Note su algoritmo di Dijkstra e sua analisi

Nel seguito si presentano alcune considerazioni alla base degli algoritmi per la costruzione dell'albero dei cammini minimi da una sorgente a tutti i nodi (raggiungibili).

Nel seguito supponiamo di avere un grafo $G = (V, E)$, tale che $|V| = n$ e $|E| = m$. G può essere diretto o meno. Alcune proprietà sono generali e si applicano sia a grafi diretti che non diretti, altre soltanto a grafi non diretti. Nel seguito consideriamo grafi diretti, per cui tutto ciò che dimostriamo si applica anche al caso non diretto (in quanto quello diretto è più generale). G è un grafo pesato. Nel seguito indichiamo con $weight(e)$ il peso dell'arco (in generale diretto) e. Se $e = (u, v)$, con un leggero abuso di notazione scriviamo $weight(u, v)$ al posto di $weight((u, v))$.

Condizioni di ottimalità per i cammini minimi.

Quella che discutiamo di seguito è la proprietà fondamentale dei cammini minimi, alla base del funzionamento degli algoritmi per cammini minimi oggetto delle nostre lezioni. Nel seguito supponiamo di essere interessati alle distanze minime di tutti i vertici da un vertice sorgente $s \in V$ (albero dei cammini minimi con sorgente s). Per ogni $v \in V$, indichiamo con $d(s, v)$ la lunghezza di *qualche cammino* da s a v. Dimostreremo la seguente

Proposizione 1. I valori $d(s, v)$ (per ogni $v \in V$) corrispondono alle distanze dei vertici da s (ossia, per ogni v, alla lunghezza di un cammino minimo da s a v) *se e solo se* vale la condizione seguente, per ogni $v \in V$ e per ogni arco $e = (u, v)$ diretto da u a v:

$$d(s, v) \leq d(s, u) + weight(u, v) \qquad (1)$$

Prova. Dimostriamo prima la necessità (solo se) e poi la sufficienza (se).

La condizione (1) è necessaria. Si supponga per assurdo che la condizione (1) non sia verificata per qualche nodo, ma che comunque, per ogni $w \in V$, $d(s, w)$ sia la distanza (lunghezza di un cammino minimo) da s a w. Per ogni $w \in V$, indichiamo con $P(s, w)$ un cammino minimo da s a w, quindi di lunghezza complessiva $d(s, w)$. Esisteranno allora un vertice v e un arco $e = (u, v)$ tali che $d(s, v) > d(s, u) + weight(u, v)$. In tal caso, possiamo sostituire il cammino minimo $P(s, v)$ da s a v con un nuovo cammino, ottenuto concatenando il cammino $P(s, u)$ e l'arco e. Il cammino risultante avrà lunghezza complessiva pari esattamente a $d(s, u) + weight(u, v) < d(s, v)$, contro l'ipotesi che $P(s, v)$ fosse un cammino minimo da s a v.

La condizione (1) è sufficiente. Si supponga che la condizione (1) sia verificata per ogni $v \in V$. Vogliamo dimostrare che, per ogni v, $d(s, v)$ non è maggiore della distanza minima da s a v. A tale scopo, consideriamo il generico nodo w e consideriamo il cammino P da s a w che abbia la minima lunghezza possibile. Sia ℓ la lunghezza di P. Vogliamo dimostrare che, se la condizione (1) è verificata per ogni nodo, allora avremo $d(s, w) \leq \ell$. A tale scopo, sia $s = v_0, v_1, \ldots, v_k = w$ la sequenza dei vertici attraversati da P (per cui gli archi sono $(v_0, v_1), \ldots, (v_{k-1}, v_k)$). Poiché per ognuno dei vertici attraversati la condizione (1) è soddisfatta per ipotesi avremo:

$$d(s, w) = d(s, v_k) \leq d(s, v_{k-1}) + weight(v_{k-1}, v_k) \leq d(s, v_{k-2}) + weight(v_{k-2}, v_{k-1}) + weight(v_{k-1}, v_k)$$
$$\ldots \leq \sum_{i=1}^{k} weight(v_{i-1}, v_i) = \ell.$$

Abbiamo cioè dimostrato che, se le condizioni (1) sono soddisfatte per ogni $w \in V$, allora le $d(s, w)$ non possono essere maggiori delle rispettive distanze minime da s. Ma ℓ è per definizione pari alla lunghezza di un cammino minimo da s a w e quindi deve essere $d(w) \geq \ell$. Insieme, queste due condizioni implicano $d(s, w) = \ell$. Ciò conclude la prova della sufficienza.

