Algoritmo de Dijkstra de camino más corto

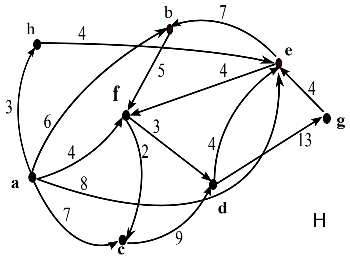
S. Bianchi P. Fekete F. Domingo

[1] Deptartamento de Matemática
Escuela de Ciencias Exactas y Naturales
UNR

4 de octubre de 2021

OUTLINE

Sea $G = (V,E)$ grafo dirigido, conexo y sin lazos.
Si a cada arista $e \in E$ se le asigna $p(e) \in \mathbb{R}^+$, se dice que G es *ponderado*.
Si $x,y \in V$ pero $(x,y) \notin E$ entonces definimos $p(x,y) = \infty$.
Veamos un ejemplo:



En este caso no existe el arco (a,d) por lo tanto $p(a,d) = \infty$.

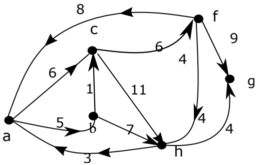
Sea $G=(V,E)$ dirigido, conexo, ponderado y sin lazos.

DEFINICIÓN

Sean $a,b \in V$ tales que existe un camino dirigido de a hacia b en G. Llamamos *camino más corto* de a a b al $a-b$ camino dirigido P para el cual el valor $\sum_{e\in E(P)} p(e)$ es el menor posible.
Definimos la función $d : V\times V \to \mathbb{R}^+ \cup \{\infty\}$ tal que $d(a,b)=\sum_{e\in E(P)} p(e)$ cuando P es el $a-b$ camino más corto, y $d(a,b)=\infty$ cuando no existe un $a-b$ camino en G. El valor $d(a,b)$ es llamado *distancia* de a hacia b. Además definimos $d(a,a)=0$ para todo $a\in V$.

Observación Si P es el $a-b$ camino dirigido más corto entonces P es simple.

Ejemplo:

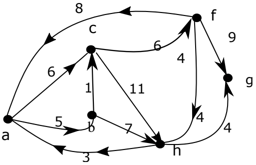


Para calcular $d(a,f)$:

- camino $P : (a,c),(c,f)$ y peso $p(a,c) + p(c,f) = 12$,
- camino $P' : (a,b),(b,c),(c,f)$ y peso $p(a,b) + p(b,c) + p(c,f) = 12$.

Entonces $d(a,f) = 12$.

Ejemplo (cont.):



Para calcular $d(f,a)$:

- arco (f,a) y peso $p(f,a) = 8$,
- camino $P : (f,h),(h,a)$ y peso $p(f,h) + p(h,a) = 7$.

Entonces $d(f,a) = 7$.

Observemos que $d(g,v) = \infty$ para cualquier $v \neq g$.

Camino más corto

Sea $G = (V, E)$ dirigido, conexo, ponderado y sin lazos.

Sea $v_0 \in V$.

Objetivo

1) Determinar $d(v_0, v)$ para cualquier $v \in V$.
2) Obtener camino dirigido más corto de v_0 a v si $d(v_0, v) < \infty$.

Para resolver ambos problemas presentaremos el algoritmo desarrollado por Edsger Wybe Dijkstra (1930-2002)

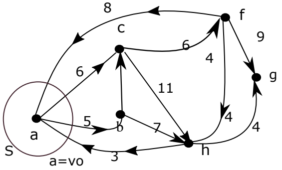
Sea $v_0 \in V$ fijo. Sea $S \subset V$ tal que $v_0 \in S$ y $\bar{S} = V - S$.

