Dijkstra (ricerca operativa)

L'algoritmo di Dijkstra trova un albero ottimo nel caso in cui $c_{ij} \geq 0$ per ogni $(i, j) \in A$.

Ad ogni iterazione l'algoritmo mantiene:

- un albero di copertura radicato in r e orientato (memorizzato in un vettore p di predecessori)
- un vettore π di potenziali dei nodi corrispondente all'albero memorizzato in p
- un insieme U di nodi tale che gli archi che potrebbero violare le condizioni di Bellman devono essere uscenti da nodi di U.

Algoritmo

0. Inizializza

$$p_i = \begin{cases} 0 & \text{se } i = r \\ -1 & \text{se } i \neq r \end{cases} \qquad \pi_i = \begin{cases} 0 & \text{se } i = r \\ +\infty & \text{se } i \neq r \end{cases} \qquad U = N$$

(l'albero inizialmente è costituito solo da archi fittizi da r agli altri nodi)

1. Se $U = \emptyset$ allora stop

2. Seleziona un nodo $u \in U$ con potenziale minimo: $u = \arg\min_{i \in U} \pi_i$

3. (Controlla le condizioni di Bellman sugli archi uscenti da u)
Per ogni arco $(u, v) \in A$:
se $\pi_v > \pi_u + c_{uv}$ allora $p_v = u$, $\pi_v = \pi_u + c_{uv}$

4. $U = U \setminus \{u\}$ e torna al passo 1.

Teorema

Se $c_{ij} \geq 0$ per ogni $(i, j) \in A$, allora l'algoritmo di Dijkstra trova un albero dei cammini minimi dopo $|N|$ iterazioni.

Dimostrazione (sketch)

- durante l'esecuzione dell'algoritmo π è il vettore dei potenziali dei nodi associato all'albero memorizzato nel vettore p
- il potenziale di ogni nodo non può mai aumentare durante l'esecuzione dell'algoritmo
- quando un nodo u viene estratto da U, il suo potenziale diventa definitivo e gli archi uscenti da u soddisfano le condizioni di Bellman fino alla fine dell'algoritmo

$\square$