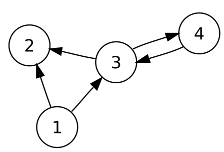
Componenti fortemente connesse

Salvatore Baglieri

Luglio 2020

1 Introduzione

Per prima cosa diamo al definzione di grafo.
Un grafo è un insieme di elementi detti nodi o vertici (V) che possono essere collegati fra loro da linee chiamate archi (E).
Ci sono due metodi standard per rappresentare un grafo: come una collezione di liste di adiacenza o come una matrice di adiacenza. Per i test utilizzerò la rappresentazione con matrice di adiacenza. La matrice è composta da 0 e 1, l'elemento 1 nella posizione $[i, j]$ corrisponde alla presenza di un arco che va da i a j con $i \neq j$, l'elemento 0 specifica la mancanza di un arco.
Di seguito viene riportato un piccolo esempio di grafo con la rispettiva matrice di adiacenza.



$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Diamo adesso le definizioni di componente fortemente connessa (SCC) e visita in profondità(DFS).

DFS: come suggerisce il nome, questo algoritmo visita il grafo sempre più in profondità ed è necessario ai fini dei test per trovare le componenti fortemente connesse. Di seguito viene descritto nel dettaglio l'algoritmo.

```
DFS(G):
  for ogni vertice u ∈ G.V do
    u.color ← WHITE
    u.π ← NIL
  end for
  time = 0
  for ogni vertice u ∈ G.V do
    if u.color == WHITE then
      DFS − VISIT(G, u)
```

```
        end if
    end for
```

```
DFS-VISIT(G,u):
    time = time + 1
    u.d = time
    u.color = GRAY
    for ogni V ∈ G.Adj[u] do
        if v.color == WHITE then
            v.π = u
            DFS − VISIT(G, u)
        end if
    end for
    u.color = BLACK
    time = time + 1
    u.f = time
```

SCC:questa permette di trovare le componenti fortemente connesse con l'ausilio di DFS e opera nel seguento modo:

SCC(G):

1. Chiama DFS(G) per calcolare i tempi di completamento $u.f$ per ciascun vertice u
2. Calcola G^T
3. Chiama DFS(G^T), ma nel ciclo principale di DFS, considera i vertici in ordine decrescente rispetto ai tempi $u.f$ (calcolari nella riga 1)
4. Genera l'output dei vertici di ciascun albero della foresta DF che è stata prodotta nella riga 3 come una singola componente fortemente connessa

2 Test

I test sono scritti in python. Ogni test è la media di 10 iterazioni dello stesso problema. La classe Graph genera di grafi casuali con un numero di nodi a scelta ed una determinata probabilità di presenza di archi tra vertici. Se la probabilità è del 100% allora tutti i vertici sono collegati, se invece la probabilità è dello 0% allora non esiste nessun arco.

2.1 Test caso base

In questo caso è stato eseguito l'SCC su due tipologie di grafi, uno con probabilità del 100% e l'altro con probabilità dello 0%. L'output del programma è il seguente:

- Numero di SCC trovate in un grafo da 50 nodi con probabilità di presenza degli archi pari a 0% = 50
- Numero di SCC trovate in un grafo da 50 nodi con probabilità di presenza degli archi pari a 100% = 1

2.2 Test incremento probabilità

Adesso analizziamo il caso in cui genero un grafo composto da 50 nodi e incremento di un unità la probabilità di presenza degli archi da 0% a 100%, infine per ogni grafo cerco le componenti fortemente connesse.

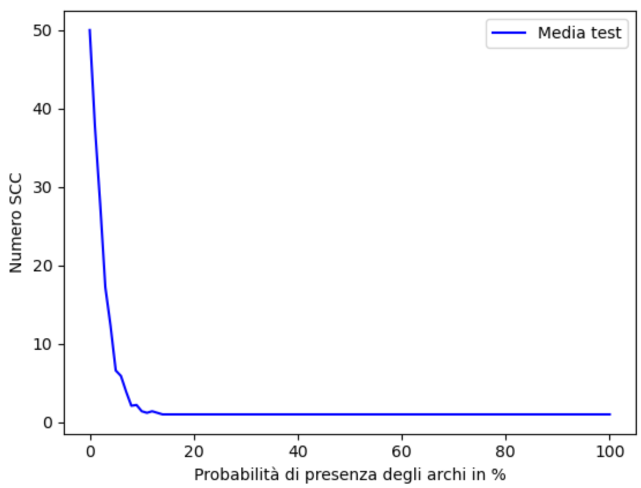


Figure 1: Numero di SCC su 50 nodi con incremento della probabilità

Notiamo dal grafico una diminuzione molto rapida del numero delle componenti fortemente connesse. Vediamo da quale probabilità in poi si hanno i nodi tutti connessi, quindi SCC = 1, facendo uno zoom del grafico sopra riportato.

