Definiciones y lemas elementales

Definición 1.

- Una **red de flujos** es un grafo $G = (V, E)$ orientado y conexo que tiene dos nodos distinguidos s *fuente* con $d_{\text{out}(s)} \geq 0$ y t *sumidero* con $d_{\text{in}(t)} \geq 0$
- Una función de capacidad $c : E \to \mathbb{R}_{\geq 0}$

Definición 2.

Dada una red $G = (V, E)$ con una función de capacidad c, fuente s, sumidero t:

- Un **flujo factible** es una función $f : E \to R_{\geq 0}$ que verifica:
- ‣ **Capacidad**: $\forall (u, v) \in E :: 0 \leq f(u, v) \leq c(u, v)$
- ‣ **Conservación de flujo**: $\forall v \in V \setminus \{s, t\}$:

$$\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$$

$\forall (u, v) \in E \underset{L}{\Rightarrow} (v, u) \notin E$ ó $c(v, u) = 0$

- El **valor del flujo** es $\forall u, v \in V \setminus \{s, t\}$:

$$F = |f| = \sum_{v \in V} f(s, v) = \sum_{u \in V} f(u, t)$$

Lema 1.

Sea $G = (V, E)$ una red con función de capacidad c, fuente s, sumidero t.

$$F = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

Problema de flujo máximo

Definición 3.

El **problema de flujo máximo** consiste en, dada una red $G = (V, E)$ con capacidades en las aristas, una *fuente* s, y un *sumidero* t, determinar el flujo factible de valor máximo F que se puede enviar de s a t.

Definición 4.

Un **corte** en la red $G = (V, E)$ es un subconjunto $S \subseteq V \setminus \{t\}$ tal que $s \in S$.

Notación: Dados $S, T \subseteq V$, llamamos $ST = \{(u, v) \in E \mid u \in S, v \in T\}$

Proposición 1.

Sea f un flujo factible definido en una red G con un valor F y sea S un corte, luego:

$$F = \sum_{(u,v) \in S\overline{S}} f(u, v) = \sum_{(u,v) \in \overline{S}S} f(u, v)$$

Donde $\overline{S} = V \setminus S$

Definición 5.

Dada una red $G = (V, E)$ con función de capacidad c, la capacidad de un corte S se define como

$$c(S) = \sum_{e \in S\overline{S}} c(e)$$

Definición 6.

El **problema del corte mínimo** consiste en, dada una red $G = (V, E)$ con función de capacidades en las aristas c, determinar un corte de capacidad mínima. Es decir, encontrar S corte de G tal que:

$$c(S) = \min\left\{c\left(\overline{S}\right) \mid \overline{S} \text{ es un corte de } N\right\}$$

Lema 2.

Dados una red $G = (V, E)$ con una función de capacidad c, una función de flujo factible con valor F y un corte S, vale que:

$$F \leq c(S)$$

Corolario 1.

Si F es el valor de un flujo factible f, y S un corte de una red G tal que $F = c(S)$

- f se define como flujo máximo
- S es un corte mínimo

Caminos de aumento

Definición 7.

Dada una red $G = (V, E)$ con una función de capacidad c y un flujo factible f

- Definimos la **red residual**, $R(G, f) = (V, E_R)$ donde $\forall (v \to w) \in E$
- ‣ $(v, w) \in E_R$ si $f(v, w) < c(v, w)$
- ‣ $(w, v) \in E_R$ si $f(v, w) > 0$

Algoritmo 1.

Dada una red $G = (V, E)$, una función de flujo f, una red residual $R = (G, f) = (V, E_R)$, **CaminoDeAumento** devuelve P camino de aumento ó S corte (que será mínimo).

procedure CaminoDeAumento(G, f, R)
    $S \leftarrow \{s\}$
    while $t \notin S$ y $\exists (v \to w) \in E_R$ y $v \in S$ y $w \notin S$ **do**
        $\text{pred}[w] \leftarrow v$
    end
    if $t \notin S$ **then**
        return S corte de G
    end
    else
        reconstruir P entre s y t usando **pred** a partir de t
        return P camino de aumento
    end
end

Proposición 2.

Dada una red $G = (V, E)$, un flujo factible f y su red residual $R(G, f) = (V, E_R)$, el algoritmo de camino de aumento determina un camino de aumento en $R(G, f)$, P, si existe y, caso contrario, determina un corte S de G.

Definición 8.

