

Programación dinámica: Bellman-Ford

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski

✉ vpodberezski@fi.uba.ar

Camino mínimo

Dado

grafo G dirigido y ponderado (con costos positivos)
de n nodos no aislados y M ejes

Dados dos nodos

“s” inicial

“t” final

Encuentra el camino mínimo que los une

Solución

Utilización de algoritmo Dijkstra

Algoritmo greedy

Complejidad $O([n+m]\log n)$

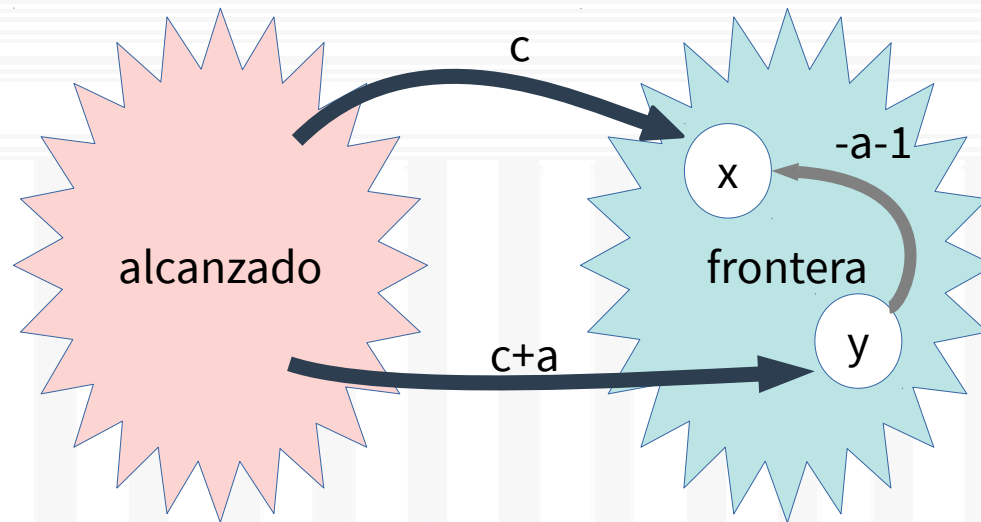
Qué pasa si los existen costos negativos?

Utilizar una arista puede disminuir el costo del camino total

Elección Greedy – con costos negativos

En cada iteración selecciona el menor costo disponible

Puede esto resultar contraproducente?



Elegir el menor costo actual, puede evitar caminos de menor costo general

La solución greedy (Dijkstra) no es optima

Ciclos negativos

Sea

Nodos $A = \{a_1, a_2, \dots, a_r\} \in G$

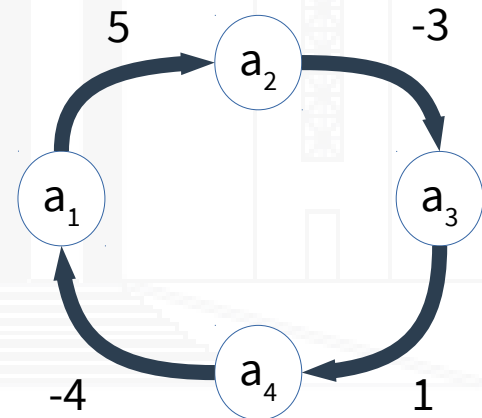
Tal que

Existen las aristas $a_i \rightarrow a_{i+1 \bmod r}$ para todo $a \in A$ conformando un ciclo C

$$\sum_{i=0 \text{ to } r} w(a_i \rightarrow a_{i+1 \bmod r}) < 0$$

Llamaremos

ciclo negativo “C”



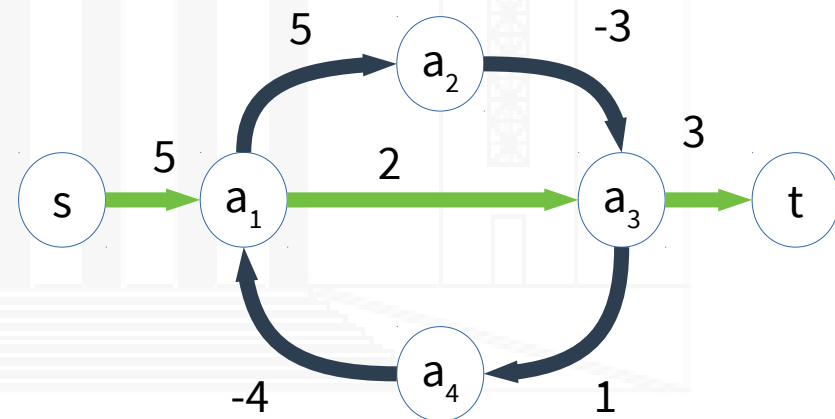
Ciclos negativos y caminos mínimos

La existencia de un ciclo negativo

dentro de un camino s-t

Genera un camino mínimo de $-\infty$

Podemos pensarlo como un loop infinito



Por Fuerza bruta

Para grafo conderado G sin ciclos negativos

Calcular camino minimo s-t

Se deben calcular

Todos costos de los caminos posibles s-t de longitud 1

Todos costos de los caminos posibles s-t de longitud 2

...

