MA3705. Algoritmos Combinatoriales. 2014.
Profesor: José Soto
Escriba(s): Piero Zanoco, Felipe Garrido, Guido Besomi
Fecha: 01 de Septiembre 2014

Cátedra 10

1. Recuerdo

Estamos estudiando el problema de encontrar caminos y paseos de costo mínimo en grafos dirigidos. Formalmente, dado un digrafo $G = (V, E)$ y una función de largo $l\colon E \to \mathbb{R}$, definimos:

- $\mathcal{P}_{\leq i}(u, v) = \{u\text{-}v \text{ caminos con } \leq i \text{ arcos}\}$.
- $\mathcal{W}_{\leq i}(u, v) = \{u\text{-}v \text{ paseos con } \leq i \text{ arcos}\}$.
- $d_i(u, v) := \min\limits_{P \in \mathcal{W}_{\leq i}(u,v)} l(P)$.

Introduciremos ahora la noción de función de largo conservativa y veremos cómo se relaciona con la definición de distancia recién presentada:

Definición 1. La función de largo l se dice conservativa si y sólo si $\nexists$ ciclo C con $l(C) < 0$.

Proposición 1. Si la función de largo l es conservativa, entonces $d_i(s, v) = \min\limits_{P \in \mathcal{P}_{\leq i}(u.v)} l(P)$.

Proposición 2. Sea l conservativo o no (no importa). Entonces

$$d_i(s, v) = \min_{w \in N^-(v)+v} \{d_{i-1}(s, w) + l(w, v)\} \text{ , donde abusando de notación suponemos } l(v, v) = 0.$$

o bien,

$$d_i(s, v) = \min\{\min_{w \in N^-(v)} \{d_{i-1}(s, w) + l(w, v)\},\ d_{i-1}(s, v)\}.$$

Observación 1. Si $d_i(s, v) < \infty$, definimos $w^* = \arg\min_{w \in N^-(v)+v}\{d_{i-1}(s, w) + l(w, v)\}$. Entonces, w^* es predecesor de v en algún s-v paseo de $\mathcal{W}_{\leq i}(s, v)$ mínimo. Si $d_i(s, v) = \infty$, entonces v no se puede alcanzar desde s con $\leq i$ arcos.

Principio de optimalidad de Bellman (para l conservativo). Si P es u-v camino mínimo, Q subcamino de P (entre x e y), entonces Q es x-y camino mínimo.

2. Algoritmo de Bellman - Ford

Algoritmo Bellman-Ford.

Algoritmo 1: Algoritmo Bellman-Ford

Input: $G = (V, E), l : E \longrightarrow \mathbb{R}$ (no necesariamente conservativo), $s \in V$.
Output: Si l es conservativo, entonces devuelve una arborescencia de caminos mínimos desde s. Si l no es conservativo y existe un ciclo negativo alcanzable desde s, entonces el algoritmo lo reporta.

```
for v ∈ V do
    d_0(s, v) = { 0    s = v
                { +∞   s ≠ v
end
for i = 1, ..., n − 1 do
    for v ∈ V do
        d_i(s, v) ← d_{i−1}(s, v)
    end
    for e = (w, v) ∈ E do
        if d_i(s, v) > d_{i−1}(s, w) + l(w, v) then
            d_i(s, v) ← d_{i−1}(s, w) + l(w, v)
            π(v) ← w
        end
    end
end
```

Dejamos pendiente hasta el final de la clase como se revisa la existencia de un ciclo negativo alcanzable desde s.

Complejidad: $\mathcal{O}(n(n+m))$

Correctitud

Teorema 1. *Al final de la iteración $i \in \{0, ..., n-1\}$. Sean*

- $R_i = \{v \in V : \ \pi(x) \text{ está definido}\} \cup \{s\}$.
- $F_i = \{(\pi(v), v) : \ v \in R_i - s\}$.
- $H_i = (R_i, F_i)$.

Entonces se tiene que:

(1) Para todo $(w, v) \in F_i$, $d_i(s, v) \geq d_i(s, w) + l(w, v)$

(2) Si H_i tiene un ciclo C, entonces $l(C) < 0$.

(3) Si l es conservativa, entonces H_i es una arborescencia con raíz s.

Antes de demostrar el teorema, recordemos la definición de arborescencia:

Definición (Arborescencia): Un digrafo $H = (R, F)$ es una arborescencia si satisface:

1. Su grafo subyacente es árbol.
2. $\forall v \in R$, el único s-v camino en el árbol subyacente está dirigido de s a v.

Demostración del teorema 1. El caso $i = 0$ es fácil de probar.

Sea $i \geq 1$. Probemos (1): Sea $e = (w, v) \in F_i$. Por la definición de F_i, $w = \pi(v)$. Esto quiere decir que en alguna iteración $j \leq i$ el algoritmo fijó $\pi(v) \leftarrow w$, y de ahí en adelante se mantuvo. En ese momento:

$d_j(s,v) > d_{j-1}(s,w) + l(w,v)$ se cambió a $d_j(s,v) = d_{j-1}(s,w) + l(s,w)$.

