ALGORITMO DE BELLMAN - FORD

Origen V_0 Destinos $V_1, \ldots, V_n$ Grafo $c(i,j)$ $i,j \in 0..n$

Inicialización $F(0) = 0$, $F(i) = \infty$ $\forall i \in 1..n$

<u>Algoritmo</u>

i) n pasadas intentando relajar con todos los arcos de G

ii) Una vuelta más: Se puede seguir relajando $\Leftrightarrow$ G contiene un ciclo accesible desde V_0 con coste negativo

<u>Relajación con arco (i,j)</u>: $(F(i) < \infty \wedge F(j) > F(i) + c(i,j)) \Rightarrow$
$$F(j) := F(i) + c(i,j)$$

<u>Invariante del algoritmo</u>: $F(i) < \infty \Rightarrow \exists\, cm(V_0, V_i)$ con $c(cm) = F(i)$

<u>Lema</u>: Cualquier algoritmo que cumpla el invariante y trate de relajar sucesivamente (aunque quizás con otros pasos intermedios) utilizando los arcos de un camino $V_0, V_{i_1}, \ldots, V_{i_k}$ con coste c cumple a partir del momento que lo haya hecho que $F(i_k) \leq c$

<u>Demost</u>: Inmediata por inducción sobre k utilizando el hecho de que $F(j)$ no puede crecer jamás durante la ejecución del algoritmo.

<u>Corolario</u>: Al terminar la etapa i) del algoritmo se cumple $\forall i \in 0..n$ que $F(i)$ es menor o igual que el coste de cualquier camino simple desde V_0 a V_i.

<u>Demost</u>: Los caminos en cuestión tienen a lo sumo n arcos, por lo que al final del algoritmo podemos aplicar el lema sobre todos ellos utilizando la k-ésima pasada para cubrir cada k-ésimo arco del camino.

<u>Lema</u>: Si G no contiene ciclos de coste negativo accesibles desde V_0, y V_i es accesible desde V_0, entonces existe un camino simple de V_0 a V_i que tiene coste mínimo entre todos los caminos de V_0 a V_i.

<u>Demost</u>: Todo ciclo en un camino de V_0 a V_i puede ser eliminado hasta obtener un camino simple, sin que aumente nunca su coste. El mínimo se alcanza pues hay un número finito de caminos simples en G.

Teorema de corrección

a) En la vuelta final del algoritmo se puede seguir relajando si y sólo sí existe un ciclo de coste negativo accesible desde V_0.

b) Si no hay ningún ciclo de coste negativo accesible desde V_0, al final del algoritmo $F(i)$ nos da el coste del camino mínimo desde V_0 a V_i para cada vértice accesible, y seguirá valiendo ∞ para los demás.

Demost: a) $\boxed{\Rightarrow}$ Tras las n vueltas, $F(i)$ es siempre menor o igual que el coste de cualquier camino simple de V_0 a V_i. Por tanto, si podemos seguir relajando es porque hay un camino con ciclos más barato que todos ellos, pero eso sólo puede suceder si contiene un ciclo (accesible desde V_0) con coste negativo.

$\boxed{\Leftarrow}$ Si $\langle V_{i_1}, \dots, V_{i_k} \rangle$ es un ciclo de coste negativo accesible desde V_0 y en ii) no podemos seguir relajando, tomando $k+1 = 1$, tenemos que para todo $j \in 1..k$ hemos comprobado que $F(i_{j+1}) \leq F(i_j) + c(i_j, i_{j+1})$, lo que sumando todas las desigualdades y cancelando nos lleva al absurdo $0 \leq \sum_{j=1}^{k} c(i_j, i_{j+1})$. Los valores $F(i_j)$ son finitos, pues todo V_{i_j} es accesible vía un camino simple desde V_0.

b) Consecuencia inmediata del invariante del algoritmo, el lema anterior, el corolario, y la implicación $\boxed{\Rightarrow}$, que utilizamos para concluir que el algoritmo terminará sin lanzar la excepción correspondiente a la detección de ciclos negativos accesibles.

Observación: Los posibles ciclos negativos no accesibles desde V_0 no presentan problemas, pues nunca formarán parte de caminos desde V_0, pero tampoco darán pie a relajación alguna, pues los valores $F(i_j)$ correspondientes a los vértices que los forman permanecerán siempre iguales a ∞.