MA37A Optimización
Camino de costo mínimo

Profesor: Héctor Ramirez
Auxiliares: Francisco Jara, Oscar Peredo

28 de septiembre de 2005

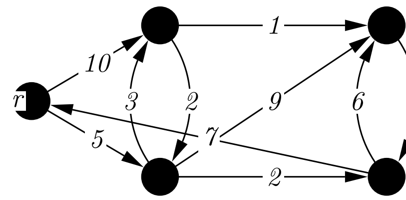
El problema del camino de costo mínimo recibe como entradas un grafo dirigido $G = (V, E)$ (con V el conjunto de vértices y $E \subseteq V \times V$ el conjunto de arcos), un nodo raíz $r \in V$ y una función de costos $c : E \to \mathbb{R}^+ \cup \{0\}$.
El objetivo es encontrar, para cada $v \in V$, un camino dirigido desde r hasta v de costo mínimo.
Para ello ocuparemos el algoritmo de **Dijkstra**, que guarda un arreglo p con el nodo predecesor en el camino óptimo, es decir, $p[v] = u$ significa que el arco $(u, v) \in E$ pertenece al camino de costo mínimo. Además guarda el costo total de llegar desde la raíz a un nodo v utilizando el camino óptimo en un arreglo y.

Algorithm 1 Dijkstra($G = (V, E), c, r$)

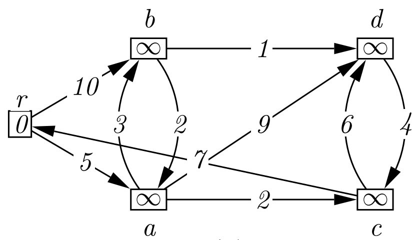
▷ INICIALIZACIÓN
for $v \in V$ **do**
    $y[v] = \infty$
    $p[v] = NULO$
end for
$y[r] = 0$
▷ ITERACIÓN
$S \leftarrow \{r\}$
$Q \leftarrow V$
while $Q \neq \{r\}$ **do**
    for v tal que $(s, v) \in (S \times Q - \{r\}) \cap E$ **do**
        $y[v] \leftarrow \min\{y[v], y[s] + c(s, v)\}$ ▷ Actualizar valores de y en los nodos de llegada partiendo de S
    end for
    Encontrar $u \in Q - \{r\}$ y $s \in S$ tal que $c(s, u)$ sea mínimo
    $p[u] = s$
    $S \leftarrow S \cup \{u\}$
    $Q \leftarrow Q \backslash \{u\}$
end while

Las estructuras para guardar la información, en este caso Q,S,p e y son arreglos.
Este algoritmo sólo sirve en el caso $c(E) \subset [0, \infty)$. Para el caso $c(E) \subset (-\infty, \infty)$ se puede revisar el algoritmo de Bellman-Ford.

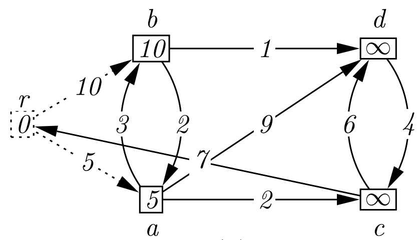
Problema 1. *Encontrar el camino de costo mínimo en el siguiente grafo:*



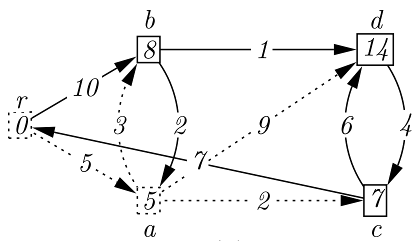
Solución 1. *Dentro de los nodos se representa el valor de $y[v]$, que representa la suma de los costos del camino mínimo hasta el nodo v:*



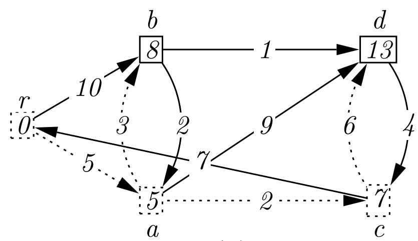
(1)



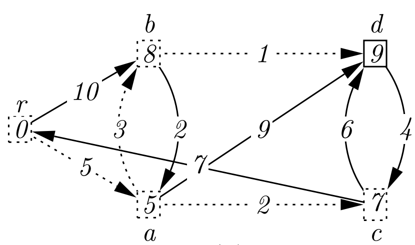
(2)



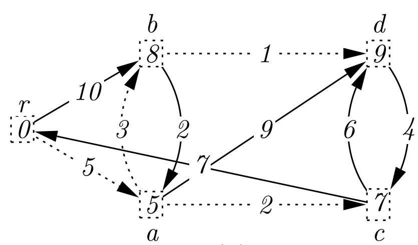
(3)



(4)



(5)



(6)

(1) Se inicializa el arreglo y con $y[r] = 0$ e $y[v] = \infty$ para $v \neq r$. Se setean los conjuntos $S = \{r\}, Q =$

$\{a, b, c, d\}$

(2) *Primera iteración.*
Se actualizan los valores de $y[a] = 5$ *e* $y[b] = 10$*, pues son los nodos que se pueden alcanzar partiendo de algun nodo de* $S = \{r\}$ *(lineas 11,12).*
Se escoge $u = a$ *y* $s = r$*, pues* $c(r, a) < c(r, b)$ *(linea 14).*
Se actualiza $p[a] = r$*,* $S \leftarrow \{r\} \cup \{a\}$ *y* $Q \leftarrow \{r, a, b, c, d\} \backslash \{a\}$ *(lineas 15,16,17).*