En los pasos siguientes, $d_i(s,v) = d_k(s,v), \forall k$ entre j e i, si no, hubiese cambiado π. Además, $d_{j-1}(s,w) \geq d_i(s,w)$ pues $d_i(s,\cdot)$ es decreciente en i. Luego,

$$\begin{aligned} & d_i(s,v) = d_j(s,v) \geq d_i(s,w) + l(w,v) \\ \Rightarrow \quad & d_i(s,v) \geq d_i(s,w) + l(w,v) \end{aligned}$$

Probemos ahora (2): Sea C el ciclo de F_i y sea (w,v) el arco que entró último a F_i de C, digamos en la j-ésima iteración. Justo antes de entrar, teníamos que:

$$\begin{aligned} & d_j(s,v) > d_{j-1}(s,w) + l(w,v) \geq d_j(s,w) + l(w,v) \\ \text{Por lo tanto} \qquad & d_j(s,w) > d_j(s,v_0) + l(w,v) \end{aligned} \tag{1}$$

donde v_0 corresponde a w.

Por la parte uno, se tiene que para todo k, $d_j(s,v_{k+1}) \geq d_j(s,v_k) + l(v_k,v_{k+1})$. Sumando esta desigualdad sobre todo k y luego sumándola a (1) se obtiene:

$$\underbrace{\sum_{v_k \in V(C)} d_j(s,v_{k+1})}_{S_1} > \underbrace{\sum_{v_k \in V(C)} d_j(s,v_k)}_{S_2} + \underbrace{\sum_{v_k \in V(C)} l(v_k,v_{k+1})}_{l(C)}$$

Notando que $S_1 = S_2$ se concluye que $l(C) < 0$.

Probemos (3): Por la parte anterior, H_i no tiene ciclos dirigidos. Además, cada vértice en $R_i - s$ posee $deg^-_{H_i}(v) = 1$ (número de arcos entrantes a v en H_i). Luego, ningún vértice tiene dos arcos entrantes. Por lo tanto, en H_i tampoco podemos encontrar ciclos no dirigidos.

Luego, el grafo subyacente es acíclico. Además, como $deg^-_{H_i}(v) = 1, \forall v \in R_i - s$ y $deg^-_{H_i}(s) = 0$, al partir de v, podemos devolvernos a $\pi(v)$, luego a $\pi(\pi(v))$ y asi, hasta llegar a s, del cual no se puede seguir pues no tiene vecinos entrantes.

$\therefore H_i$ es arboresencia con raíz s. □

Volvemos al estudio de la correctitud de Bellman-Ford.

Si l es conservativo, el único s-v camino ($\forall v \in R_{n-1}$) en H_{n-1} es un camino en $P_{\leq i}(s,v)$ de largo mínimo. Probémoslo:

Por la parte (1) del teorema anterior, $d_{n-1}(s,v_k) \geq d_{n-1}(s,v_{k-1}) + l(v_{k-1},v_k)$, pero el camino más largo es de $n-1$ pasos, luego $d_{n-1}(s,v_k) = d(s,v_k)$ (largo del camino mínimo entre s y v_k).

$$\Rightarrow d(s,v_k) \geq d(s,v_{k-1}) + l(v_{k-1},v_k) \geq d(s,v_{k-1}) + d(v_{k-1},v_k),$$

pero por desigualdad triangular; $d(v_{k-1},v_k) + d(s,v_{k-1}) \geq d(s,v_k)$, entonces $d_{n-1}(s,v_k) = d_{n-1}(s,v_{k-1}) + l(v_k,v_{k-1})$.

Por lo tanto $d_{n-1}(s,v) = \sum_{k=1}^{t} l(v_{k-1},v_k)$, es decir, el camino de s a v en H_i es el camino mínimo y se conlcuye la correctitud. Hemos probado el siguiente teorema.

Teorema 2. *Si l es conservativo, $H = (R_{n-1}, F_{n-1})$ es una arborescencia que codifica caminos mínimos de s a todo nodo.*

Surge la pregunta de, en caso que exista un ciclo dirigido, cual es la mejor forma de encontrarlo.

Teorema 3. *Sea l función de largos arbitraria. Existirá C ciclo negativo alcanzable desde s ssi $\exists v$ tal que $d_n(s,v) < d_{n-1}(s,v)$ y para encontrarlo, basta realizar el "for" por una unidad más $i = n$.*

Luego, para:

(i) Revisar si existe ciclo negativo alcanzable desde s se revisa si $d_n(s,\cdot) = d_{n-1}(s,\cdot)$ (si son distintos, existe ciclo negativo alcanzable desde s).

(ii) Devolver ciclo negativo, se revisa cada H_i, $i = \{1,...,n\}$. Si en algún momento se crea un ciclo, parar y devolver (se utiliza BFS para encontrarlo).

Observación 1. Es fácil verificar que incluso con estas operaciones extras, el algoritmo tiene complejidad $O(n(n+m))$.

Observación 2. Es posible determinar si l es conservativa en tiempo $O(n^2(n+m))$ aplicando Bellman-Ford desde cada nodo. Existen formas más rápidas. Una de ella es usar el algoritmo de Floyd-Warshall que se ve más adelante en el curso para detectar si las matrices $d_n(\cdot,\cdot)$ y $d_{n-1}(\cdot,\cdot)$ son distintas.