MA3705. Algoritmos Combinatoriales. 2014.
Profesor: José Soto
Escribas: Christopher Cabezas, Catalina Pesce y Valentina Toro
Fecha: 13 de Octubre 2014.

Cátedra 18

1. Recuerdo de la clase pasada

Volveremos a presentar los resultados de la clase anterior para plantear un algoritmo que calcule el conjunto independiente común de cardinalidad máxima entre las matroides $M_1 = (S, \mathcal{I}_1)$ y $M_2 = (S, \mathcal{I}_2)$ con $I = \mathcal{I}_1 \cap \mathcal{I}_2$.

Lema 1. *Sean $X_1, X_2 \subseteq S \setminus I$, con $X_1 \cap X_2 = \emptyset$. Si no existe X_1-X_2 camino en $D = D(M_1, M_2, I)$, entonces I es un conjunto independiente común y de hecho:*

$$X = \{x \in S : \exists\, x\text{-}X_2 \textit{ camino en } D\}$$

satisface:

$$|I| = r_1(X) + r_2(S \setminus X).$$

Lema 2. *Sea P un X_1-X_2 camino en D (si es que existe) con el menor número de arcos posibles, entonces*

$$I\Delta V(P) \in \mathcal{I}_1 \cap \mathcal{I}_2, \quad \textit{con } |I\Delta V(P)| = |I| + 1.$$

2. Sobre intersección de matroides

A partir de los resultados anteriores, se obtiene el siguiente teorema.

Teorema 1. *Sean $M_1 = (S, \mathcal{I}_1)$ y $M_2 = (S, \mathcal{I}_2)$ matroides, $r_1(\cdot)$ el rango con respecto a M_1 y $r_2(\cdot)$ el rango con respecto a M_2. Entonces*

$$\max_{J \in \mathcal{I}_1 \cap \mathcal{I}_2} |J| = \min_{X \subseteq S} \left[r_1(X) + r_2(S \setminus X) \right].$$

Ejemplo 1. *Veamos como se interpreta este teorema en un grafo bipartito. Sea $G = G(L, R, E)$ grafo, $M_1 = (E, \mathcal{I}_1)$ y $M_2 = (E, \mathcal{I}_2)$ con $\mathcal{I}_1 = \{F \subseteq E : |F \cap \delta(F)| \leq 1, \forall\, v \in L\}$ e $\mathcal{I}_2 = \{F \subseteq E : |F \cap \delta(F)| \leq 1, \forall\, v \in R\}$.*

Notemos que M es un matching en G si y solo si $M \in \mathcal{I}_1 \cap \mathcal{I}_2$, es decir, si M tiene a lo más una arista incidente en v, para cada v en L y R.

Tenemos que

$$\max_{M \textit{ matching}} |M| = \min_{X \subseteq E} [r_1(X) + r_2(E \setminus X)].$$

Analicemos lo que representa $r_1(X)$:

$$\begin{aligned} r_1(X) &= \max_{\substack{J \in \mathcal{I}_1 \\ J \subseteq X}} |J| \\ &= \#\textit{ de vértices de } L \textit{ que son tocados por alguna arista de } X \\ &= \min \#\textit{ de vértices de } L \textit{ necesarios para cubrir } X. \end{aligned}$$

Luego,

$$\begin{aligned} r_1(X) + r_2(S \setminus X) &= \min \#\begin{array}{l}\textit{de vértices de } L \\ \textit{necesarios para cubrir } X\end{array} + \min \#\begin{array}{l}\textit{de vértices de } R \\ \textit{necesarios para cubrir } E \setminus X\end{array} \\ &= \min \#\textit{ de vértices de cubrimiento de } X. \end{aligned}$$

Entonces, el teorema anterior equivale a

$$\max_{M \textit{ matching}} |M| = \min_{C \textit{ cubrimiento}} |C|.$$

que corresponde al teorema de König visto en la clase 13.

A continuación se presenta un algoritmo que calcula el conjunto independiente común de cardinalidad máxima en D, cuya correctitud está garantizada por las proposiciones anteriores.

Algoritmo 1: Algoritmo de conjunto independiente común de cardinalidad máxima en digrafo de intersección.

Entrada: M_1; M_2.
$I \leftarrow \emptyset$
$D \leftarrow D(M_1, M_2, I)$
Elegir X_1, $X_2 \subseteq S \setminus I$, con $X_1 \cap X_2 = \emptyset$
while $\exists\, X_1$-X_2 *camino en* D **do**
    $P \leftarrow X_1$-X_2 camino más corto en D
    $I \leftarrow I \Delta V(P)$
    Recalcular D, X_1, X_2
end
$X \leftarrow \{x \in S : \exists\, x\text{-}X_2 \text{ camino en } D\}$
return (I, X)

Complejidad. Sea $n = |E|$, donde E es el conjunto de arcos de las matroides. Notemos que obtener el digrafo D, X_1 y X_2 en el algoritmo toma $O(n^2)$ unidades de trabajo y llamadas a oráculos de independencia para M_1 y M_2. Además, cada vez que se pasa por el *while* se debe revisar si existe un camino entre los conjuntos X_1 y X_2 en D, lo que se puede hacer agregando dos nodos s y t al digrafo, uno conectando a todos los vértices de X_1 y el otro conectando a todos los vértices de X_2, luego se busca con BFS un s-t camino, lo que toma $O(n^2)$.

