MA4702. Algoritmos Combinatoriales. 2013.
Profesor: José Soto
Escriba(s): Camilo Rojas , Cristóbal Rojas y Francisco Venegas.
Fecha: Lunes 20 de Octubre, 2014.

Cátedra 20

1. Teorema de Descomposición de flujos

Teorema 1. *(Descomposición de flujos) Todo (s,t)-flujo se puede descomponer en ciclos y (s,t)-caminos.*
Sea (G,u,s,t) una red y f flujo en ésta red (factible o no). Sean

$\mathcal{P}_{s\to t}$ el conjunto de todos los (s,t)-caminos en G.

$\mathcal{P}_{t\to s}$ el conjunto de todos los (t,s)-caminos en G.

$\mathcal{C}$ el conjunto de todos los ciclos en G.

Si valor$(f)\geq 0$, podemos escribir

$$f=\sum_{p\in\mathcal{P}_{s\to t}}\lambda_p\chi^p+\sum_{c\in\mathcal{C}}\lambda_c\chi^p$$

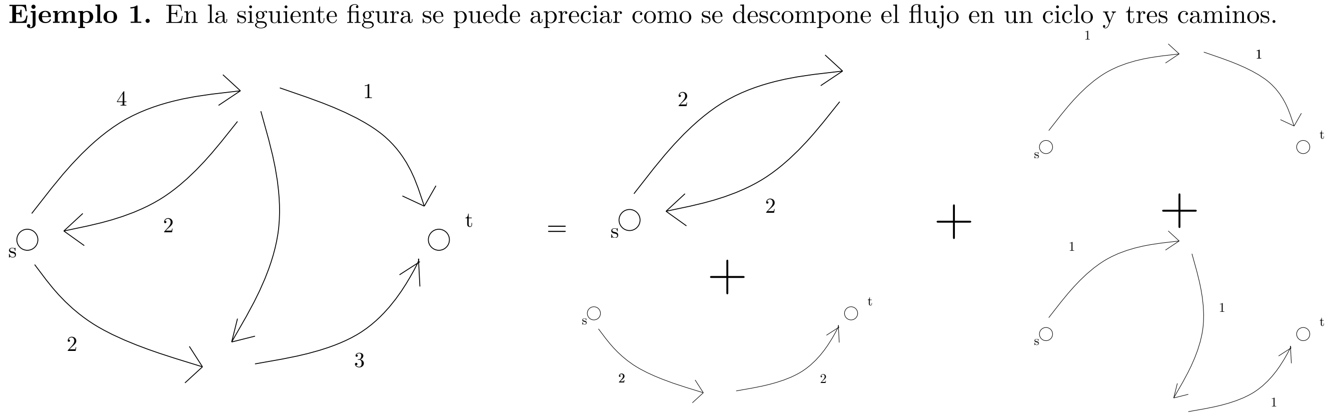
Si valor(f)<0, podemos escribir

$$f=\sum_{p\in\mathcal{P}_{t\to s}}\lambda_p\chi^p+\sum_{c\in\mathcal{C}}\lambda_c\chi^p$$

De hecho, todos los coeficientes $\lambda\geq 0$ y el número de coeficientes no nulos son a lo más $|E(G)|$.

Observación 1. Para cada λ_p, se tiene que $0\geq\lambda_p\chi^p\leq f$. Por lo que se verifica que si f es factible, entonces $\lambda_p\chi^p$.

Ejemplo 1. En la siguiente figura se puede apreciar como se descompone el flujo en un ciclo y tres caminos.



Demostración. Definamos para todo vector $x\in\mathbb{R}^E$, el soporte de x como $\text{Sop}(x)=\{e\in E:\chi_e>0\}$

Definiremos iterativamente flujos factibles f^i en la red (G,u,s,t), y grafos $G^i=(V,\text{Sop}(f^i))$, donde f^0 se define como f. Para ello repetiremos el siguiente proceso:

Si $G^i=(V,\text{Sop}(f^i))$ contiene un ciclo $\mathcal{C}=\mathcal{C}_i$. Definimos $f^{i+1}=f^i-f^i_c\chi^c$, con $f^i_c=\min\limits_{e\in\mathcal{C}}f^i_e$ así f^{i+1} es un flujo en (G,u,s,t) y valor$(f^{i+1})=$ valor (f^i).

Notemos que $\text{Sop}f^{i+1}\subsetneq\text{Sop}f^i$ pues para $e\in\mathcal{C}$ pues al definir f^{i+1} anulamos el flujo de la arista $e*$ correspondiente a $\min\limits_{e\in\mathcal{C}}$ luego esta arista sale del soporte, ademàs se cumple que valor$(f^i_e)=$valor$(f^{i+1}_e)=0$.
Repetiremos este proceso creando flujos $f^0,f^1,\cdots,f^k$ hasta que el grafo $G^k=(V,\text{Sop}(f))$ sea áciclico.
Veamos que se verifica que:
$(f\geq 0)$ si valor$(f^{\hat{\imath}})\geq 0$. Entonces existe un $s-t$ camino en $\text{Sop}(f^i)$, el caso $f<0$ es análogo
En Efecto,
Si hacemos BFS desde s en $\text{Sop}(f^i)$ para calcular el conjunto S de los nodos alcanzables desde s obtenemos:

si $t \in S \to$ BFS encuentra un $x - t$ camino.

si $t \notin S \to S$ es un $s - t$ corte.

