Componenti Fortemente Connesse

Indice

1 Introduzione

La seguente relazione implementa l'algoritmo per la ricerca delle componenti fortemente connesse generando grafi in maniera casuale, testando il comportamento dell'algoritmo al crescere del numero di nodi all'interno del grafo e della probabilità di trovare archi tra due nodi.

2 Descrizione algoritmi

Data un grado diretto $G = (V, E)$ con V numero dei vertici ed E numero degli archi del grado, una componente fortemente connessa (SCC, Strongly Connected Component) è definita come l'insieme massimale di vertici U sottoinsieme di V tale che per ogni coppia di vertici u e v in U, u è raggiungibile da v e viceversa.

2.1 Strongly Connected Component

Alla base dell'algoritmo per il calcolo delle componenti fortemente connesse di un grafo vi sono: l'algoritmo per il calcolo del grafo trasposto e quello per la visita in profondità (DFS) del grado stesso.
L'algoritmo consiste nell'eseguire due volte il DFS: una prima volta sul grafo stesso e una seconda sul grafo trasposto, considerando i vertici in ordine decrescente rispetto ai tempi di fine ottenuti durante la prima visita.
In questo modo si riesce a identificare le componenti fortemente connesse del grafo in $\theta(V + E)$

2.2 Depth First Search

Dato un grafo, la visita in profondità, ispeziona gli archi di un grafo a partire dall'ultimo vertice che ha ancora archi non ispezionato uscenti da esso. Durante l'ispezione i vari vertici del grafo vengono contrassegnati con tre differenti colori per indicare il loro stato: inizialmente tutti i vertici sono bianchi; quando un nodo viene scoperto per la prima volta diventa gridio; infine quando viene completato, ovvero quando si è analizzato completamente la sua lista di adiacenza, diventa nero. Inoltre a ogni vertice viene assegnato un tempo di inizio e un tempo di fine che indicano il momento in cui si è scoperto il vertice e quello in cui si è terminata l'ispezione della sua lista di adiacenza.
Questo processo continua finché non si scoprono tutti i vertici raggiungibili dalla sorgente originale.
Nel caso in cui restassero vertici non scoperti, uno di questi verrebbe considerato come nuova sorgente.
Una visitò in profondità forma anche un sotto-grafo di predecessori detto

depth-forest, costituito da degli alberi DF che comprendono tutti i nodi raggiungibili da una radici comune.

3 Prestazioni attese

Sappiamo che fissato il numero di vertici e incrementando la probabilità della presenza di archi dallo 0 al 100% il numero delle componenti fortemente connesse, inizialmente pari al numero di vertici, tenderà a diventare uno. Pertanto si andrà ad avere un'unica componente fortemente connessa, che raggrupperà tutti i vertici del grafo.

4 Esperimenti

I grafi su cui si sono effettuati i test, sono stati rappresentati tramite matrici di adiacenza, pertanto al variare delle dimensioni, si è generato tali matrici tenendo conto dell'incremento della probabilità della presenza di archi. In queste prova sperimentale è stato analizzato l'andamento del numero di componenti fortemente connesse al crescere del numero di nodi e della probabilità di trovare archi tra i nodi. In particolare si sono fatte delle prove con 50, 100, 250 nodi.

4.1 Grafici

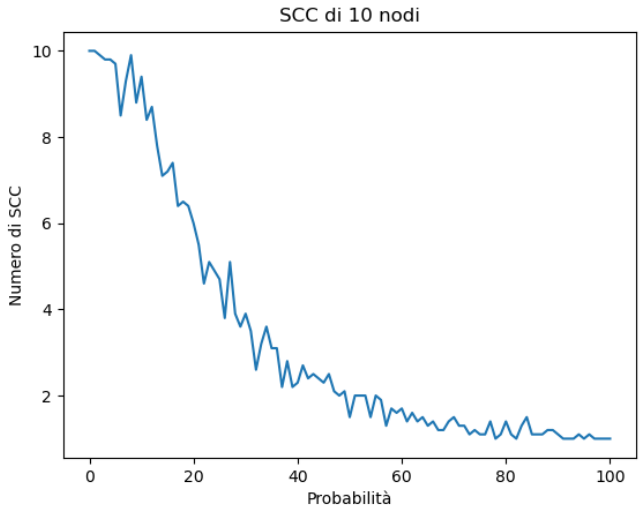

Figura 1: Numero di componenti fortemente connesse al variare della probabilità di un arco tra due nodi su un grafo di 10 nodi.

