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

Cátedra 9

1. Principio de Optimalidad de Bellman Especializado

Sea $G = (V, E)$ un grafo dirigido, $s \in V$. Sea $l : E \to \mathbb{R}$ una función de largos conservativa (esto es, no hay ciclos negativos). Queremos calcular los caminos de costo mínimo de s a cualquier $v \in V$, con $d(s, v)$ el largo de este camino. Para esto, enunciaremos el Principio de Optimalidad de Bellman Especializado.

Teorema 1. *Sea P un camino de s a v, de largo mínimo usando a lo más i nodos intermedios. Sea $e = (u, v)$ el último arco de P. Entonces $P - e$ es un camino de s a u de largo mínimo usando a lo más $i - 1$ nodos intermedios.*

Demostración. Sea R un camino de largo mínimo de s a u que usa a lo más $i - 1$ nodos internos. Veamos primero que pasaría si R no tomara la arista e. Si R no pasara por v, entonces $R+e$ es $(s, v)-$camino que usa a lo más i nodos internos. Esto implica que $l(R + e) = l(R) + l(e) \geq l(P)$, por lo tanto $l(R) \geq l(P) - l(e) = l(P - e)$. Además, como suponemos que R es un camino de largo mínimo entre s y u que usa a lo más $i - 1$ nodos internos, tenemos la otra desigualdad, lo que nos dice que $l(R) = l(P - e)$.

Por otra parte, considerando el caso cuando R toma el nodo v, consideremos R_{sv} y R_{vu} las dos partes de R, divididas según el nodo v. En el caso de agregar la arista e se forma un ciclo dirigido C, es decir, $R + e = R_{sv} + C$, con $C = R_{vu} + e$. Sabemos que $l(R) \leq l(P - e)$. Supongamos por contradicción que $l(R) < l(P - e)$. Luego $l(C) = l(R_{vu}) + l(e) = l(R) - l(R_{sv}) + l(e) \leq l(R) - l(P) + l(e) \leq l(R) - l(P - e) < 0$, pues P es mínimo, y además suponíamos que $l(R) < l(P - e)$, y que l es una función de largos conservativos. Esto implica que $l(C) < 0$, lo que es una contradicción.

$\square$

1.1. Algoritmo de Bellman-Ford

Definimos el algoritmo de Bellman–Ford como en el Algoritmo 1. Notaremos al predecesor de v en el camino $i-$óptimo de s a v como $\pi_i(v)$. Este algoritmo llena una tabla de $d_i(s, v)$, y una tabla $\pi_i(v)$, con $i = 0, \ldots, n - 1$ y $v \in V$. La idea es probar que $d_i(s, v)$ es el largo del camino mínimo de s a v usando a lo más i nodos intermedios.

Además, si llamamos $B_i(s, v)$ al paseo de s a v dado por el reverso de $(v, \pi_i(v), \pi_{i-1}(\pi_i(v)), \ldots, s)$, entonces $B_i(v, s)$ es el camino de costo mínimo de s a v usando a lo más i nodos.

Notaremos el valor null como $\perp$.

Algoritmo 1 (Schimbel 1955, Ford 1956, Bellman 1958, Moore 1959)

Require: $G = (V, E)$, $s \in V$, función de largos l.
Ensure: $d_i(s, v)$, $\pi_i(v)$.
1: Dado $s \in V$
2: $d_0(s, s) = 0, \pi_0(s) = s$ {por convención}
3: **for all** $v \in V$ **do**
4: $\quad (d_0(s, v), \pi_0(v)) = \begin{cases} (+\infty, \perp) & \text{si } (s, v) \notin E \\ (l(s, v), s) & \text{si no} \end{cases}$
5: **end for**
6: **for all** $i = 1 : n - 1$ **do**
7: $\quad$ **for all** $v \in V$ **do**
8: $\quad\quad A = \min\limits_{U \in \delta^-(v)} \quad d_{i-1}(s, u) + l(u, v)$
9: $\quad\quad U^\star = \arg\min\limits_{U \in \delta^-(v)} \quad d_{i-1}(s, u) + l(u, v)$
10: $\quad\quad$ **if** $A < d_{i-1}(s, u)$ **then**
11: $\quad\quad\quad d_i(s, v) = A$ **and** $\pi_i(v) = U^\star$
12: $\quad\quad$ **else**
13: $\quad\quad\quad d_i(s, v) = d_{i-1}(s, v)$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end for**

1.2. Complejidad del Algoritmo 1

En la linea 3, tenemos un *for* que recorre todos los nodos y hace un trabajo constante para cada uno, por lo que es $O(n)$.

