MST e COMPONENTI CONNESSE di un grafo

Luca Bindini

1 Introduzione

In questa relazione si andranno ad analizzare due algoritmi sui grafi non orientati: KRUSCAL e COMPONENTI CONNESSE. Entrambi gli algoritmi necessitano di una struttura dati per insiemi disgiunti (UNION-FIND) e dei metodi offerti da essa (MAKE-SET, FIND-SET e UNION). Si andranno a testare i suddetti algoritmi su grafi non orientati di diversa dimensione e con probabilità di avere un arco tra i vari nodi crescente.

2 Componenti connesse

Per componente connessa di un grafo non orientato si intende una classe di equivalenza tra i nodi secondo la relazione "è raggiungibile da" cioè se vi è un arco che li connette. L'algoritmo per trovare le componenti connesse si basa sullo scorrere tutti gli archi del grafo e verificare se i nuovi nodi "scoperti" appartengono o meno alla componente connessa trovata. Possiamo verificare come all'aumentare della probabilità di avere un arco tra due nodi diminuisca il numero di componenti connesse.

2.1 Test su algoritmo Componenti connesse

Ai fini di analizzare l'andamento del numero di componenti connesse su un grafo casuale al variare della probabilità di avere un arco tra i vari nodi, sono state eseguite 10 prove (con dimensione del grafo fissa) per ogni probabilità (espressa in percentuale) e fatta una media tra queste prove. Successivamente i test sono stati ripetuti con una diversa dimensione del grafo (10,100,500).

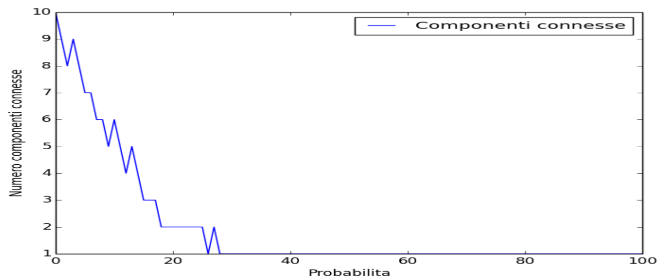

Figura 1: Andamento componenti connesse in un grafo di dimensione 10.

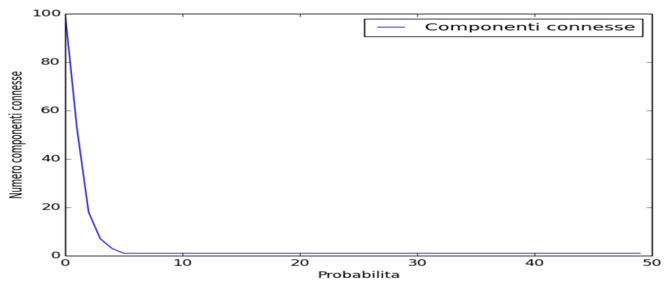

Figura 2: Andamento componenti connesse in un grafo di dimensione 100.

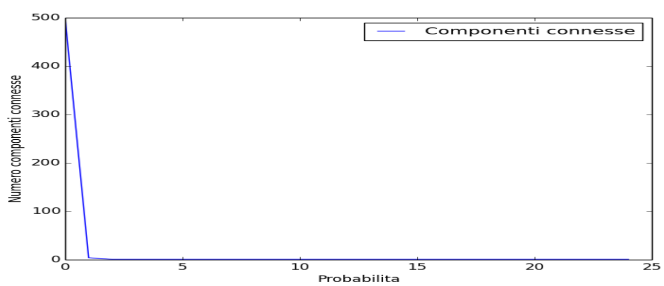

Figura 3: Andamento componenti connesse in un grafo di dimensione 500.

Come si può notare sperimentalmente dai grafici all'aumentare della probabilità di avere degli archi diminuisce il numero di componenti connesse del grafo. Inoltre l'aumento della dimensione fa si che il numero delle componenti connesse diminuisca più rapidamente con l'aumento della probabilità di avere un arco tra un nodo e un qualsiasi altro nodo.

3 MST (Alberi di connessione minimi)

Un MST, albero di connessione minimo, è un sottoinsieme aciclico di un grafo pesato $G(V, E)$ non orientato che connette tutti i nodi di tale grafo. Un algoritmo per trovare l'MST è l'algoritmo di Kruscal.

3.1 Algoritmo di Kruscal

L'algoritmo di Kruscal trova un MST a partire da un insieme vuoto A che è sottoinsieme di un qualche MST e via via aggiunge un arco sicuro (u, v), cioè un arco tale che l'insieme $A \bigcup (u, v)$ sia comunque sottoinsieme di un qualche MST. Quando l'algoritmo termina l'insieme A conterrà gli archi che compongono l' MST. Per l'implementazione di tale algoritmo si usa la struttura dati Union-Find. Il tempo di esecuzione di Kruscal dipende sia dal numero di nodi che dal numero degli archi: $T(n) = O(ElgV)$.