Definición

La distancia de v_0 a $\bar{S}$ es

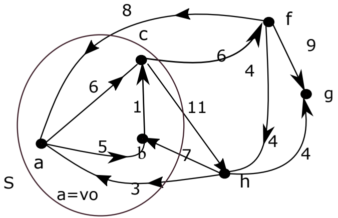
$$d(v_0, \bar{S}) = min\{d(v_0, v) : v \in \bar{S}\}.$$

Veamos un ejemplo:



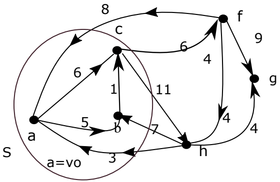
Aquí $S = \{v_0\}$ y $\bar{S} = \{b, c, h, f, g\}$.
Entonces $d(v_0, \bar{S}) = min\{p(a,b), p(a,c)\} = p(a,b) = 5$.

Observación Cuando $d(v_0, \bar{S}) < \infty$, entonces $d(v_0, \bar{S})$ representa la longitud de un camino más corto dirigido de v_0 a un vértice $v \in \bar{S}$. Es decir, existe un vértice $v_{m+1} \in \bar{S}$ tal que $d(v_0, \bar{S}) = d(v_0, v_{m+1})$ y existe un camino simple dirigido $P : (v_0, v_1), (v_1, v_2), \ldots, (v_{m-1}, v_m), (v_m, v_{m+1})$ en G, que además es el camino más corto entre v_0 y v_{m+1}.
Veamos un ejemplo:



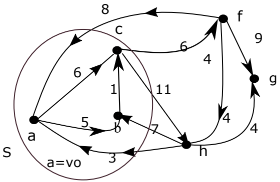
Aquí $S = \{v_0, b, c\}$ y $\bar{S} = \{h, f, g\}$.
Vamos a calcular $d(v_0, \bar{S})$

Ejemplo(cont.)



- camino $P_1 : (a,c),(c,f)$ de costo $p(a,c)+p(c,f) = 12$.
- camino $P_2 : (a,c),(c,h)$ de costo $p(a,c)+p(c,h) = 17$.
- camino $P_3 : (a,b),(b,c),(c,f)$ de costo $p(a,b)+p(b,c)+p(c,f) = 12$.
- camino $P_4 : (a,b),(b,c),(c,h)$ de costo $p(a,b)+p(b,c)+p(c,h) = 17$.

Ejemplo(cont.)



Hay dos caminos más cortos:

- $P_1 : (a,c),(c,f)$ de costo $p(a,c)+p(c,f) = 12$,
- $P_3 : (a,b),(b,c),(c,f)$ de costo $p(a,b)+p(b,c)+p(c,f) = 12$.

En los dos se observa que el último vértice del camino, f en P_1 y P_3, pertenece a $\bar{S}$ y los restantes vértices en cada camino están en S.

Lema

Sea $G=(V,E)$ grafo dirigido y ponderado. Para cada arista $(a,b)\in E$, $p(a,b)\in\mathbb{R}^+$. Si $(a,b)\notin E$, $p(a,b)=\infty$. Sea $v_0\in V$ y sea $S\subset V$ tal que $v_0\in S$ y $\bar{S}=V-S$. Supongamos que $d(v_0,\bar{S})=d(v_0,v_{m+1})$ y $P:(v_0,v_1),(v_1,v_2),\ldots,(v_{m-1},v_m),(v_m,v_{m+1})$ es un camino más corto entre v_0 y v_{m+1} en G.
Entonces:

- $\{v_0,v_1,\ldots,v_m\}\subset S$,
- $P_k:(v_0,v_1),(v_1,v_2),\ldots,(v_{k-1},v_k)$ es el camino más corto entre v_0 y v_k en G, para todo $1\leq k\leq m$.

Demostración

Ejercicio

COROLARIO

Sea $G = (V,E)$ grafo dirigido, conexo, ponderado. Sea $v_0 \in S \subset V$ y $\bar{S} = V - S$. Entonces

$$d(v_0,\bar{S}) = min\{d(v_0,u) + p(u,w) : u \in S \text{ y } w \in \bar{S}\}.$$

Sigue que existen $x \in S$ e $y \in \bar{S}$ tales que $d(v_0,\bar{S}) = d(v_0,x) + p(x,y)$.