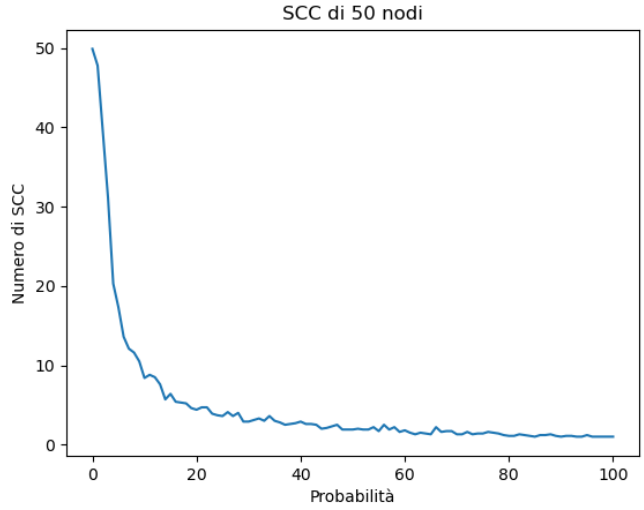

Figura 2: Numero di componenti fortemente connesse al variare della probabilità di un arco tra due nodi su un grafo di 50 nodi.

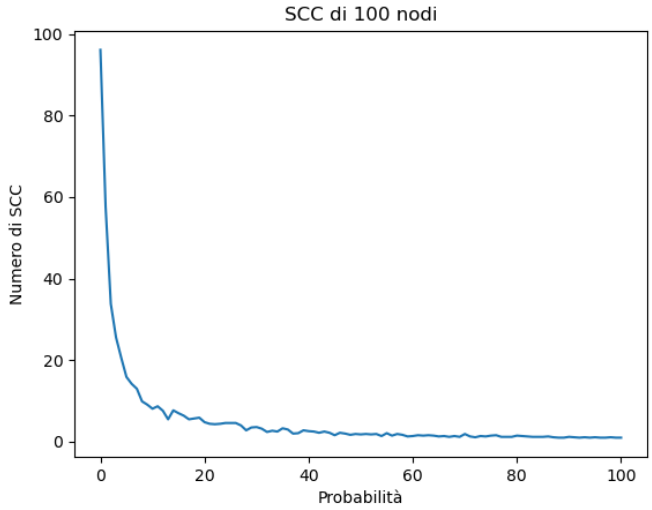

Figura 3: Numero di componenti fortemente connesse al variare della probabilità di un arco tra due nodi su un grafo di 100 nodi.

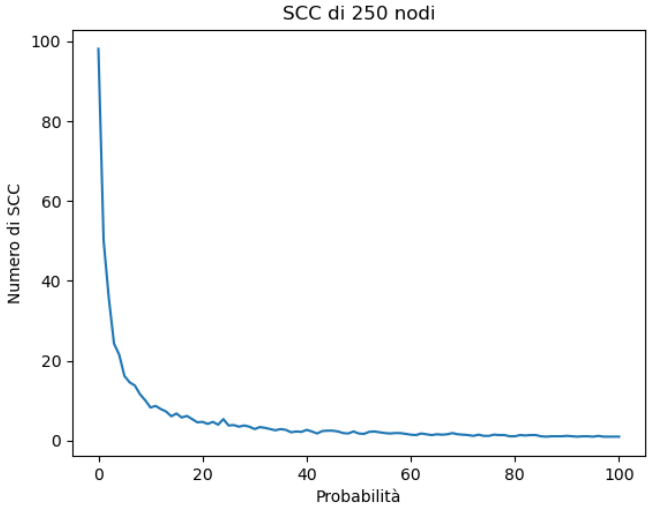

Figura 4: Numero di componenti fortemente connesse al variare della probabilità di un arco tra due nodi su un grafo di 250 nodi.

5 Documentazione del codice

L'elaborato scritto in linguaggio Python oltre ad importare le librerie per la generazione di numeri pseudo casuali e generazione di grafici, contiene due classi che sono Node e Graph che rappresentano rispettivamente un nodo e un grafo. Le due classi sono dotate di attributi e metodi tali da realizzare l'elaborato. Il grado è strutturato in maniera tale da poterne effettuare la visita in profondità che contrassegnino i differenti vertici con colori, tempi di scoperta e completamento in maniera opportuna.
Con l'obiettivo, di scrivere un algoritmo per il calcolo delle componenti fortemente connesse si è effettuato alcune modifiche all'algoritmo della visita in profondità. Tali modifiche hanno permesso di ottenere l'elenco dei vertici del grafo rispetto l'ordine decrescente dei tempi di completamento, di rendere corretta la visita di un grado trasposto secondo l'ordine decrescente e di tenere traccia dei vertici della SCC.

6 Conclusione

Gli esperimenti mostrano come l'incremento della probabilità della presenza di archi, abbia portato ad ottenere un numero decrescente di componenti fortemente connesse. Il risultato si spiega nel seguente modo: quando la probabilità è dello 0%, il grafo, indipendentemente dal numero di nodi che lo costituisce, è privo di archi, dunque non presenta alcuna SCC. Con una probabilità del 100% è costituito da $N(N-1)$ archi e dunque ogni nodo

del grafo risulta connesso con i restanti. Al crescere del numero di archi è quindi ragionevole pensare che il numero di SCC decresca all'incrementare della probabilità.

7 Informazioni Sul Calcolatore

7.1 Hardware

CPU: Intel Core i7-8565u @ 1.80Ghz
RAM: 8 GB DDR3
Graphic: Intel HD Graphics 620

7.2 Software

OS: Windows 10 Home
IDE: Pycharm Community Edition 2020.1.1