3.1.1 Test su algoritmo di Kruscal

Per analizzare l'algoritmo di Kruscal al variare della probabilità e delle dimensioni del grafo sono state eseguite 10 prove (con dimensione del grafo fissa) per ogni probabilità (espressa in percentuale) e fatta una media tra queste prove. Successivamente i test sono stati ripetuti con una diversa dimensione del grafo (10,100,500).

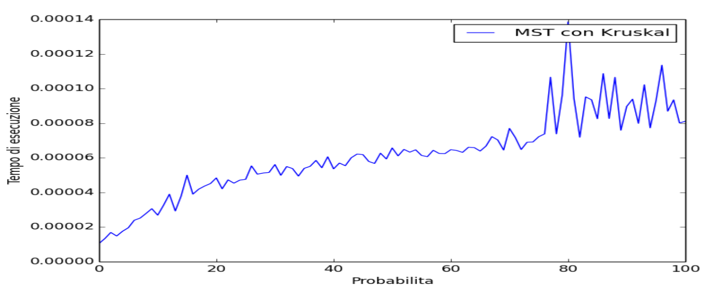

Figura 4: Tempo di esecuzione di Kruscal al variare della probabilità in un grafo di dimensione 10.

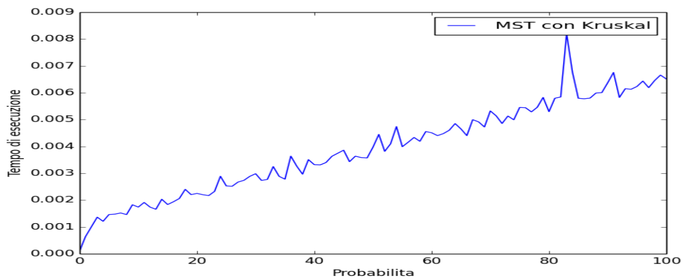

Figura 5: Tempo di esecuzione di Kruscal al variare della probabilità in un grafo di dimensione 100.

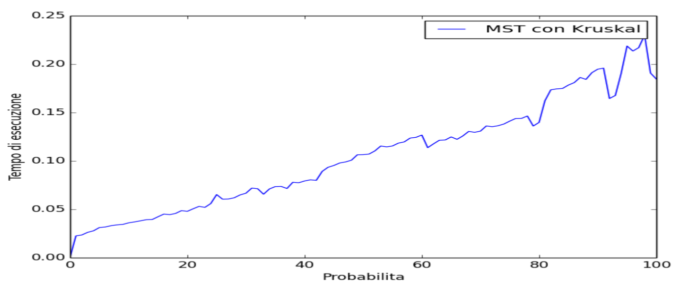

Figura 6: Tempo di esecuzione di Kruscal al variare della probabilità in un grafo di dimensione 500.

Probabilità	1	20	40	60	80	100
Prova 1	0.02120	0.04873	0.07819	0.12940	0.13479	0.21617
Prova 2	0.02120	0.05142	0.07561	0.12957	0.13432	0.26541
Prova 3	0.02111	0.04747	0.07560	0.10725	0.13768	0.23658
Prova 4	0.02137	0.04762	0.08647	0.13679	0.13342	0.22314
Prova 5	0.02626	0.04810	0.07525	0.12332	0.13647	0.23025
Prova 6	0.02173	0.04810	0.07525	0.12323	0.13457	0.23025
Prova 7	0.02471	0.04760	0.08167	0.12838	0.16569	0.23134
Prova 8	0.02163	0.04783	0.07549	0.16101	0.141741	0.23514
Prova 9	0.02183	0.04637	0.07544	0.10407	0.14377	0.21764
Prova 10	0.02484	0.04714	0.08658	0.12487	0.13592	0.22845

Tabella 1: Tempi di esecuzione di Kruscal in un grafo di dimensione 500 al variare della probabilità di avere archi.

Come possiamo notare all'aumentare della probabilità di avere degli archi il tempo di esecuzione aumenta.

4 Conclusioni

Analizzando i dati raccolti si può vedere come entrambi gli algoritmi siano influenzati dall'aumento della dimensione e dall'aumento della probabilità di avere archi. In particolare aumentando gli archi il numero di componenti connesse diminuisce mentre il tempo di esecuzione per trovare l'MST aumenta. Per entrambi gli algoritmi l'aumento della dimensione del grafo comporta un aumento del tempo di esecuzione.

[1]

[1] I test sono stati effettuati su una piattaforma con le seguenti specifiche:

1. Processore: 2,9 GHz Intel Core i5
2. Ram: 8 GB 2133 MHz LPDDR3
3. SO: Mac OS 10.14.5 Mojave