En la linea 6, tenemos un *for* sobre i que es $O(n)$, dentro del cual, en la linea 7, encontramos un *for* sobre $v \in V$ que también será $O(n)$. El trabajo que se realiza dentro de estos *for*, en las lineas 8 y 9 es el de encontrar un mínimo buscando en las aristas del grafo, por lo que puede ser de orden $O(m)$. Por lo que en esta sección tenemos un orden de $O(n^2 m)$.

En la linea 10 hay un if dentro del cual realizamos un trabajo constante.

A la rápida, se concluye que el algoritmo demora $O(n + n^2 m)$. Sin embargo, se puede hacer un análisis más exhaustivo del algoritmo para mejorar el tiempo de ejecución y obtener $O(nm)$.

Notemos que la búsqueda del mínimo y argmin en las lineas 8 y 9 se realiza sobre $U \in \delta^-(v)$, por lo que solo necesitamos probar $O(deg(v)$ casos. Por lo tanto, un mejor análisis del orden del Algoritmo 1 sería $O(n + m \sum\limits_{v \in V} \sum\limits_{e \in \delta^-(v)} O(1))$ por el "Handshaking Lemma", esto es igual a $O(n + n \sum\limits_{e \in E} O(1)) = O(n + nm) = O(nm)$.

1.3. Correctitud del Algoritmo 1

Teorema 2. *El Algoritmo 1 es correcto, es decir, $d_i(s, v)$ es el largo del camino mínimo de s a v usando a lo más i nodos, $\pi_i(v)$ son los predecesores de v en el camino $i-$óptimo de s a v y $B_i(s, v)$ es un camino de costo mínimo de s a v usando a lo más i nodos.*

Demostración. Por definición del algoritmo 1, si $d_i(s, v) < +\infty$, entonces $d_i(s, v) = l(B_i(s, v))$, siendo B_i un paseo de s a v con a lo más i nodos intermedios. Por otra parte, si $d_i(s, v) = +\infty$, luego v no es alcanzable desde s en a lo más $i + 1$ pasos. Además, $B_i(s, v)$ es un paseo mínimo de s a v con a lo más i nodos intermedios. Probaremos que $B_i(s, v)$ es camino, es decir, no repite vértices.

Observación 1. *Esto vale incluso si la función de largos l no es conservativa. Ahora usaremos que l es conservativa para probar que $B_i(s, v)$ es camino.*

La demostración a continuación procederá por inducción sobre i, las iteraciones del algoritmo.

$i = 0$ Es trivial, porque tenemos un paseo sin aristas.

$i \geq 1$ Sea $v \in V$ y P un camino mínimo de s a v usando a lo más i nodos intermedios. Así que se tiene que $l(P) \geq l(B_i(s,v))$. Sea $e = (u,v)$ el último arco de P. Por el Principio de Bellman, sigue que $l(P-e) = l(B_{i-1}(s,u))$, ya que $B_{i-1}(s,u)$ es el camino de largo mínimo de s a u que usa a lo más $i-1$ nodos intermedios, por hipótesis inductiva. Así, se tiene $P-e$ es camino óptimo de s a u que usa a lo más $i-1$ nodos intermedios.

Además,

$$l(P) = l(P-e) + l(e)$$
$$l(P) = d_{i-1}(s,u) + l(u,v)$$
$$l(P) \geq d_i(s,v) = l(B_{i-1}(s,\pi_i(v)) + l(\pi_i(v),v)).$$

Verifiquemos que, si $l(P) > l(B_{i-1}(s,\pi_i(v)) + l(\pi_i(v),v)$, se llega a una contradicción. En efecto, asumamos cierto que $l(P) > l(B_{i-1}(s,\pi_i(v)) + l(\pi_i(v),v)$. Luego, $B_i(s,v)$ sería un no-camino, es decir, $B_i(s,v) = B \cup C$, con B un $(s,v)-$camino y C un ciclo. Entonces se puede decir que $l(C) = l(B_i(s,v)) - l(B) \leq 0$, de lo que se infiere que $l(C) = 0$, ya que l es conservativo. Sea $w = \pi_j(v)$, con j número de nodos internos de B, y $u = \pi_i(v)$ un nodo en en ciclo. Claramente $w \neq u$. Pero entonces el predecesor de v cambió entre las iteraciones j e i. Luego $d_j(s,v) > d_i(s,v)$, por lo que $l(B) > l(B_i(s,v))$, con lo que se concluye que $l(C) < 0$, lo cual es una contradicción con el hecho de que l era conservativa.