Dada una red $G = (V, E)$ con una función de capacidad c, un flujo factible f y un camino de aumento P en $R(G, f)$:

- Para cada arista $(v \to w)$ de P, definimos:

$$\Delta(v \to w) = \begin{cases} c(v \to w) - f(v \to w) & \text{si } (v \to w) \in E \\ f(v \to w) & \text{si } (w \to v) \in E \end{cases}$$

- $\Delta(P) = \min_{e \in P}\{\Delta(e)\}$

Proposición 3.

Sea f un flujo factible definido sobre una red G con un valor F y sea P un camino de aumento en $R(G, f)$. Entonces el flujo f, definido por

$$\overline{f}(v \to w) = \begin{cases} f(v \to w) & \text{si } (v \to w) \notin P \\ f(v \to w) + \Delta(P) & \text{si } (v \to w) \in P \\ f(v \to w) - \Delta(P) & \text{si } (w \to v) \in P \end{cases}$$

es un flujo factible sobre G con valor $\overline{F} = F + \Delta(P)$.

Teorema 1.

Sea f un flujo factible definido sobre una red $G = (V, E)$. f es un flujo factible de valor máximo $\Longleftrightarrow$ no existe camino de aumento en $R(G, f)$.

Teorema 2.

Dada una red $G = (V, E)$, el valor del flujo máximo es igual a la capacidad del corte mínimo.

Algoritmo de Ford-Furkerson

El algoritmo toma $G = (V, E)$ una red, c una función de capacidad y devuelve f flujo máximo.

procedure FORD&FURKERSON(G)
    ▷ definir un flujo inicial en G (ejemplo, $f(e) \leftarrow 0, \forall e \in E$)
    while existe P camino de aumento en $R(G, f)$ **do**
        for cada arista $(v \to w)$ de P **do**
            if $(v \to w) \in E$ **then**
                $f(v \to w) \leftarrow f(v \to w) + \Delta(P)$
            end
            else
                $f(w \to v) \leftarrow f(w \to v) - \Delta(P)$
            end
        end
    end
end

Proposición 4.

Si las capacidades de las aristas de $G = (V, E)$ son enteras, el problema de flujo máximo tiene una función de flujo máximo f entera, $f(e) \in Z_{\geq 0}, \forall e \in E$.

Proposición 5.
Si los valores del flujo inicial y las capacidades de las aristas de la red son enteras, el método de F&F realiza a lo sumo F iteraciones, siendo F el valor del flujo máximo.

Proposición 6.
Si los valores del flujo inicial y las capacidades de las aristas de la red son enteras, F&F es $O(nmU)$, con U una cota superior finita para el valor de las capacidades.

Edmond-Karp

La implementación del paradigma planteado por Ford y Furkerson usa BFS en el algoritmo de camino de aumento para marcar los nodos. Su complejidad es $O(m^2 n)$

Teorema 3.
Dada una red $G = (V, E)$ con n nodos y m aristas, el algoritmo de E&K realiza a lo sumo nm iteraciones.

Lema 3.
Sea f_i el flujo máximo de la iteración i del algoritmo E&K. Entonces, para todo $v \in V$, $d_{R(G,f_i)}(s, v) \leq d_{R(G,f_{i+1})}(s, v)$ para toda iteración i del algoritmo.

Útiles + variantes del problema de flujo máximo

Múltiples fuentes y sumideros:

Para una red $G = (V, E)$

Un problema de flujo máximo puede tener múltiples fuentes y sumideros. Ejemplo, tenemos $s_1, ..., s_p$ nodos *fuente* $t_1, ..., t_q$ y nodos *sumidero*.

Podemos plantear sumideros y arcos ficticios con capacidad infinita, o sea:

$\forall i :: 1 \leq i \leq p \underset{L}{\rightarrow} (s, s_i) \in E, c(s, s_i) = \infty$

$\forall i :: 1 \leq j \leq q \underset{L}{\rightarrow} (t_j, t) \in E, c(t_j, t) = \infty$

Con s el nuevo nodo *fuente* y t el nuevo nodo *sumidero*.

La fuente ficticia s generará tanto flujo como requieran las fuentes s_i y el sumidero ficticio t consumirá tanto flujo como lo hagan los sumideros t_j.

Matching máximo en grafos bipartitos:

Dado el grafo bipartito $G = (V_1 \cup V_2, E)$, definimos la red $M = (V', E')$:

- $V' = V_1 \cup V_2 \cup \{s, t\}$, con s, t nodos ficticios representando la *fuente* y el *sumidero* de M.
- $E' = \{(u \rightarrow w), u \in V_1, w \in V_2, (u, w) \in E\} \cup \{(s, u), u \in V_1\} \cup \{(w, t), w \in V_2\}$
- $c(e) = 1, \forall e \in E'$.

El cardinal del matching máximo de G será igual al valor del flujo máximo en la red M