MA4701. Optimización Combinatorial. 2013.
Profesor: José Soto
Escriba(s): Rodolfo Núñez U. y Sebastián Urzúa B..
Fecha: 23 de Agosto de 2013.

Cátedra 2

1. Recuerdo de las Definiciones de Grafos

Sea $G = (V, E)$ un grafo.

Definición 1 (Bosque)**.** Se dice que G es un *bosque* ssi G es acíclico.

Definición 2 (Árbol)**.** G se dice *árbol* ssi G es un bosque conexo.

Definición 3 (Hoja de un grafo)**.** Una *hoja* es un vértice $v \in V$ tal que $d(v) = 1$.

Definición 4 (Vértice aislado)**.** Un *vértice aislado* es un vértice $v \in V$ tal que $d(v) = 0$.

2. Problema del Subgrafo Generador Conexo de Peso Mínimo

Notación 1.

Si $c : A \to \mathbb{R}$, $X \subseteq A$,

$$c(X) = \sum_{x \in X} c(x).$$

Dado $G = (V, E)$ grafo, $w : E \to \mathbb{R}^+$, el problema del subgrafo generador de peso mínimo consiste en encontrar $H \subseteq G$ generador, conexo, que minimice $w(E(H))$.

2.1. Preliminares

Es importante notar que este problema tiene soluciones que son árboles. Para ésto, se probarán los siguientes resultados previos.

Lema 1. *$e = uv$ es una arista de un ciclo C en $G \iff G - e$ contiene un camino de u a v.*

Demostración.

$\Rightarrow$ Sea $C = uw_1w_2 \cdots w_nv$ un ciclo en G. Al retirar $e = uv$ del grafo G, un (u, v)-camino posible es $P = uw_1w_2 \cdots w_nv$.

$\Leftarrow$ Sea G grafo tal que $G - e$ tenga un (u, v)-camino. Sea $P = uw_1w_2 \cdots w_nv$ dicho camino. Al agregar $e = uv$ a G se forma un ciclo.

□

Corolario 1. *Si G es conexo y e pertenece a un ciclo, entonces $G - e$ es conexo.*

Demostración. Sea $e = uv$ una arista de G. Como G es conexo, el único problema es que u y v no se conecten en $G - e$. Pero como $e \in C$, u y v siguen conectados por el otro camino del ciclo. □

Lema 2. *Si $G' \subseteq G$ es generador de G y e pertenece a un ciclo de G', entonces $G' - e$ es generador de G.*

Demostración. Como G' es generador, y $e \in C \subseteq G'$, $G' - e$ sigue teniendo aristas que inciden en todos los vértices de G. Luego, $G' - e$ es generador de G. □

Con lo anterior, y recordando que w es una función de pesos no negativa, el problema es equivalente a encontrar un árbol generador de peso mínimo.

2.2. Algoritmo genérico para encontrar un Árbol Generador

Algoritmo 1 Árbol Generador

1: Dado $G = (V, E)$
2: Sea $r \in V$
3: $U = \{r\}$ {vértices visitados}
4: $T = \emptyset$
5: **while** $\delta(U) \neq \emptyset$ **do**
6: Sea $e \in \delta(U)$, $e = uv$, $u \in U$, $v \notin U$
7: $U \leftarrow U + v$
8: $T \leftarrow T + e$
9: **end while**
10: Devolver $H = (U, T)$

Queda propuesto demostrar que tanto BFS como DFS son casos particulares de este algoritmo.

Observación 1. *Si $w(e) \equiv 1$, $\forall e \in E$, todo árbol generador tiene peso $n - 1$.*

Lema 3. *El Algoritmo 1 devuelve un árbol generador.*

Demostración.

Parte 1 Veamos primero que en cada iteración, (U, T) es árbol.
Razonemos por inducción. Para el caso base ($i = 1$) es directo que $H_1 = (\{r\}, \emptyset)$ es un árbol. Por hipótesis inductiva, sea $H_i = (U_i, T_i)$ el grafo al principio de la iteración i, que supondremos que es un árbol. Así, $H_{i+1} = (U_i + v_i, T_i + e_i)$, donde $e_i = u_i v_i$ es la arista elegida en la iteración i y $u_i \in U_i$. Es claro que H_{i+1} es conexo, dada la conexidad de H_i. Además, H_{i+1} no tiene ciclos. En efecto, de tener un ciclo $C \subseteq T_i + e_i$, entonces $e_i \in C$, por el Lema 1, $T_i = (T_i + e_i) - e_i$ debería contener un camino de u_i a v_i, pero v_i está fuera de U_i, lo que es una contradicción.

