Algoritmo randomizzato per taglio di capacità minima

Sia

$$C^* = \text{capacità del minimo taglio}, \qquad C_\Sigma = \sum_{e \in E} c_e, \quad C_i = \sum_{e \in \delta(i)} c_e$$

Algoritmo: un arco viene scelto con probabilità c_e/C_Σ, sia $\hat{e} = (u, v)$, i nodi u e v vengono fusi, $\hat{e}$ viene rimosso e gli archi eventualmente incidenti in u e in v vengono fusi (con somma delle capacità). Se $\hat{e}$ non appartiene al taglio minimo, il nuovo grafo ha il medesimo taglio minimo con lo stesso valore (infatti ogni taglio del grafo originario che non contiene $\hat{e}$ è presente anche nel grafo contratto, mentre ogni taglio del grafo originario che contiene $\hat{e}$ non esiste nel grafo contratto, inoltre i tagli corrispondenti hanno le medesime capacità di taglio). La procedura continua finché rimangono solo due nodi che individuano il taglio finale.
Analisi:

$$C_i \geq C^*$$

sommando su tutti i nodi

$$\sum_i C_i \geq n\, C^*$$

siccome

$$\sum_i C_i = 2\, C_\Sigma$$

$$\frac{C^*}{C_\Sigma} \leq \frac{2}{n}$$

quindi la probabilità di non scegliere un arco del taglio minimo è almeno

$$1 - \frac{2}{n}$$

Nel grafo successivo con $(n-1)$ nodi la probabilità diventa

$$1 - \frac{2}{n-1}$$

L'ultimo grafo ha 3 nodi e quindi la probabilità è

$$1 - \frac{2}{3}$$

Quindi la probabilità di non scegliere mai un arco del taglio minimo (e quindi fornire come soluzione proprio il taglio minimo) è almeno

$$(1 - \frac{2}{n})\,(1 - \frac{2}{n-1})\,(1 - \frac{2}{n-2})\,\dots\,(1 - \frac{2}{5})\,(1 - \frac{2}{4})\,(1 - \frac{2}{3}) = \frac{2}{n\,(n-1)} \geq \frac{2}{n^2}$$

e la probabilità di non trovare il taglio minimo è al più

$$1 - \frac{2}{n^2}$$

dopo k tentativi la probabilità di non trovare mai il taglio minimo è al più

$$(1 - \frac{2}{n^2})^k \approx e^{-2k/n^2}$$

quindi

$$e^{-2k/n^2} \leq \varepsilon \quad \Longrightarrow \quad \frac{-2\,k}{n^2} \leq \ln \varepsilon \quad \Longrightarrow \quad k \geq \frac{n^2\, \ln \varepsilon^{-1}}{2}$$

Complessità: ogni fusione realizzata con struttura Union-Find ha un costo $O(\log n)$. La scelta casuale di un arco richiede una ricerca binaria fra gli archi ancora da scegliere ed ha complessità $O(\log m) = O(\log n)$. Anche l'aggiornamento della struttura dati per la scelta casuale ha complessità $O(\log m)$. Il numero di iterazioni per la ricerca di un taglio è al più pari a $m-1$ (ad esempio due cliques connesse con un solo arco potrebbero richiedere la scelta di tutti gli archi delle due cliques). Quindi un taglio si trova con complessità $O(m\, \log n)$. Quindi per avere il taglio minimo con probabilità $1 - \varepsilon$, la complessità è $O(m\, n^2\, \log n\, \log \varepsilon^{-1})$.