El algoritmo comienza con $S_0 = \{v_0\}$ y calcula

$$d(v_0,\bar{S}_0) = min\{d(v_0,u) + p(u,w) : u \in S_0 \text{ y } w \in \bar{S}_0\}.$$

Pero $u \in S_0 = \{v_0\}$ si y sólo si $u = v_0$. Entonces $d(v_0,u) = 0$ y además

$$d(v_0,\bar{S}_0) = min\{p(v_0,w) : w \in \bar{S}_0\}.$$

Si $v_1 \in \bar{S}_0$ es tal que $p(v_0, v_1) = min\{p(v_0, w) : w \in \bar{S_0}\}$, entonces definimos $S_1 = S_0 \cup \{v_1\}$.

El proceso consiste en:
una vez obtenido el conjunto $S_i = \{v_0, v_1, \ldots, v_i\}$, calcular $d(v_0, \bar{S}_i)$.

Si $v_{i+1} \in \bar{S}_i$ es tal $d(v_0, \bar{S}_i) = d(v_0, v_{i+1})$ entonces $S_{i+1} = S_i \cup \{v_{i+1}\}$.

El proceso se detiene cuando $\bar{S}_{n-1} = \emptyset$ o $d(v_0, \bar{S}_i) = \infty$ para algún $0 \leq i \leq n-2$.

El procedimiento utiliza *etiquetas* en cada $v \in V$ de la forma $(L(v), u)$.

Al finalizar, $L(v) = d(v_0, v)$ para todo $v \in V$, $v \neq v_0$ y u es el vértice que precede a v en el camino más corto desde v_0 (si existe).

Al inicio v_0 recibe la etiqueta $(0, -)$ mientras que $v \neq v_0$ es etiquetado por $(\infty, -)$.

Sea $G = (V, E)$ ponderado, dirigido y sin lazos. Sea $n = |V|$ y $v_0 \in V$.

Algoritmo de Dijkstra

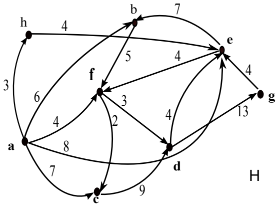
1. $i := 0$, sean $S_0 = \{v_0\}$, $\bar{S}_0 = V - S_0$, $L(v_0) = 0$ y $L(v) = \infty$ para todo $v \in V$, $v \neq v_0$.
   - Si $n = 1$, entonces $V = \{v_0\}$. Fin.
   - Si $n > 1$, ir al Paso 2.
2. Para cada $v \in \bar{S}_i$, hacer $L(v) = min\{L(v), L(u) + p(u, v) : u \in S_i\}$.
   Si $L(v)$ cambia su valor, sea $y \in S_i$ el valor que produce el mínimo y v adquiere la nueva etiqueta $(L(v), y)$.
3. Si cada vértice de $\bar{S}_i$ (para algún $i := 0, \ldots, n-2$) tiene etiqueta $(\infty, -)$, entonces las etiquetas de los vértices del grafo tiene la información buscada.
   Si existe $v \in \bar{S}_i$ con $L(v) < \infty$, entonces:

Algoritmo de Dijkstra

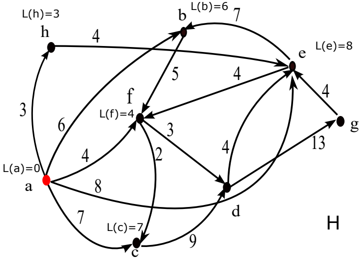
3 (cont.)

- Sea $v_{i+1} \in \bar{S}_i$ tal que $L(v_{i+1}) = min\{L(v) : v \in \bar{S}_i\}$ (si hay más de uno elegir cualquiera).
- Sea $S_{i+1} = S_i \cup \{v_{i+1}\}$.
- $i := i+1$. Si $i = n-1$, las etiquetas del grafo tienen la información buscada. Si $i < n-1$ ir al Paso 2.