$0 <$valor$(f^i) = f^i(\delta^+(S)) - f^i(\delta^-(S)) \leq f^i(\delta^+(S))$

Luego existe un arco $e = uv \in \delta^+(S)$ con $f^i(e) > 0$ entonces $e \in \text{Sop}(f^i)$ lo que es una contradicción pues $u \in S, v \in S$. Con este resultado podemos tomar un P un s-t camino en $G^{\hat{\imath}}$ y definir:

$$f^{i+1} = f^i - f^i_P \chi^P$$
$$\text{Sop}(f^{i+1}) \subsetneq \text{Sop}(f^i)$$
$$f^i_P = \min_{e \in P} f^i_e$$

donde se cumplira que $0 \leq$ valor(f^{i+1}) =valor $(f^i) - f^{\hat{\imath}}_P$. De otra manera se podría haber encontrado un ciclo en $\text{Sop}(f^i)$. Repetiremos este proceso hasta que $\text{Sop}(f^i) = \emptyset$. Finalmente se obtendra:

$$f = \sum_{i=0}^{k-1} f^i_{c_i} \chi^{c_i} + \sum_{i=k}^{l} f^i_{P_i} \chi^{P_i}$$

■

Observación 2. Todo (s,t)-flujo factible de valor positivo es combinación cónica de indicatrices de (s,t)-caminos y de ciclos.

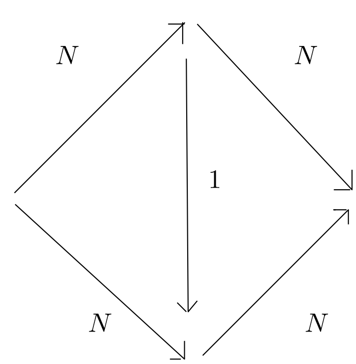
Corolario 1. *Si f es (s,t)-flujo factible en (G,u,s,t) (de valor positivo), entonces existe un (s,t)-camino en esta red con capacidad mayor o igual que* $\dfrac{valor(f)}{m}$.

Demostración. Por el teorema, sabemos que se cumple la siguiente igualdad $f = \sum \lambda_p \chi^p + \sum \lambda_c \chi^c$. Esto implica que $valor(f) = \sum \lambda_p$. Como la combinación es cónica y hay a lo más m términos no nulos, existe un (s,t)-camino p con $\lambda_p >$ valor $(f)/m$.

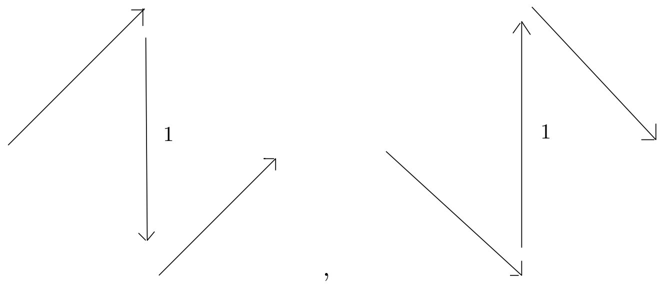
Algorithm 1 Recuerdo Algoritmo de Ford-Fulkerson.

```
Dado (G, u, s, t) una red.
f ← ∅.
while Existe un camino aumentante P en G_t^f do
    Aumentar f con P(f ← f + χ̄^p u_p^f).
end while
Return f.
```

Ejemplo 2. **Malo para FF**. Sea $u = N \in \mathbb{N}$, con $N \gg 0$.



FF podría elegir mal y partir con



Repitiendo esto, FF toma 2N iteraciones para llegar al óptimo. Esto NO es polinomial, pues la entrada se codifica en $\mathcal{O}(\ln(N))$ bits.

2. Algoritmo de Edmonds-Karp 1

Algorithm 2 E-K 1

Aumentar por el camino con mayor capacidad residual.

