MA3705. Algoritmos Combinatoriales. 2014.
Profesor: José Soto
Escribas: Catalina Pesce e Ian Vidal
Fecha: 29 de Agosto 2014.

Cátedra 9

1. Recuerdo de la clase pasada

En la clase anterior describimos el algoritmo de paseo de largo mínimo desde un vértice s, utilizando k arcos. Volveremos a plantear este algoritmo para estudiar su correctitud y complejidad.

Algoritmo 1: Algoritmo de paseo mínimo desde s hasta un vértice $v \in V$ con k arcos.

Entrada: $G = (V, E)$; $l : E \longrightarrow \mathbb{R}$.
for $v \in V$ **do**

$$D(s, v; 0) = \begin{cases} 0 & \text{si } v = s, \\ +\infty & \text{si } v \neq s. \end{cases}$$

end
for $j = 1 \ldots k$ **do**
  for $v \in V$ **do**

$$D(s, v; j) = \min_{w \in N^-(v)} \{D(s, w; j-1) + l(w, v)\}.$$

$$\Pi(s, v; j) = \operatorname{argmín}_{w \in N^-(v)} \{D(s, w; j-1) + l(w, v)\}.$$

  end
end

1.1. Correctitud del algoritmo 1

Primero, notemos que el algoritmo opera sobre un número finito de nodos una cantidad finita de veces sin entrar en ciclos, por lo que el algoritmo termina. Ahora, veamos que cumple con nuestra definición de distancia, es decir, que si consideramos $D(s, v; j)$ el valor del paseo mínimo entre s y v con j arcos, el valor calculado por el algoritmo es realmente el que deseamos, para ello lo probaremos a través de una doble desigualdad.

- ($\leqslant$) Basta notar que si R es un paseo mínimo desde s hasta w con $j-1$ arcos, con $w \in N^-(v)$, esto nos asegura que $R + (w, v)$ es un paseo de j arcos desde s hasta v. Por lo que de nuestra definición de $D(s, v; j)$ se obtiene la desigualdad deseada.

- ($\geqslant$) Sea P un s-v camino con j arcos tal que $l(P) = D(s, v; j)$. Suponiendo que $D(s, v; j)$ es finito, este pase existe pues es el mínimo de un conjunto finito, entonces $P = u_1 u_2 \ldots u_{n-1} u_n; u_1 = s, u_n = v$, notando que:

$$\begin{aligned} l(P) &= l(u_1 \ldots u_{n-1}) + l(u_{n-1} u_n) \\ l(P) &\geqslant \min_{u_{n-1} \in N^-(u_n)} (D(u_1, u_n - 1; j-1) + l(u_{n-1}, u_n)) \\ l(P) &\geqslant \min_{u_{n-1} \in N^-(v)} (D(s, u_n - 1; j-1) + l(u_{n-1}, v)) \\ D(s, v; j) &\geqslant \min_{w \in N^-(v)} (D(s, w; j-1) + l(u_w, v)) \end{aligned} \tag{1}$$

(1): La demostración anterior también vale si no existe paseo desde s a v con j arcos.En dicho caso tanto como el lado izquerdo de nuestra ecuación vale $+\infty$.

1.2. Complejidad del algoritmo 1

Estudiemos ahora la complejidad del algoritmo de paseos mínimos. Consideremos $n = |V|$, $m = |E|$, entonces tenemos lo siguiente:

- Asignar las distancias de s a v con $j = 0$ es $O(n)$.
- Dado $j \in [k]$, $v \in V$, asignar el valor a $D(s, v; j)$ es $O(n)$.

Esto se realiza para cada v en V y cada $j \in [k]$, se tiene un total de $O(kn^2)$ operaciones. Luego la complejidad total es $O(n) + O(kn^2) = O(kn^2)$. Ahora bien, podemos analizar en mayor profundidad este resultado para llegar a una cota más fina, como $O(k(n + m)) = O(kn + km)$, dónde $O(kn)$ viene de llenar la tabla y $O(km)$ es el cálculo de los mínimos, ya que cada término que aparece en algún mínimo se puede poner en correspondencia con un arco del digrafo.