Vamos a aplicar el algoritmo en este grafo:

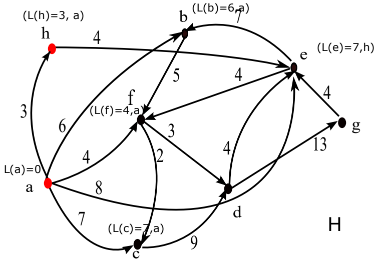


Paso 1 Inicialización $i := 0$. $S_0 = \{a\}$, $L(a) = 0$ y $L(v) = \infty$ para $v \neq a$.
$n = 8 > 1$, ir Paso 2
Paso 2 Para $v \in \bar{S}_0 = \{b, c, d, e, f, g, h\}$ calcular $L(v) = min\{\infty, p(a, v)\}$.



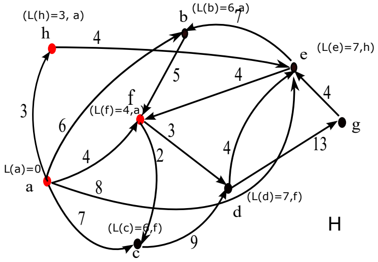
Observar que $L(d) = L(g) = \infty$ y entonces estas etiquetas no cambian.
Paso 3 existe $v \in \bar{S}_0$ con $L(v) < \infty$. $L(h) = min\{L(v) : v \in \bar{S}_0\}$.
$S_1 = \{a, h\}$. $i := 1$. $1 < 8 - 1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_1 = \{b, c, d, e, f, g\}$ calcular
$L(v) = min\{L(v), L(u) + p(u, v) : u \in S_1\}$.



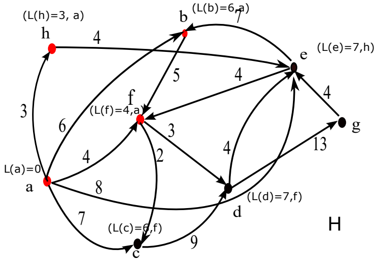
Observar que todas las etiquetas quedan iguales salvo
$L(e) = min\{L(e), L(h) + p(h, e)\} = min\{8, 3+4\} = 7$ y h es el predecesor.
Paso 3 existe $v \in \bar{S}_1$ con $L(v) < \infty$. $L(f) = min\{L(v) : v \in \bar{S}_1\}$.
$S_2 = \{a, h, f\}$. $i := 2$. $2 < 8 - 1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_2 = \{b,c,d,e,g\}$ calcular
$L(v) = min\{L(v), L(u) + p(u,v) : u \in S_2\}$.



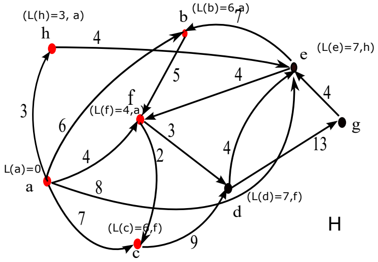
Observar que todas las etiquetas que cambian son
$L(c) = min\{L(c), L(f) + p(f,c)\} = min\{7, 4+2\} = 6$ y f es el predecesor y
$L(d) = min\{L(d), L(f) + p(f,d)\} = min\{\infty, 4+3\} = 7$ y f es el predecesor.
Paso 3 existe $v \in \bar{S}_2$ con $L(v) < \infty$. $L(b) = L(c) = min\{L(v) : v \in \bar{S}_2\}$.
Elijo a b. $S_3 = \{a,b,h,f\}$. $i := 3$. $3 < 8-1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_3 = \{c, d, e, g\}$ calcular
$L(v) = min\{L(v), L(u) + p(u, v) : u \in S_3\}$.



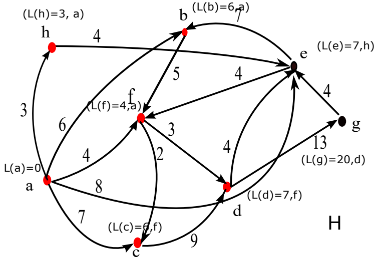
Observar que ninguna etiqueta cambia, ya que
$L(f) = min\{L(f), L(b) + p(b, f)\} = min\{4, 6 + 5\} = 4$.
Paso 3 existe $v \in \bar{S}_3$ con $L(v) < \infty$. $L(c) = min\{L(v) : v \in \bar{S}_3\}$.
$S_4 = \{a, b, c, h, f\}$. $i := 4$. $4 < 8 - 1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_4 = \{d, e, g\}$ calcular
$L(v) = min\{L(v), L(u) + p(u, v) : u \in S_4\}$.