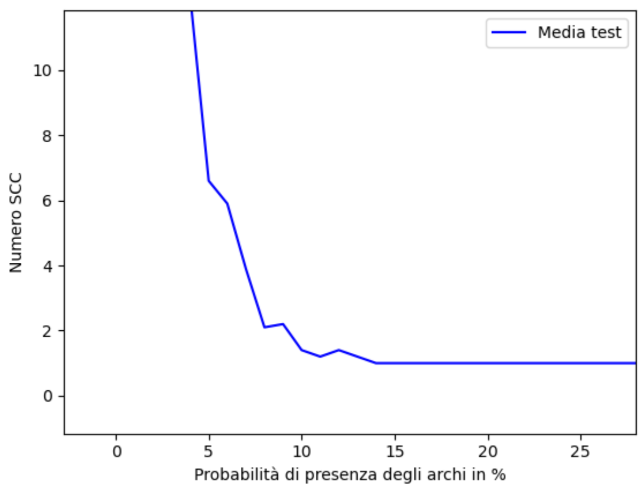


Figure 2: Numero di SCC su 50 nodi con incremento della probabilità

Vediamo che con una probabilità di presenza di archi $\geq 14\%$ abbiamo tutti i nodi connessi, quindi un'unica componente fortemente connessa.

2.3 Test incremento numero di nodi

Per quanto visto nel caso precedente sappiamo che se la probabilità è $\geq 14\%$ si hanno ottime possibilità di avere tutti i nodi connessi.
Con il segunete test genero dei grafi con la stessa probabilità fissata al 15% e numero di nodi crescente da 1 a 100.

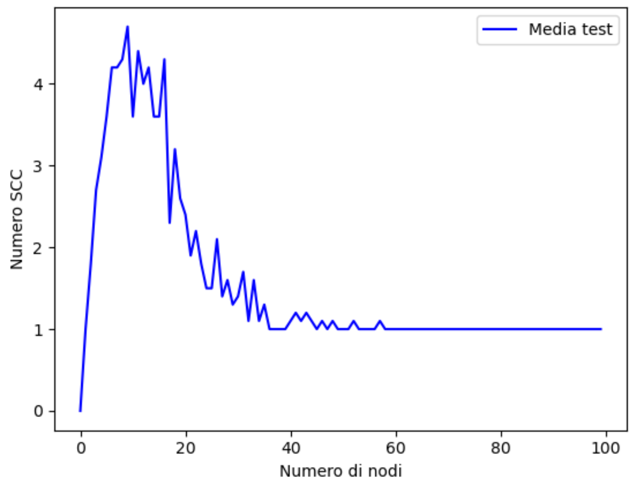


Figure 3: Numero di SCC con probabilità del 15%

Il grafico dimostra che quanto detto precedentemente è vero. Se ho quindi un grafo da 50 elementi e una probabilità $\geq 15\%$ è molto probabile che tutti i nodi siano connessi.
Notiamo però che se il numero di nodi non è abbastanza grande si hanno più SCC. Se vogliamo quindi avere una sola SCC devo assicurarmi di non avere un numero di nodi inferiore a 50, con questa probabilità.

3 Conclusione

Con il test del caso base abbiamo ottenuto quello che ci aspettavamo, se quindi la matrice è formata da soli 0 abbiamo tante SCC quanti sono gli elementi del grafo, questo perchè non esistono archi; se la matrice è formata da soli 1 ho una sola SCC. Con gli altri casi di test abbiamo ottenuto dei risultati poco intuitivi. Abbiamo scoperto quindi che per avere tutti i nodi fortemente connessi non ho bisogno di probabilità elevate, tranne nel caso in cui il numero di nodi è ridotto, e inoltre più nodi si hanno e meno SCC ci sono.