Todos costos de los caminos posibles s-t de longitud $n-1$

El camino mínimo tendrá longitud $n-1$ como máximo

Bellman-Ford

Algoritmo para camino mínimo

Para grafos con ponderados sin ciclos negativos

Utiliza programación dinámica

Propuesto por

Alfonso Shimbel (1955)

Lester Ford Jr. (1956)

Edward F. Moore (1957)

Richard Bellman (1958)

Análisis

Para llegar desde “s” a un nodo n_i

puede haber utilizado diferentes caminos (nodos y longitudes diferentes)

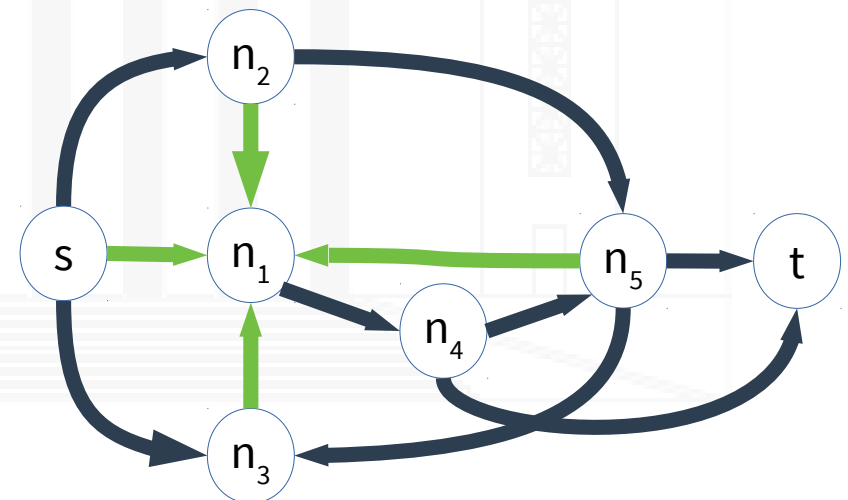
Unicamente puedo llegar desde sus nodos que lo conectan ($\text{pred}[n_i]$)

A los que tambien puedo llegar por diferentes caminos

Para llegar desde “s” a n_i en j pasos

Tengo que haber llegado a sus predecesores en $j-1$ pasos

Y para llegar a estos, se debe haber llegado a sus predecesores en $j-2$ pasos

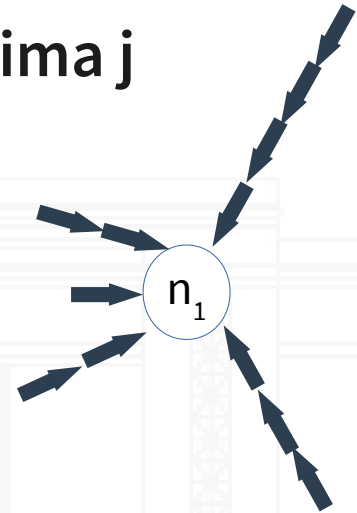


Análisis (cont.)

Al camino mínimo hasta el nodo n_i con longitud máxima j

Lo llamaremos $\text{minPath}(n_i, j)$

Es el mínimo camino que llega a n_i con longitud entre 0 a j



Es el mínimo entre los caminos mínimos que llegan a sus predecesores en máximo $j-1$ pasos + el peso de llegar a n_i

$$\text{minPath}(n_i, j) = \min \left\{ \begin{array}{l} \text{minPath}(n_i, j-1) \\ \min \{ \text{minPath}(n_x, j-1) + w(n_x, j-1) \} \end{array} \right\} \quad \text{con } n_x \in \text{pred}(n_i)$$

Subproblemas y recurrencia

El camino mínimo s-t

No contiene ciclos internos. (no hay ciclos negativos en el grafo)

Agregar ciclos solo aumenta el peso del camino

$\text{minPath}(t, n-1)$ con n la cantidad de nodos en el grafo

Reccurencia

$$\text{minPath}('s', j) = 0$$

$$\text{minPath}(n_{i,0}) = +\infty \text{ con } n_i \neq s$$

$$\text{minPath}(n_i, j) = \min \left\{ \begin{array}{l} \text{minPath}(n_i, j-1) \\ \min \{ \text{minPath}(n_x, j-1) + w(n_x, j-1) \} \end{array} \right\} \text{ con } n_x \in \text{pred}(n_i)$$

Solución iterativa

Llamaremos $OPT[l][v]$

Al camino mínimo de “s” al nodo n_v
con máxima longitud l

El nodo “s” se encuentra en $v=0$

El nodo “t” se encuentra en $v=n$

Lo almacenaremos en una matriz de
 $n \times n$

```
Desde  $l=0$  a  $n-1$   
     $OPT[l][0] = 0$   
  
Desde  $v=0$  a  $n-1$   
     $OPT[0][v] = +\infty$   
  
Desde  $l=1$  a  $n-1$  // max longitud del camino  
    Desde  $v=1$  a  $n$  // nodo  
         $OPT[l][v] = OPT[l-1][v]$   
        Por cada  $p$  predesor de  $v$   
            si  $OPT[l][v] > OPT[l-1][p] + w(p,v)$   
                 $OPT[l][v] = OPT[l-1][p] + w(p,v)$   
  
Retornar  $OPT[n-1,n]$ 
```