Por lo tanto, $B_i(s,v)$ es camino y $l(B_i(s,v)) = l(P)$.

□

Teorema 3. *$d_{n-2}(s,v) = d(s,v)$, $\forall v \in V$.*

Demostración. Notando que cualquier camino de n nodos desde s hasta v tiene $n-2$ nodos intermedios, se concluye. □

Luego, $\pi(v) := \pi_{n-2}(v)$ es predecesor de v en algún $(s,v)-$camino óptimo.
Definamos ahora $H_s := (V,F)$, donde $F := \{(\pi(v),v), v \in V - s \text{ tal que } \pi(v) \neq \perp\}$.

Observación 2. *H_s es un árbol dirigido desde s. Ésto se conoce como **arborescencia**.*

Además, el único camino de s a v (si existe) en H_s es camino de largo mínimo en G de s a v. H_s se llama árbol dirigido de caminos mínimos.

Teorema 4. *Sea l función de largos arbitrarios. Se aplica el Algoritmo 1 de todas maneras. Luego, existe C ciclo negativo alcanzable desde s sí y solo si $\exists v \in V$ tal que $d_{n-1}(s,v) < d_n(s,v)$, y en ese caso podemos encontrar C en tiempo $O(nm)$.*

Demostración.

$\Leftarrow$ Si $d_{n-1}(s,v) < d_n(s,v)$, entonces $B_{n-1}(s,v) \neq B_{n-2}(s,v)$. Luego, $B_{n-1}(s,v) = B_{n-2}(s,v) + (\pi_{n-1}(v),v)$. Como $B_{n-1}(s,v)$ debe tener ciclo, eso implica que dicho ciclo debe tener largo negativo. Queda propuesto encontrarlo. Usando la contrucción de arriba, podemos detectar si existe algún ciclo negativo en todo G.

$\Rightarrow$ Razonando por contradicción, supongamos que, $\forall v \in V, d_n(s,v) = d_{n-1}(s,v)$, ya que el otro caso no tiene sentido por la definición de $d_n(s,v)$, ya que es un mínimo y al aumentar n solo le damos más opciones para minimizar. La idea de la demostración es ver que todo ciclo alcanzable desde s es no negativo.

Sea C un ciclo alcanzable desde s, denotamos $C = v_1, v_2, \ldots, v_k$ su secuencia de vértices. Entonces se tiene que todos los $d_{n-1}(s,v_i)$ son finitos, y además, por la descripción del algoritmo, $d_n(s,v_1) \leq d_{n-1}(s,v_k) + l(v_k,v_1)$ y además $d_n(s,v_i) \leq d_{n-1}(s,v_i) + l(v_{i-1},v_i)$ para $i = 2,\ldots,k$. Sumando todas estas desigualdades obtenemos $\sum_{i=1}^k d_n(s,v_i) \leq (\sum_{i=1}^k d_{n-1}(s,v_i)) + l(C)$. Es decir, $\sum_{i=1}^k (d_n(s,v_i) - d_{n-1}(s,v_i)) \leq l(C) < 0$. Luego, el lado izquierdo es 0 por la suposición por contradicción y obtenemos que $0 < 0$ que es una contradicción, con lo cual concluimos.

□

Corolario 1. *En $O(nm)$ se puede detectar la existencia de un ciclo negativo, sin importar si es o no alcanzable desde s. Además, de existir, puede ser reportado en el mismo tiempo asintótico.*

Demostración. Para esto, basta modificar el digrafo, agregando un nodo s_0 y un arco de largo 0 desde s hacia cada nodo del grafo G, y luego aplicar el Teorema 4 al digrafo aumentado, partiendo desde s_0. □