Parte 2 Una vez probado que el algoritmo va construyendo un árbol por cada iteración, veamos ahora que si el algoritmo termina, entonces H es generador.

Para ello, se probará el siguiente Teorema:

Teorema 1. *G es conexo $\iff$ $(\forall U \subseteq V, \emptyset \neq U \neq V), \delta(U) \neq \emptyset$.*

Demostración.

- $\Rightarrow$ Sea $G = (E, V)$ un grafo conexo, y $U \subseteq V$ tal que $\emptyset \neq U \neq V$. Eso implica que $V \setminus U \neq \emptyset$. Por ser G conexo, existe un (u, w)-camino, con $u \in U$ y $w \in V \setminus U$. Veamos que dicho camino contiene una arista en el corte. En efecto, sean $u = v_1 v_2 \cdots v_k = w$, con $u \in U$ y $w \in V \setminus U$, vértices que visitan el camino y sea i el mínimo índice tal que $v_i \notin U$, luego, $e = v_{i-1} v_i \in \delta(U)$.
- $\Leftarrow$ Supongamos que G no es conexo. Luego, tiene más de una componente conexa. Sea U el conjunto de los vértices de una de las componentes conexas. Esto implica que $\delta(U) = \emptyset$ y, además, $\emptyset \neq U \neq V$, lo cual es una contradicción. □

Para terminar la parte 2, notemos que cuando el algoritmo termina, $\delta(U) = \emptyset$. Como $U \neq \emptyset$ y G es conexo, por el Teorema 1, tendremos que $U = V$. Por lo tanto H es generador.

Parte 3 Sólo falta probar que el algoritmo termina.
En efecto, en cada iteración, V crece en un vértice. Como hay n vértices, el algoritmo termina en $n-1$ iteraciones. □

2.3. Algoritmo de Prim

Volvamos a la versión con pesos. Una idea es modificar el algoritmo anterior de la siguiente manera: Cada vez que podemos elegir una arista, elegimos la más barata.

Observación 2. *Este es un algoritmo de tipo "avaro", "glotón" o "greedy" en inglés. Este tipo de algoritmos tratan de tomar la opción más fácil o cada vez que pueden hacer algo, lo hacen.*

Algoritmo 2 Algoritmo de Prim (Prim 1957 – Jarník 1930)

Dado $G = (V, E)$
Sea $r \in V$
Iniciar $H = (U, T)$
$U = \{r\}$
$T = \emptyset$
while $\delta(U) \neq \emptyset$ **do**
    Sea $e = uv \in \delta(U)$, $u \in U$, $v \notin U$ tal que e es la arista de menor peso en $\delta(U)$
    $U \leftarrow U + v$
    $T \leftarrow T + e$
end while
Devolver $H = (U, T)$

Teorema 2. *El Algoritmo 2 (Prim) devuelve un árbol generador de peso mínimo.*

Demostración. Por el Lema 3, devolvemos un árbol generador. Veamos ahora que en cada iteración, H_i es subgrafo de algún árbol de costo mínimo. Al principio, $H_1 = (\{r\}, \emptyset)$ está contenido en alguna solución. Por inducción, asumiremos que $H_i \subseteq S$, con $S = (V, F)$ árbol generador de peso mínimo y demostraremos que $H_{i+1} = (U_i + v_i, T_i + e_i)$ también lo es.
Caso 1: Si $e_i \in F$, entonces $T_i + e_i \subseteq F$, por lo tanto $H_{i+1} \subseteq S$.
Caso 2: Si $e_i = u_i v_i \notin F$. Sea $f \in \delta(U_i) \cap F$. Notemos que $\delta(U_i) \cap F \neq \emptyset$. En primer lugar, $\delta(U_i) \neq \emptyset$, ya que si no lo fuera, el algoritmo ya habría terminado. Luego, como F es un árbol generador, incide en todos los vértices. En particular, incide en los vértices de U_i. Por lo tanto, si no existiera una arista en F que esté en el corte de U_i, entonces U_i no estaría conectado al resto del grafo por S, lo que contradice que S debe ser conexo (porque es árbol). Por lo tanto $\delta(U_i) \cap F \neq \emptyset$. Sea $f \in \delta(U_i) \cap F$. Es fácil ver que $F + e_i - f$ también es el conjunto de aristas de un árbol generador, y $w(F + e_i - f) = w(F) + w(e_i) - w(f) \leq w(F)$ ya que $w(e_i) \leq w(f)$. Por lo tanto $S' = (V, F + e_i - f)$ también es árbol generador de peso mínimo y $H_{i+1} \subseteq S'$.

□