La clase anterior se probó que si f es factible en (G, u, s, t) y g es factible en (G^f, u^f, s, t) entonces $f + \bar{g}$ es factible en (G, u, s, t), con $\bar{g}_e = g_e + g_{e^{\leftarrow}}$ donde $\bar{g} : E \longrightarrow \mathbb{R}$ y $g : E \cup E^{\leftarrow} \longrightarrow \mathbb{R}$. Recordar que el dominio de g es $E \cup E^{\leftarrow}$, mientras que el dominio de $\bar{g}$ es E Más aún, este lema tiene una recíproca:

Lema 1. *Si f y h son factibles entonces $h - f = \bar{g}$ para cierto g factible en (G^f, u^f, s, t).*

Demostración. Defina $g : E \cup E^{\leftarrow} \longmapsto \mathbb{R}$ del siguiente modo: Para $e \in E$

si $h_e = f_e$

$g_e = h_e - f_e$

si $f_e > h_e$

$g_e = 0$

$g_{e^{\leftarrow}} = f_e - h_e$

Del lema anterior se deduce que: Si f es factible en (G, u, s, t) y f^* es el flujo máximo en (G, u, s, t) entonces existe g flujo en (G^f, u^f, s, t) con $valor(g) = valor(f^*) - valor(f)$.

Por el corolario anterior, sabemos que existe s-t camino residual en (G^f, u^f, s, t) de capacidad residual al menos $\frac{valor(g)}{2m} = \frac{valor(f^*) - valor(f)}{2m}$.

Teorema 2. *Sea $u : E \longrightarrow \mathbb{Z} \cup \infty$ y no existe s-t camino de capacidad infinita en (G, u, s, t). Entonces Edmonds-Karp 1 termina en $\mathcal{O}(n \log(\alpha^*) = \mathcal{O}(m + \log(m \max_{e \in E : u_e < \infty} u_e))$ iteraciones, por lo cual su complejidad es $\mathcal{O}((m + n \log(n)(m \log(\alpha^*))$*

Demostración. sea f^i el flujo en la i-ésima iteración. ($f^0 = 0_E$) Sea α^* el valor de un flujo máximo y α_i el valor de un flujo f_i. La observación anterior nos dice que en la i-ésima iteración se encuentra un camino aumentante de capacidad residual al menos $\frac{\alpha^* - \alpha_i}{2m}$.

Luego $\alpha_{i+1} \geq \alpha_i + \frac{\alpha^* - \alpha_i}{2m}$.

Entonces $\alpha^* - \alpha_{i+1} \leq \alpha^* - \alpha_i - \frac{\alpha^* - \alpha_i}{2m} = (\alpha^* - \alpha_i)\left(1 - \dfrac{1}{2m}\right)$

Después de k iteraciones:

$$\begin{aligned}
\alpha^* - \alpha_k &\leq (\alpha^* - \alpha_{k-1})(1 - \frac{1}{2m}) \\
&\leq (\alpha^* - \alpha_{k-2})(1 - \frac{1}{2m})^2 \\
&\vdots \\
&\leq (\alpha^* - \alpha_0)(1 - \frac{1}{2m})^k \\
&= \alpha^*(1 - \frac{1}{2m})^k \\
\text{Observación: } & \text{ Si } k > 2m\ln(\alpha^*) \\
\alpha^* - \alpha_k &\leq \alpha^*(1 - \frac{1}{2m})^{2m\ln(\alpha^*)} \\
&\leq \alpha^* \exp\left(\frac{-2m\ln\alpha^*}{2m}\right) = \frac{\alpha^*}{\alpha^*} = 1
\end{aligned}$$

Usamos que $(1-x) \leq \exp(-x)$

Es decir, después de $k = \mathcal{O}(n\ln(\alpha))$ iteraciones $\alpha - \alpha_k < 1$. Se concluye que $\alpha_k = \alpha$, con lo que se llega al óptimo.

Complejidad

Hay $\mathcal{O}\left(n\ln(\alpha^*)\right)$ iteraciones. En cada una se debe encontrar el camino con mayor cuello de botella de s a t en el grafo residual (G^f, u^f, s, t).
Esto se puede hacer modificando Dijkstra en tiempo $\mathcal{O}(m + n\log(n))$.

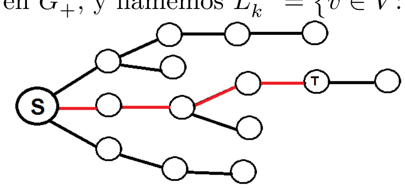
Observación:
Esta versión de Edmonds Karp (1) es débilmente polinomial, pues su complejidad depende del valor del número de datos.

3. Algoritmo de Edmonds-Karp 2

Algorithm 3 E-K 2

Edmonds y Karp dieron un segundo algoritmo, que es fuertemente polinomial.
Aumentar por el camino P más corto (según el # de arcos)

Demostremos que $E-K(2)$ es fuertemente polinomial; mas aún, usa a lo más $O(nm)$.
Sea f^i el flujo a la i-ésima iteraciones, y $G^i = (G^{f^i}, u^{f^i}, s, t)$
Usamos BFS en G^i_+, y llamemos $L^{(i)}_k = \left\{v \in V\text{: el } s \text{ - } v \text{ camino mínimo en } G^i_+ \text{ usa } k\text{arcos } \right\}$



La próxima clase demostraremos el siguiente lema.

Lema 2. *Si $v \in L^{(i)}_k$, entonces $\forall j \geq i, v \in L^{(i)}_{k'}$, con $k' \geq k$*