Complejidad temporal

El primer loop esta acotado por n

El acceso de los predecesor se puede hacer en forma eficiente si se invierte el grafo

El segundo loop y el “por cada predecesor” se ejecuta m veces

(m = número de aristas)

$O(m \times n)$

```
Desde l=0 a n-1
    OPT[l][0] = 0

Desde v=0 a n-1
    OPT[0][v] =  $+\infty$ 

Desde l=1 a n-1    // max longitud del camino
    Desde v=1 a n  // nodo
        OPT[l][v] = OPT[l-1][v]
        Por cada p predecesor de v
            si OPT[l][v] > OPT[l-1][p] + w(p,v)
                OPT[l][v] = OPT[l-1][p] + w(p,v)

Retornar OPT[n-1,n]
```

Complejidad espacial

La matriz ocupa $m \times n$

El grafo se puede mantener en $n+m$
(lista de adjacencias)

$O(m \times n)$

Se puede mejorar, solo es necesario
los resultado de la última iteración
principal.

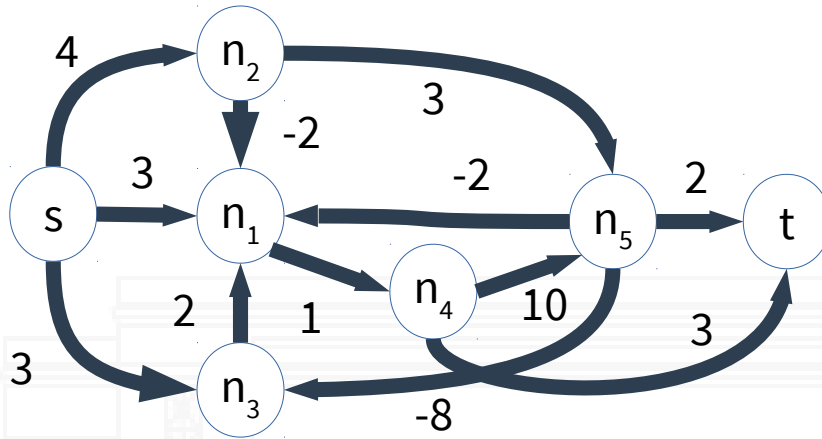
```
Desde l=0 a n-1
    OPT[l][0] = 0

Desde v=0 a n-1
    OPT[0][v] =  $+\infty$ 

Desde l=1 a n-1    // max longitud del camino
    Desde v=1 a n  // nodo
        OPT[l][v] = OPT[l-1][v]
        Por cada p predesor de v
            si OPT[l][v] > OPT[l-1][p] +
                w(p,v)
                OPT[l][v] = OPT[l-1][p] +
                    w(p,v)

Retornar OPT[n-1,n]
```

Ejemplo



	s	n ₁	n ₂	n ₃	n ₄	n ₅	t
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	3	4	3	$+\infty$	$+\infty$	$+\infty$
2	0	2	4	3	4	7	$+\infty$
3	0	2	4	-1	3	7	7
4	0	1	4	-1	3	7	7
5	0	1	4	-1	2	7	7
6	0	1	4	-1	2	7	5

Desde $l=0$ a $n-1$
 $OPT[l][0] = 0$

Desde $v=0$ a $n-1$
 $OPT[0][v] = +\infty$

Desde $l=1$ a $n-1$ // max longitud del camino

Desde $v=1$ a n // nodo

$OPT[l][v] = OPT[l-1][v]$

Por cada p predesor de v

si $OPT[l][v] > OPT[l-1][p] + w(p,v)$

$OPT[l][v] = OPT[l-1][p] + w(p,v)$

Retornar $OPT[n-1,n]$

Camino mínimo:

s-n₂-n₅-n₃-n₁-n₄-t

Costo:

5

Reconstruir las elecciones

Basta agregar un vector

“pred” De longitud n

Que almacene en la posición i

Desde que predecesor se llega actualmente a ni con el menor costo desde s

Al finalizar la ejecución

Se itera desde $pred[n]$ con $n = t$

$pred[pred[n]]$, $pred[pred[pred[n]]]$

Hasta llegar a “ s ”

¿Qué pasa si hay ciclos negativos?

Se puede encontrar un camino desde “s” a un n_i

Con longitud “n” menor al los caminos con máxima longitud $n-1$
(incluso cuanto mayor la longitud a n más disminuirá el camino mínimo)

Ejecutando Bellman-Ford

una iteración principal más (longitud camino “n”)

si cambia el íinimo de al menos un nodo

Entonces el grafo tiene un ciclo negativo.



Presentación realizada en Abril de 2020