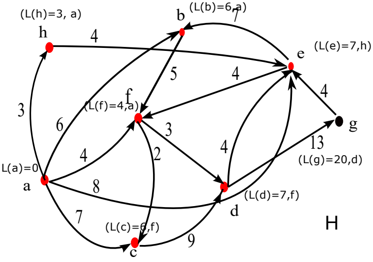
Observar que ninguna etiqueta cambia, ya que
$L(d) = min\{L(d), L(c) + p(c, d)\} = min\{7, 6 + 9\} = 7$.
Paso 3 existe $v \in \bar{S}_4$ con $L(v) < \infty$. $L(d) = L(e) = min\{L(v) : v \in \bar{S}_4\}$.
Elijo d. $S_5 = \{a, b, c, d, h, f\}$. $i := 5$. $5 < 8 - 1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_5 = \{e, g\}$ calcular $L(v) = min\{L(v), L(u) + p(u, v) : u \in S_5\}$.



Observar que cambia la etiqueta
$L(g) = min\{L(g), L(d) + p(d, g)\} = min\{\infty, 7 + 13\} = 20$ y el predecesor es d.
Pero no cambia $L(e) = min\{L(e), L(d) + p(d, e)\} = min\{7, 7 + 4\} = 7$
Paso 3 existe $v \in \bar{S}_5$ con $L(v) < \infty$. $L(e) = min\{L(v) : v \in \bar{S}_5\}$.
$S_6 = \{a, b, c, d, e, h, f\}$. $i := 6$. $6 < 8 - 1 = 7$. ir al Paso 2.

Paso 2 Para $v \in \bar{S}_6 = \{g\}$ calcular $L(v) = min\{L(v), L(u) + p(u,v) : u \in S_6\}$.



Observar que no hay cambios de etiquetas ya que
$L(b) = min\{L(b), L(e) + p(e,b)\} = min\{6, 7+7\} = 6$ y
$L(f) = min\{L(f), L(e) + p(e,f)\} = min\{4, 7+4\} = 4$
Paso 3 existe $v \in \bar{S}_6$ con $L(v) < \infty$. $L(g) = min\{L(v) : v \in \bar{S}_6\}$.
$S_7 = \{a, b, c, d, e, h, f, g\}$. $i := 7$. $7 = 8 - 1 = 7$. Fin

DEFINICIÓN

Dado $G = (V,E)$ ponderado con $n = |V|$, sea $f(n)$ el número máximo de operaciones que realiza el Algoritmo para encontrar el camino más corto.

Después del paso de inicialización, el procedimiento realiza $n-1$ iteraciones. (En la iteración i con $i = 1,\ldots,n-1$, se establece el i-ésimo vértice más cercano a v_0.)

En el Paso 2, se calcula $L(v) = min\{L(v), L(u) + p(u,v) : u \in S_i\}$.
Es decir, para cada $u \in S_i$

- se realizan a lo sumo $n-1$ sumas y
- se realizan $n-1$ comparaciones.

Se realizan un total de $2(n-1)$ cómputos para cada $v \in \bar{S}_i$ ($|\bar{S}_i| \leq n-1$).
Se obtienen a lo sumo $2(n-1)^2$ operaciones.

En el Paso 3, calculamos $min\{L(v) : v \in \bar{S}_i\}$ (recordar $|\bar{S}_i| \leq n-1$).

Esto requiere a lo sumo $n-2$ comparaciones.
Entonces

$$f(n) \leq (n-1)[2(n-1)^2 + (n-2)].$$

Que se expresa como:

$$f(n) \in O(n^3).$$

El algoritmo puede mejorarse de modo de obtener

$$f(n) \in O(n^2).$$