Presentazioni dell'algoritmo di ricerca del flusso massimo monodirezionale con ottimizzazione agli ultimi livelli

Filippo Magi

February 16, 2022

1 strutture dati utilizzate

1.1 BiEdge

I nodi vengono collegati tra di loro da archi BiEdge, che contiene le informazioni da quale nodo esce e in quale nodo entra, in maniera tale da non dover aver bisogno di modificare, durante l'invio del flusso, due archi ma solo uno. Ovviamente conserva in memoria la quantità di flusso che passa e la sua capacità residua, oltre a un booleano (reversed) che mi indica che quel arco, durante l'invio del flusso, dovrà inviarlo o ritirarlo.

1.2 Node

Node contiene al suo interno informazioni sugli archi a lui collegati (Lista di BiEdge), la label, cioè la distanza tra lui e il nodo sorgente s, un booleano che indica se il nodo è valido o meno,le indicazioni sul percorso che deve fare, quindi sia previousNode e previousEdge, oltre ad avere quanto flusso gli arriva dal percorso indicato, indicato come InFlow.

1.3 Graph

Graph è rappresentato da un Lista di vari insiemi, ogni insieme contiene i nodi che hanno una certa label, oltre ad avere un insieme di nodi non validi.

2 descrizione

Dopo aver trovato un cammino tra il nodo sorgente s e il nodo destinazione t, mentre invio il flusso, controllo che la capacità residua degli archi sia maggiore di zero, in caso la capacità sia pari a 0, il nodo successivo viene marcato salvato in una pila qui chiamata **malati**. Poi effettuo la ricerca del cammino

2.1 Ricerca del cammino

Controllo se la pila **malati** è vuota, in cancello tutte le informazioni di indirizzamento (previousNode,previousEdge e InFlow) e accodo alla coda **coda** il nodo sorgente s. Se la pila non è vuota, provo a riparare i nodi di **malati**, da qui abbiamo due possibilità:

- tutti i nodi di **malati** vengono riparati, così facendo ho già trovato un percorso da s a t, usando informazioni conosciute precedentemente.
- trovo un nodo $n \in$ **malati** che non è possibile riparare, in tal caso cancello tutte le informazioni di indirizzamento dei nodi con label pari o maggiore a quella di n, e inserisco in **coda** tutti i nodi $m \in V(G) | m.label = n.label - 1$

Nel caso in cui non abbia già trovato il cammino, procedo con la ricerca di t partendo dai nodi presenti in **coda**.

2.2 RepairNode

È possibile "guarire il nodo" **malato**, cioè cercare un nodo **sostituto** a lui collegato con le seguenti proprietà

1. l'arco tra **malato** e **sostituto** deve avere capacità (o flusso,nel caso **malato** sia nodo uscente dell'arco) positivo, cioè maggiore di zero
2. la label di **sostituto** deve essere uguale alla label di **malato** -1, cioè **sostituto**.Label = **malato**.Label-1
3. **sostituto** deve essere un nodo valido
4. il valore del flusso passante per **sostituto** deve essere positivo

se **sostituto** rispetta queste qualità, aggiorno **malato** e confermo di essere riuscito a ripararlo.