Propuesto 1. Modifique el algoritmo anterior para calcular un paseo mínimo con k arcos, $\forall$ par $(u, v) \in V \times V$. Esto se puede lograr en $O(kn^3) \to O(k(n^2 + mn))$.

Ejemplo 1. Veamos la relación entre encontrar un paseo mínimo y un camino mínimo del vértice s al vértice t, sin restringir la cantidad de arcos a usar. En el siguiente ejemplo (Figura 1.) hay un ciclo negativo, si buscamos el paseo de largo mínimo podríamos pasar infinitas veces por el ciclo para obtener un paseo de largo $-\infty$. En cambio, cuando buscamos un camino entre v_1 y v_2 no se pueden repetir vértices, por lo que no se puede entrar al ciclo, eliminando la posibilidad de sumar largos negativos indefinidamente. Entonces el camino $P_{v_1\text{-}v_2}$ de largo mínimo será $\{v_1, u_1, v_2\}$ con $l(P_{v_1\text{-}v_2}) = 2$, mientras el paseo $W_{v_1\text{-}v_2}$ de largo mínimo tendrá infinitas repeticiones del ciclo y su largo será $l(W_{v_1\text{-}v_2}) = -\infty$

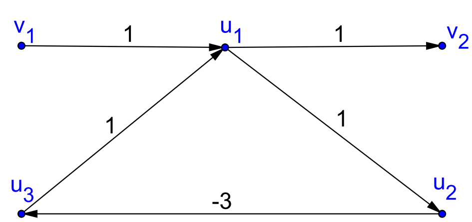

Figura 1.

2. Sobre paseos mínimos con l conservativa

Definición 1 (Función conservativa)**.** Sea $l : E \to \mathbb{R}$ una función de largos, se dice conservativa si el digrafo $G = (V, E)$ no tiene diciclos negativos.

A continuación veremos varios resultados aplicados a funciones de largo conservativas, lo cual nos permitirá presentar al final de la clase el algoritmo de *Bellman-Ford* que entrega un camino de largo mínimo en un grafo dirigido.

Teorema 1. *Sea $l : E \to \mathbb{R}$ una función de largos conservativa, entonces* $\min_{P \in \mathcal{W}_{\leq k}(u,v)} l(P) = \min_{P \in \mathcal{P}_{\leq k}(u,v)} l(P)$

con:

$\mathcal{W}_{\leq k}(u, v)$ *el conjunto de paseos de u a v con a lo más k arcos.*

$\mathcal{P}_{\leq k}(u, v)$ *el conjunto de caminos de u a v con a lo más k arcos.*

Demostración.

- $(\leq)$ Ya que $\mathcal{P}_{\leq k}(u, v) \subseteq \mathcal{W}_{\leq k}(u, v)$, pues todo camino es un paseo, luego se tiene:

$$\min_{P \in \mathcal{W}_{\leq k}(u,v)} l(P) \leq \min_{W \in \mathcal{P}_{\leq k}(u,v)} l(P) \ .$$

- ($\geq$) Sea $P \in \mathcal{W}_{\leq k}(u,v)$ un mínimo con la menor cantidad de arcos. Demostraremos que P es un camino: Si P no es un camino, entonces $P = v_1 v_2 \dots v_{i-1} v_i v_{i+1} \dots v_{j-1} v_j v_{j+1} \dots v_k$, con $v_i = v_j, i \neq j$.

  Notemos que $C = v_i \dots v_j$ es un ciclo, pues es un paseo que empieza y termina en el mismo nodo, por lo que $l(C) \geqslant 0$. Consideremos el paseo $P' = v_1 v_2 \dots v_{i-1} v_i v_{j+1} \dots v_k$, luego

  $$l(P') = l(P) - l(C) \leqslant l(P) \ .$$

  Como se tenía que P era un paseo mínimo, P' también lo es, contradiciendo la minimalidad (en el sentido del número de arcos).

