal. □

La calse que viene veremos como usar este concepto para mejorar greedy. Por ahora, veamos que efectivamente siempre hay flujos maximales:

Theorem 3. *En todo network, siempre existe un flujo maximal.*

Prueba: Sea A el conjunto $\{v(f) : f \text{ es un flujo}\}$.

Como $\{s\}$ es corte, tenemos que $v(f) \leq cap(\{s\})$ para todo f , es decir, A es un conjunto acotado superiormente. Por lo tanto, tiene supremo. Sea $a = SupA$. Queremos probar que a es en realidad un maximo.

Por ser supremo, existe una sucesión de flujos $\{f_i\}_{i=1}^{\infty}$ tal que

$$\lim_{i\to\infty} v(f_i) = a \qquad (0)$$

Ordenemos los lados arbitrariamente $e_1, ..., e_m$. Como $f_i(e_1) \in [0, c(e_1)]$, tenemos una sucesion acotada. Por el teorema de Bolzano-Weierstrass, esa sucesión tiene una subsucesión convergente. Es decir, existe una subsucesión $\{f_{i_j}\}_{j=1}^{\infty}$ tal que existe $\lim_{j\to\infty} f_{i_j}(e_1)$. Llamemos ℓ_1 a ese limite. Es decir:

$$\ell_1 = \lim_{j\to\infty} f_{i_j}(e_1) \qquad (1)$$

Podemos repetir el argumento con la sucesión $f_{i_j}(e_2)$, obteniendo una sub-sub-sucesión $f_{i_{j_k}}$ tal que existe el limite de $f_{i_{j_k}}(e_2)$, es decir, existe ℓ_2 con:

$$\ell_2 = \lim_{k\to\infty} f_{i_{j_k}}(e_2) \qquad (2)$$

Por ser subsubsucesion, tambien vale que $\ell_1 = \lim_{k\to\infty} f_{i_{j_k}}(e_1)$ y que $\lim_{k\to\infty} v(f_{i_{j_k}}) = a$.

Podemos seguir asi, obteniendo sub-sub-sub....sucesiones sucesivas, lado por lado, hasta llegar a una sub-sub-....sucesion $f_{i_{j_{k_{\ddots_z}}}}$, que por comodidad denotaremos por g_z, y numeros ℓ_r tal que $\lim_{z\to\infty} g_z(e_r) = \ell_r$ para todo r y tal que $\lim_{z\to\infty} v(g_z) = a$. Definiendo ahora $f(e_r) = \ell_r$, es facil ver que f es flujo y que $v(f) = a \geq v(g)$ para todo flujo g. Por lo tanto, f es flujo maximal. □