AlgoReti (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 13 Febbraio 2008
Soluzioni

Esercizio 1. Si diano una delimitazione per il numero di push saturanti e una delimitazione per il numero di push non saturanti eseguite dall'algoritmo di preflow-push generico (con relative dimostrazioni).

Soluzione. Vedere il libro di testo (*Network flows*, Ahuja, Magnanti, Orlin): in particolare, le dimostrazioni dei lemmi 7.8, 7.12 e 7.15.

◁ ◇ ▷

Esercizio 2. Sia G un grafo orientato con pesi positivi sugli archi. Siano s ed (u, v) rispettivamente un nodo e un arco di G. Proporre due algoritmi per:

1. determinare se esiste un cammino minimo da s a v che *contiene* l'arco (u, v);
2. determinare se esiste un cammino minimo da s a v che *non contiene* l'arco (u, v).

Discutere formalmente correttezza e tempo di esecuzione degli algoritmi proposti, che dovrebbero essere il più possibile efficienti.

Soluzione. Il fatto che siano dati una sorgente s e che i pesi siano positivi suggerisce di cercare di utilizzare l'algoritmo di Dijkstra: non serve ricorrere a Bellmann-Ford (che ha senso applicare se ci sono pesi negativi) né a Floyd-Warshall (che risolve il problema all-pairs, ovvero tra tutte le coppie di nodi). Osserviamo inoltre che la risposta può essere:

- `falso` per entrambi i problemi: u e v potrebbero non essere raggiungibili da s;
- `vero` per entrambi i problemi: basta considerare il grafo orientato con archi (s, v) di peso 2, (s, u) di peso 1 ed (u, v) di peso 1;
- `falso` per il primo probleme e `vero` per il secondo:basta considerare il grafo orientato con archi (s, v) di peso 1, (s, u) di peso 1 ed (u, v) di peso 1;
- `vero` per il primo probleme e `falso` per il secondo:basta considerare il grafo orientato con archi (s, v) di peso 3, (s, u) di peso 1 ed (u, v) di peso 1.

Dato che non c'è un nesso evidente tra le risposte ai due problemi, risulta più semplice proporre due algoritmi differenti.

1. Per determinare se esiste un cammino minimo da s a v che *contiene* l'arco (u, v), basta eseguire l'algoritmo di Dijkstra con sorgente s per calcolare le distanze d da s a ogni altro nodo: la risposta è `vero` se $d_{sv} = d_{su} + w(u, v)$, `falso` altrimenti. La correttezza deriva dal fatto che l'algoritmo di Dijkstra calcola correttamente le distanze e dalla proprietà di appartenenza di un arco ad un cammino minimo (Lemma 13.5 del libro di testo *Algoritmi e strutture dati*, Demetrescu, Finocchi, Italiano).
2. Un possibile approccio (non l'unico) per determinare se esiste un cammino minimo da s a v che *non contiene* l'arco (u, v) è il seguente:
   - eseguire l'algoritmo di Dijkstra con sorgente s per calcolare le distanze d da s a ogni altro nodo;
   - rimuovere dal grafo l'arco (u, v);

- rieseguire l'algoritmo di Dijkstra con sorgente s per calcolare le distanze d' da s a ogni altro nodo nel nuovo grafo;
- restituire `vero` se $d_{sv} = d'_{sv}$, `falso` altrimenti.

La correttezza deriva dal fatto che per calcolare d' l'arco (u, v) non è preso in considerazione. Quindi, se $d_{sv} = d'_{sv}$, vuol dire che esiste un cammino da s a v che non usa (u, v) e ha costo minimo pari a d_{sv}. Altrimenti (in tal caso non può che essere $d_{sv} < d'_{sv}$), l'arco (u, v) era indispensabile per raggiungere v con costo minimo, e quindi il cammino cercato non esiste.

Per entrambi gli algoritmi, il tempo di esecuzione è dominato da quello dell'algoritmo di Dijkstra, ovvero $O(m + n \log n)$ assumendo di usare heap di Fibonacci.

◁ ◇ ▷

Esercizio 3. Si consideri il problema di identificare, tra tutti i minimi tagli di un grafo, un minimo taglio con il minimo numero di archi. Dimostrare che, rimpiazzando le capacità u_{ij} con nuove capacità $u'_{ij} = m \cdot u_{ij} + 1$ (m denota il numero di archi del grafo), il minimo taglio calcolato rispetto alle nuove capacità u'_{ij} è un minimo taglio con il minimo numero di archi rispetto alle capacità originarie u_{ij}.

Soluzione. Dimostriamo innanzitutto una utile relazione tra le vecchie e le nuove capacità. Sia $T = [S, \overline{S}]$ un taglio $s - t$ nel grafo e sia $|T|$ il numero di archi in T. Siano inoltre $u(T)$ ed $u'(T)$ rispettivamente le capacità del taglio T prima e dopo la ripesatura. Risulta:

$$u'(T) = \sum_{(i,j)\in(S,\overline{S})} u'_{ij} = \sum_{(i,j)\in(S,\overline{S})} (m \cdot u_{ij} + 1) = m \cdot u(T) + |T| \tag{1}$$

Consideriamo ora un taglio C con capacità $u(C)$ minima e contenente il minimo numero di archi. Per risolvere l'esercizio basta mostrare due proprietà:

1. un taglio D tale che $u(D) > u(C)$ non può essere un taglio minimo rispetto a u', ovvero
$$u(D) > u(C) \quad \Rightarrow \quad u'(D) > u'(C)$$
2. se D è un taglio minimo rispetto a u ma non contiene il minimo numero di archi (ovvero $|D| > |C|$), allora D non può essere un taglio minimo rispetto a u'; più formalmente
$$u(D) = u(C) \;\; e \;\; |D| > |C| \quad \Rightarrow \quad u'(D) > u'(C)$$

La 1 mostra che gli unici tagli candidati ad essere tagli minimi rispetto a u' sono quelli che erano già minimi rispetto a u. La 2 mostra che, tra tutti i tagli minimi rispetto ad u, solo quelli con il minimo numero di archi sono minimi anche rispetto a u'. Iniziamo col dimostrare il punto 2, che risulta più semplice.

Dimostrazione punto 2. Usando le ipotesi $u(D) = u(C)$, $|D| > |C|$, e l'Equazione 1 sappiamo che

$$u'(D) = m \cdot u(D) + |D| > m \cdot u(C) + |C| = u'(C)$$

e quindi $u'(D) > u'(C)$, come volevamo.

Dimostrazione punto 1. Usando l'ipotesi $u(D) > u(C)$ e l'Equazione 1 sappiamo che

$$u'(D) = m \cdot u(D) + |D| > m \cdot u(C) + |D|$$

Se $|D| \geq |C|$, abbiamo $m \cdot u(C) + |D| \geq m \cdot u(C) + |C| = u'(C)$, che conclude la dimostrazione. D potrebbe però essere un taglio con capacità grande, ma con pochi archi: potrebbe quindi risultare $|D| < |C|$. Consideriamo ora questo secondo caso. Osserviamo innanzitutto che $u(D) > u(C)$ equivale a dire $u(D) \geq u(C) + 1$ essendo le capacità intere. Quindi possiamo scrivere:

$$u'(D) = m \cdot u(D) + |D| \geq m \cdot (u(C) + 1) + |D| > m \cdot u(C) + m \geq m \cdot u(C) + |C| = u'(C)$$

Nella precedente catena di disuguaglianze abbiamo usato i fatti, banali, che $|D| > 0$ e $|C| \leq m$. In entrambi i casi abbiamo quindi ottenuto $u'(D) > u'(C)$, come volevamo.