Luego, como se tiene ($\leqslant$) y ($\geqslant$) se cumple la igualdad deseada. ■

Definición 2. Sea $G = (V, E)$ digrafo, $l : E \longrightarrow \mathbb{R}$ conservativo. Se definen:

$$d(u,v) = \min_{P \in \mathcal{P}(u,v)} l(P) = \min_{P \in \mathcal{W}(u,v)} l(P) \ .$$

$$d_k(u,v) = \min_{P \in \mathcal{P}_{\leq k}(u,v)} l(P) = \min_{P \in \mathcal{W}_{\leq k}(u,v)} l(P) \ .$$

Observación 1. De la definición anterior podemos pensar que $d(\cdot,\cdot)$ es una “distancia” con signo, pero:

- Debido a que estamos trabajando con caminos dirigidos podemos tener un camino de u a v pero ninguno de v a u, por lo que $d(u,v) < +\infty$ y $d(v,u) = +\infty$. Por lo tanto, $d(\cdot,\cdot)$ no cumple con la simetría, es decir, $d(v,u) \neq d(v,u)$.
- $d(u,v)$ no es necesariamente positiva como una distancia, podemos tener aristas de largo negativo.

Sin embargo, $d(\cdot,\cdot)$ satisface la desigualdad triangular, enunciemos esto en forma de proposición.

Proposición 1. *Sea $d(\cdot,\cdot)$ la función de la* Definición 2.*, entonces $\forall\, u,\ v,\ w \in V$ se tiene $d(u,v) \leq d(u,w) + d(w,v)$.*

Demostración. Tenemos dos casos, si el lado derecho de la desigualdad es $+\infty$ entonces se tiene directamente la desigualdad. Si esto no sucede, tomamos Q un u-w camino con $l(Q) = d(u,w)$ y R un w-v camino con $l(R) = d(w,v)$, entonces $Q+R$ es un u-v paseo de largo $d(u,w)+d(w,v) \Longrightarrow d(u,v) \leq l(R+Q) = d(u,w)+d(w,v)$, con lo que se tiene lo pedido. ■

Corolario 1. *Sea $l : E \to \mathbb{R}$ una función de largos conservativa, $\mathcal{W}(u,v)$ el conjunto de todos los paseos de u a v, $\mathcal{P}(u,v)$ el conjunto de todos los caminos de u a v y $d(u,v)$ la distancia mínima de u a v. Entonces:*

$$d(u,v) = \min_{P \in \mathcal{W}(u,v)} l(P) = \min_{P \in \mathcal{P}(u,v)} l(P) \ .$$

Demostración. Es una consecuencia directa del teorema anterior cuando $k \longrightarrow \infty$ y de nuestra definición de $d(u,v)$. ■

Teorema 2. *Sea $l : E \to \mathbb{R}$ una función de largos conservativa, $k \geqslant 1$, $P \in \mathcal{P}_{\leq k}(u,v)$ de largo mínimo y $e = (w,v)$ el último arco de P. Entonces:*

$$P' = P \in \mathcal{P}_{\leq k-1}(u,w) \text{ es de largo mínimo.}$$

Demostración. Sea $R \in \mathcal{P}_{\leq k-1}(u,v)$ de largo mínimo. Consideremos:

- (*Caso 1*) Si $R + e$ es un camino, la demostración es idéntica al teorema de la clase anterior.

- (*Caso 2*) Si $R+e$ es un paseo (contiene un ciclo) procedemos de la siguiente manera. Como $R \in \mathcal{P}_{\leq k-1}(u,v)$ entonces R no tiene ciclos, luego $R+e$ posee un único ciclo y se cumple que $v \in R$. Sea Q el subcamino de R que conecta u y v, sea C el ciclo de $R+e$ (ver *Figura 2.*), entonces:

$$\begin{aligned} l(R+e) &= l(Q) + L(C) \\ 0 \leqslant l(C) &= l(R+e) - L(Q) && (2) \\ &\leqslant l(R+e) - l(P) && (3) \\ &= l(R) - l(P-e) \leqslant 0\,. && (4) \end{aligned}$$