(3) *Segunda iteración.*
Se actualizan los valores de $y[b] = 8$*,*$y[c] = 7$ *e* $y[d] = 14$*, pues son los nodos que se pueden alcanzar partiendo de algun nodo de* $S = \{r, a\}$ *(siempre se actualiza con el minimo valor posible, en este caso pudo haberse actualizado el nodo b como* $y[b] = 10$*, pero existe una actualización menor, que utiliza el arco* (a, b) *y resulta* $y[b] = 5 + c(a, b) = 8$*) (lineas 11,12).*
Se escoge $u = c$ *y* $s = a$*, pues* $c(a, c) = \min\{3, 9, 2\} = 2$ *(linea 14).*
Se actualiza $p[c] = a$*,* $S \leftarrow \{r, a\} \cup \{c\}$ *y* $Q \leftarrow \{r, b, c, d\} \backslash \{c\}$ *(lineas 15,16,17).*

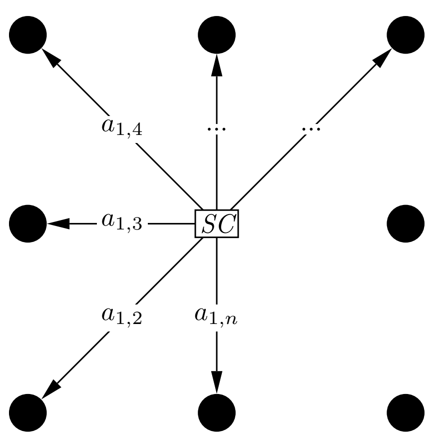
(4) *Tercera iteración.*
Se actualizan los valores de $y[b] = 8$ *e* $y[d] = 13$*, pues son los nodos que se pueden alcanzar partiendo de algun nodo de* $S = \{r, a, c\}$ *(lineas 11,12).*
Se escoge $u = b$ *y* $s = a$*, pues* $c(b, a) = \min\{3, 6\}$ *(linea 14).*
Se actualiza $p[b] = a$*,* $S \leftarrow \{r, a, c\} \cup \{b\}$ *y* $Q \leftarrow \{r, b, d\} \backslash \{b\}$ *(lineas 15,16,17).*

(5) *Cuarta iteración.*
Se actualizan el valor de $y[d] = 9$*, pues es el único nodo que se puede alcanzar partiendo de algun nodo de* $S = \{r, a, b, c\}$ *(lineas 11,12).*
Se escoge $u = d$ *y* $s = b$*, pues* $c(d, b) = \min\{1\}$ *(linea 14).*
Se actualiza $p[d] = b$*,* $S \leftarrow \{r, a, b, c\} \cup \{d\}$ *y* $Q \leftarrow \{r, d\} \backslash \{d\}$ *(lineas 15,16,17).*

(6) *Se sale de la iteración pues* $Q = \{r\}$*.*

Problema 2. *En una empresa se instala una red de* N *computadores, todos conectados entre si y además conectados a un supercomputador central (SC).*
Para testear la velocidad de la comunicación en la red, los técnicos deciden probar la eficiencia del sistema enviando $N-1$ *mensajes en forma paralela (el SC tiene N procesadores) desde el SC hacia los demás equipos. Desafortunadamente, el SC esta conectado solamente a uno de los equipos.*
Describa de que manera se podría realizar el envío de mensajes de tal forma que se minimize el tiempo entre el envío por parte del SC y la recepción por parte de un equipo (considere que el tiempo de demora del mensaje es $a_{i,j}$ *entre el equipo* i *y* j*. El SC es el equipo* 1*).*

Solución 2. *Si el SC estuviera conectado a todos los equipos, la situación seria de la forma:*



y el tiempo de espera sería $\max_{i=2,\ldots,n} \{a_{1,i}\}$*, pues como los mensajes se envian en paralelo, la demora total será la demora máxima.*

Ahora, como el SC está conectado a un solo equipo, supongamos al equipo 2, una opción es que SC envíe un mensaje a cada equipo a través del 2, lo cual demoraría $\sum_{i=2}^{n} a_{2,i}$*, pues solo tiene un procesador y no puede realizar tareas en paralelo, solo de manera secuencial.*
Sin embargo, si 2 le envia un mensaje a 3, 3 tambien podrá enviar el mensaje a otro equipo y sucesivamente los que reciban en mensaje. Para encontrar el camino que toma menos tiempo en enviar el mensaje, podemos utilizar Dijsktra, pues se forma un grafo dirigido, y nos interesa encontrar el camino de costo mínimo entre un equipo i y el equipo 2, que actua como despachador.

Problema 3 (Propuesto). *Dado un grafo* $G = (V, E)$*, donde cada arco tiene asociado un valor* $0 \leq r(u, v) \leq 1$*, el cual representa la probabilidad de que un mensaje pueda atravezar el arco* (u, v)*, y suponiendo además que esas probabilidades son independientes, entregue un algoritmo eficiente que encuentre el camino mas confiable (con mayor probabilidad) entre dos nodos* u_0 *y* v_0*.*
Indicación: Utilize la función $-\log$ *y el algoritmo de Dijkstra.*