Dentro del *while*, calcular el X_1-X_2 camino más corto demora $O(n^2)$ y luego calcular $I\Delta V(P)$ demora $O(n)$. Recalcular D, X_1 y X_2 toma, al igual que al comienzo, $O(n^2)$. Finalmente, asignar a X aquellos elementos $x \in X$ tal que haya un $x - X_2$ camino en D toma lo mismo que encontrar un $x - X_2$ camino, por lo tanto esta operación demora $O(n^2)$. Además, como el *while* se ejecuta a lo más n veces, entonces el algoritmo es $O(n^3)$.

Veamos ahora el siguiente problema. Se requiere diseñar un algoritmo para determinar si un grafo $G = (V, E)$ tiene dos árboles generadores disjuntos y, en caso de que existan, encontrarlos.

Planteemos el problema en términos de matroides. Sea $M = M(G)$ la matroide gráfica de G. Recordemos que para $T \subseteq E$

$$T \in \mathcal{I}(M) \Leftrightarrow T \text{ no tiene ciclos} \Leftrightarrow T \text{ es un bosque}$$

Buscamos entonces un conjunto $T \subseteq E$ tal que:

- $T \in \mathcal{I}(M)$
- $|T| = n - 1$, es decir, que T sea bosque generador.
- En $E \setminus T$ haya un árbol generador de G, es decir, $T \in \mathcal{I}(M^*)$.

Considerando esto, queremos revisar si

$$n - 1 \; = \max_{T \in \mathcal{I}(G) \cap \mathcal{I}^*(G)} |T|.$$

Para ello, estudiemos el siguiente algoritmo.

Algoritmo 2:

Usamos el Algoritmo 1 con $M_1 = M(G)$, $M_2 = M^*(G)$, para encontrar T que maximice $|T|$, sujeto a que $T \in \mathcal{I}_1 \cap \mathcal{I}_2$.
Si $|T| < n - 1$ no existe solución (se puede encontrar un conjunto $X \subseteq E$ tal que $|T| = r_1(X) + r_2(E \setminus X)$).
Si $|T| = n - 1$, buscamos/encontramos T_2 en $G \setminus T$ árbol generador.

Del teorema de intersección de matroides, existen dos árboles disjuntos si y sólo si

$$n-1 = \min_{X\subseteq E}\ r_1(X) + r_2(E\setminus X)$$

Realizamos el siguiente cálculo:

$$\begin{aligned} r_1(X)+r_2(E\setminus X) &= r_1(X) + [\ |E\setminus X| + r_1(X) - r_1(E)\ ]\,,\\ &= |E\setminus X| + 2r_1(X) - r1(E),\\ &= |E\setminus X| + 2[\ |V| - cc(X)\ ] - r1(E),\\ &= n+1+|E\setminus X| - 2cc(X). \end{aligned}$$

Reemplazando en la expresión anterior obtenemos que existen dos árboles disjuntos si y sólo si

$$n-1 = \min_{X\subseteq E}\ n+1+|E\setminus X| - 2cc(X),$$

es decir, $\forall X\subseteq E$, $|E\setminus X| \geq 2(cc(X)-1)$.

Problema controlable. Probar que existen k árboles generadores arista-disjuntos en G si y solo si para toda partición de V en p partes se tiene que

$$d(V_1,\ldots,V_p) \geq k\,(p-1).$$

3. Flujos en redes

Comenzamos ahora el estudio de un concepto fundamental en optimización combinatorial: el de flujos en redes. Supongamos, a modo de ejemplo, que tenemos que transportar agua a través de tuberías desde s (nodo origen) hasta t (nodo destino), como se muestra en la Figura 1. Para este propósito tenemos distintas rutas que podemos utilizar y cada tubería posee distinta capacidad, es decir, cada tubería soporta un volumen distinto de agua. Queremos elegir la ruta de tuberías que permita llevar la mayor cantidad de agua al destino t respetando las capacidades de cada tubería. Para solucionar problemas como este plantearemos la siguiente teoría.

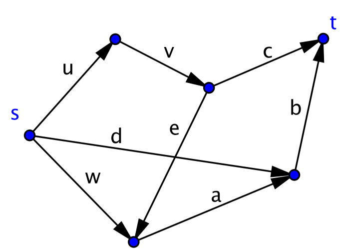

Figura 1. Ejemplo de flujo en redes.

Definición 1. Sea $G=(V,E)$ un digrafo. Una función de capacidad es una función $U: E\longrightarrow \mathbb{R}_+$, es decir, que a cada arista le asigna un valor real positivo.

Definición 2. Llamamos red a una tupla (G,u,s,t) donde $G=(V,E)$ es un digrafo, u una función de capacidad, s un nodo origen y t un nodo destino.