Un'altra proprietà semplice e intuitiva, che può essere anche derivata dalla Proposizione 1, è la seguente:

Proposizione 2. Ogni sottocammino di un cammino minimo è a sua volta minimo.

Prova. Si consideri un cammino minimo $P(s, w)$ dalla sorgente s a un vertice w. Sia $s = v_0, v_1, \ldots, v_k = w$ la sequenza dei vertici attraversati da $P(s, w)$. Per un generico vertice v_i su tale cammino, vogliamo mostrare che il cammino Q che attraversa i vertici $s = v_0, v_1, \ldots, v_i$ è un cammino minimo per v_i, nel senso che la sua lunghezza $\sum_{j=1}^{i} weight(v_{j-1}, v_j) = d(s, v_i)$, la distanza minima di v_i da s. Supponiamo per assurdo che Q non sia un cammino minimo da s a v_i. Avremo allora $\sum_{j=1}^{i} weight(v_{j-1}, v_j) > d(s, v_i)$. Sia $\hat{Q}$ un cammino minimo per v_i. Possiamo allora costruire un nuovo cammino $\hat{P}(s, w)$ per w nel modo seguente: i) la prima parte di $\hat{P}(s, w)$ coincide con $\hat{Q}$; ii) la secondo parte è costituita dagli archi $(v_i, v_{i+1}), \ldots, (v_{k-1}, w)$ di $P(s, w)$. Tale cammino ha lunghezza inferiore a quella di $P(s, w)$, contro l'ipotesi che quest'ultimo fosse un cammino minimo per w.

Algoritmo di Dijkstra

Di seguito dimostriamo la correttezza dell'algoritmo di Dijkstra offrendo una prova alternativa che usa le condizioni di ottimalità. Tale prova è del tutto equivalente a quella presentata nel libro e viene riportata qui per completezza, in modo che gli studenti possano ripercorrere il funzionamento dell'algoritmo di Dijkstra da una diversa angolazione.

Proposizione 3. Sia $G = (V, E)$ un grafo pesato (diretto o meno) privo di archi aventi peso negativo. L'algoritmo di Dijkstra applicato a partire dalla sorgente $s \in V$ assegna a ogni vertice $v \in V$ la distanza minima da s.

Prova. La prova consiste nel mostrare che, per ogni $w \in V$, la distanza $D[w]$ calcolata dall'algoritmo per il nodo w soddisfa le condizioni di ottimalità (1): per ogni arco $e = (v, w)$ diretto da v a w abbiamo:

$$D[w] \leq D[v] + weight(v, w) \qquad (2)$$

Consideriamo perciò il generico nodo w e il generico arco (v, w) incidente in w. Se w è estratto dalla coda di priorità prima di v la disuguaglianza sopra vale nel momento in cui v stesso è estratto dalla coda di priorità, in quanto in quel momento dobbiamo per forza avere $D[w] \leq D[v]$ e stiamo assumendo $weight(v, w) \geq 0$. Altrimenti, l'arco (v, w) viene rilassato esattamente una volta, ossia quando v è estratto dalla coda di priorità. In tale occasione viene eseguito il seguente frammento di pseudo-codice:

```
if (D[w] > D[v] + e.weight)
    D[w] = D[v] + e.weight;
```

Dopo tale rilassamento, abbiamo ovviamente $D[w] \leq D[v] + weight(v, w)$. Si noti che in tal caso w è estratto dalla coda di prorità dopo v e il valore $D[w]$ potrebbe decrescere ulteriormente a seguito di altri rilassamenti (ad esempio, esiste un arco diretto (z, w) e a un certo punto estraiamo z dalla coda di priorità, prima di w). D'altra parte, il valore di $D[v]$ non è più aggiornato una volta che v sia stato estratto dalla coda di priorità (il rilassamento si applica ai soli vertici non ancora estratti dalla coda di priorità), quindi la condizione (2) è mantenuta fino al termine dell'algoritmo. Poiché tale condizione è stata dimostrata per il generico arco (v, w), la

Proposizione 1 implica immediatamente l'ottimalità dei percorsi calcolati dall'algoritmo di Dijkstra.