(2) : l es conservativa
(3) : Recordar que $P \in \mathcal{P}_{\leq k}(u,v)$, por lo que $l(Q) \geqslant l(P)$
(4) : $(P-e) \in \mathcal{P}_{\leq k-1}(u,v)$, y R es el camino de largo mínimo de $\mathcal{P}_{\leq k-1}$

$$\therefore\ l(R) = l(P-e) = (P')\ .$$

■

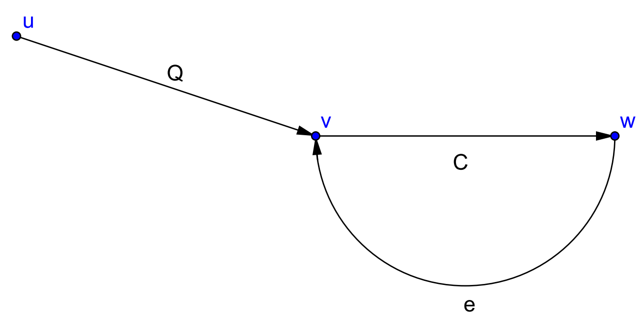

Figura 2.

Teorema 3 (Criterio de optimalidad de Bellman)**.** *Sea l una función de largo conservativa. Si P es un u-v camino de largo mínimo (con $\leq k$ arcos) y Q es un subdicamino (un subcamino dirigido) de P, digamos de a a b con t arcos. Entonces Q es un dicamino de largo mínimo (de entre aquellos con $\leq t$ arcos) de a a b.*

Demostración. Se obtiene directamente del Teorema 2., removiendo arcos al principio y al final de P. ■

Proposición 2. *Sea l conservativa. Entonces*

$$d(u,v) = \min_{w \in N^-(v)} d(u,w) + d(w,v)\,, \tag{5}$$

Además, para $k \geq 1$ se tiene

$$d_k(u,v) = \min\left\{ \min_{w \in N^-(v)} \{d_{k-1}(u,w) + l(w,v)\},\ d_{k-1}(u,v) \right\}\ . \tag{6}$$

Demostración. Primero demostremos (5).

- ($\leq$) Notemos que cada término del lado derecho es el largo de un u-v paseo, dado que l es conservativa, entonces el mínimo paseo/camino tiene largo menor que cualquier otro paseo. Por lo tanto se tiene la desigualdad.

- ($\geq$) Debido al criterio de Bellman tenemos que si $d(u,w)$ es el largo de un u-w camino mínimo, entonces $d(u,w) + l(w,v)$, tomando aquel w que tenga $l(w,v)$ mínimo, será el largo de un u-v camino mínimo. Por lo que se tiene la otra desigualdad. Concluimos, entonces, que la la igualdad es cierta.

La demostración de (6) es análoga. ■

Observación 2. La fórmula (5), *en general*, no sirve para hacer un algoritmo de programación dinámica donde calculemos la tabla con los valores de la "distancia" $d(\cdot,\cdot)$. En cambio, la fórmula (6) sí lo permite.

3. Algoritmo de Bellman-Ford

Finalmente, podemos introducir el algoritmo de *Bellman-Ford* (Schimbel 1955, Ford (1956), Bellman (1958), Moore (1959)), que entrega un s-v camino de largo mínimo en un grafo dirigido, para todo $v \in V$. A continuación se presenta el pseudo-ćodigo del algoritmo, lo estudiaremos en profundidad la próxima clase.

Algoritmo 2: Algoritmo de *Bellman-Ford*

Entrada: $G = (V, E)$ digrafo; $l : E \longrightarrow \mathbb{R}$ conservativa; $s \in V$.

for $v \in V$ **do**

$$d_0(s, v) = \begin{cases} 0 & \text{si } v = s, \\ +\infty & \text{si } v \neq s. \end{cases}$$

end

for $i = 1 \ldots n-1$ **do**

  for $v \in V$ **do**

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

    $w^* = \operatorname{argmín}\limits_{w \in N^-(v)} \{d_{i-1}(s, w) + l(w, v)\}$, si no existe w^* =null.

  end

  if $A < d_{i-1}(s, v)$ **then**

    $d_i(s, v) = A$.

    $\Pi_i(v) = w^*$.

  end

  else

    $d_i(s, v) = d_{i-1}(s, v)$.

    $\Pi_i(v) = \Pi_{i-1}(v)$.

  end

end