

Report progetto Ingegneria degli Algoritmi

Alessio Susco mat. 266383

30 aprile 2021

1 Introduzione

Al giorno d'oggi, moltissime sono le applicazioni che fanno uso intensivo di algoritmi per il calcolo di cammini minimi su grafi. Il problema principale dei grafi nelle applicazioni moderne è che cambiano la loro struttura continuamente ed in modo non predicibile a priori. Gli eventi che vanno a cambiare la struttura del grafo possono essere: aggiunta o rimozione di nodi e/o archi e incrementi/decrementi dei pesi degli archi. L'obiettivo di questo studio è quello di andare a valutare le performance di due algoritmi per il calcolo dei cammini minimi di un grafo a fronte di eventi di aggiunta di archi ed incremento dei pesi degli archi. I due algoritmi presi in esame sono quelli messi a disposizione dal toolkit Networkit nella libreria distance, in particolare: distance.Dijkstra e distance.DynDijkstra.

1.1 DynDijkstra

L'algoritmo DynDijkstra, dato un grafo e un evento, permette di calcolare i cammini minimi da un nodo sorgente andando a ricalcolare solo i cammini che sono stati influenzati dall'evento.

1.2 Dijkstra

L'algoritmo Dijkstra, dato un grafo, permette di calcolare i cammini minimi da un nodo sorgente. Al contrario del precedente appena descritto è intuibile che, a fronte di tali eventi, è necessario rieseguirlo per avere i cammini minimi corretti ed aggiornati. Pertanto, tale algoritmo è stato utilizzato per quantificare quanto fosse più veloce l'algoritmo dinamico in particolari set di grafi.

2 Progettazione degli esperimenti

Con questo studio si vuole determinare quali fattori influenzano maggiormente lo speedup tra l'algoritmo statico e quello dinamico, pertanto per la scelta dei fattori e dei design points è stato fatto riferimento alle linee guida relative alle categorie di domande di tipo Assessment. Di seguito verranno brevemente riportate le metriche ed i parametri scelti:

- Performance Metrics: Speedup e Time

- Performance Indicator: CPU Time speso dai due algoritmi, scelta obbligata per ridurre al minimo l'influenza degli altri processi sui risultati durante l'elaborazione
- Algorithm Parameters: istanze di grafi e nodo sorgente (Dijkstra(graph, source)). Inoltre, anche gli eventi di modifica delle istanze dei grafi sono stati considerati come parametri algoritmici
- Instance Parameters:
 - Input Source: I grafi sono stati generati randomicamente da due classi di generatori, cioè ErdosRenyi e BarabasiAlbert
 - Input Size: n : numero dei nodi, p : probabilità dell'esistenza di un arco per i grafi ER e k : numero di archi uscenti da un nodo per i grafi BA
- Factors: I parametri che vengono esplicitamente manipolati nei seguenti esperimenti sono il numero dei nodi nei grafi. In particolare, i livelli sono stati scelti seguendo la logica del Doubling Experiment partendo da un valore iniziale abbastanza grande che permettesse di evitare risultati spuri dovuti al floor effect. Il numero di raddoppi è stato scelto come compromesso tra generalità dei risultati, somiglianza a taglie di grafi dinamici reali e tempi di sperimentazione accettabili.
- Fixed Parameters: Il numero di eventi di modifica del grafo durante gli esperimenti è fissato a priori. Inoltre, anche p e k (parametri di istanza) sono stati fissati. In particolare $p = 0.02$ per i grafi Erdos-Renyi è stato scelto in modo da soddisfare le ipotesi di un teorema che ci garantisce che la probabilità del grafo generato $P(\{G(n, p) \text{ isconnected}\}) \rightarrow 1$
- Noise Parameters: Il tipo di evento dinamico di modifica del grafo è scelto casualmente ma in modo semicontrollato dato che i possibili eventi sono due, ovvero l'aggiunta di un arco al grafo e l'incremento del peso di un arco già presente nel grafo. Anche i pesi degli archi vengono scelti randomicamente tra due valori fissati a priori. Per diminuire il bias statistico dovuto al nodo sorgente, è stato scelto di cambiare, ad ogni evento, il nodo sorgente per il calcolo dei cammini minimi..
- Design Points: Date 6 taglie di input per ogni grafo

$$n = (500, 1000, 2000, 4000, 8000, 16000, 32000)$$

e due tipologie di grafi

$$GraphType = (BarabasiAlbert, ErdosRenyi)$$

abbiamo ottenuto 12 design points

3 Test Environment

3.1 Implementazioni

3.2 Istanze di input

3.3 Script

3.4 Programmi e tools

4 Risultati

5 Conclusioni