Definición 3. Un flujo es una asignación $f: E\longrightarrow\mathbb{R}$ tal que

$$\forall\, v\in V\setminus\{s,t\} : f(\delta^-(v)) = f(\delta^+(v)), \qquad \text{donde } f(X) = \sum_{e\in X} f(e),$$

es decir, todo lo que entra a cada nodo tiene que salir.

Esta condición se llama condición de conservación de flujo. Además, se dice que un flujo es factible si

$$\forall\, e\in E \quad 0\leq f(e)\leq u(e),$$

es decir, se dice factible si el flujo respeta la capacidad de los arcos de la red.

Definición 4. Definimos el flujo neto de entrada para un vector $X : E \to \mathbb{R}$ como:

$$X^{IN}(v) = X(\delta^-(v)) - X(\delta^+(v))$$

Observación 1. *Sea $X : E \to \mathbb{R}$. $X(e)$ también se interpreta como $X \in \mathbb{R}^E$, X_e. Además uno puede extender X a conjuntos $F \subseteq E$,*

$$X(F) = \sum_{e \in F} X_e.$$

Observación 2. *f es flujo si y sólo si $f^{IN}(v) = 0,\ \forall v \in V \setminus \{s,t\}$*

Veamos ahora el siguiente problema. Se requiere encontrar el flujo f factible en (G, U, s, t) que maximice $f^{IN}(t)$, es decir,

$$\begin{aligned} \text{máx} \quad & f^{IN}(t) \\ \text{s.a} \quad & f = 0,\ \forall v \in V \setminus \{s,t\} \\ & 0 \le f_e \le U_e,\ \forall e \in E \end{aligned}$$

En realidad, esto corresponde a un PL. Veremos más adelante que el PL se puede resolver en tiempo polinomial en el tamaño de la entrada.

Ahora, estudiaremos específicamente el problema de flujos. Definimos el valor de un flujo f como

$$valor(f) = f^{IN}(t).$$

Teorema 2. *Si f es un flujo factible, entonces*

1. $valor(f) = f^{OUT}(s) := -f^{IN}(s)$.

2. $\forall S \subseteq V$ tal que $s \in S$ y $t \notin S$ se tiene que

$$valor(s) = f^{OUT}(S) = f(\delta^-(S)) - f(\delta^+(S)).$$

Demostración.

1. Como f es un flujo factible, sabemos que $\forall v \in V \setminus \{s,t\}$ $f^{IN}(v) = 0$. Entonces

$$f^{IN}(t) - f^{OUT}(s) = f^{IN}(t) + f^{IN}(s).$$

Por lo que,

$$\begin{aligned} f^{IN}(t) + f^{IN}(s) &= \sum_{v \in V} f^{IN}(v), \\ &= \sum_{v \in V} (f(\delta^-(v)) - f(\delta^+(v))), \\ &= \sum_{v \in V} \left(\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e \right), \\ &= \sum_{v \in V} \sum_{e \in \delta^-(v)} f_e - \sum_{v \in V} \sum_{e \in \delta^+(v)} f_e, \\ &= f(E) - f(E), \\ &= 0. \end{aligned}$$

Por lo tanto $f^{IN}(t) - f^{OUT}(s) = 0$. Concluimos que $valor(f) = f^{OUT}(s)$.

2. Como f es un flujo factible, tenemos que $\forall v \in V \setminus \{s,t\}$ $f^{OUT}(v) = 0$. Luego

$$
\begin{aligned}
f^{OUT}(s) &= \sum_{v \in S} f^{OUT}(v), \\
&= \sum_{v \in S} (f(\delta^+(v)) - f(\delta^-(v))), \\
&= \sum_{v \in S} \sum_{e \in \delta^+(v)} f_e - \sum_{v \in S} \sum_{e \in \delta^-(v)} f_e, \\
&= (f(\delta^+(S)) + f(E(S))) - (f(\delta^-(S)) + f(E(S))), \\
&= f(\delta^+(S)) - f(\delta^-(S)), \\
&= f^{OUT}(S).
\end{aligned}
$$

Donde $E(S) = \{e \in E \colon t(e), h(e) \in S\}$. Por la parte 1, tenemos que $valor(f) = f(\delta^+(S)) - f(\delta^-(S))$.

■

Definición 5. Sea (G, u, s, t) una red y $S \subset V$. Diremos que S es un s-t corte si $s \in S$ y $t \notin S$. Luego, denotamos la capacidad de un s-t corte por

$$u(\delta^+(S)) = \sum_{e \in \delta^+(S)} u_e.$$

Teorema 3. *Si f es factible, entonces $\forall S \subset V$ tal que $s \in S$ y $t \notin S$, $valor(f) \leq u(\delta^+(S))$.*

Demostración. Se tiene que

$$valor(f) = f(\delta^+(S)) - f(\delta^-(S)) \leq f(\delta^+(S)) \leq u(\delta^+(S)).$$

■

Notemos que por el *Teorema 3* tenemos el siguiente resultado de dualidad débil:

$$\max_{f\colon \text{factible en } (G,u,s,t)} \text{valor}(f) \leq \min_{S\ s-t\colon \text{corte}} u(\delta^+(S)).$$

La próxima clase veremos en más detalle la aplicación de dualidad débil a flujos en redes.