



UNIVERSITÀ  
DEGLI STUDI  
DI PALERMO

**Corso di Laurea in Informatica Magistrale (LM 18)**

*A.A. 2021/2022*

# SEIRD Epidemic Simulator on Network

*Giulia Maraventano*

# ABSTRACT

---

Questo progetto, realizzato per l'esame di Reti Complesse presso l'Università degli Studi di Palermo, è volto all'analisi grafica dell'evoluzione di un'epidemia virale, secondo il modello SEIR (Susceptible, exposed, infected, recovered), in una popolazione rappresentata da un network di nodi connessi.

Il network è basato sul modello *Poisson random graph*  $G(n,p)$  (*Erdős-Rényi*), ovvero un modello di network random, generato fissando il numero di nodi e la probabilità di connessione tra un nodo ed un altro. Il software implementato è un simulatore composto da un'interfaccia grafica dove inserire i parametri di input, e da un output grafico che mostra l'andamento degli stati epidemici nel tempo.

# Modello SEIRD

Il modello epidemiologico SEIRD prevede i seguenti stati:

- **Susceptible** (stato in cui il soggetto ancora sano non è entrato a contatto con virus)
- **Exposed** (stato in cui il soggetto è esposto alla malattia)
- **Infected** (stato in cui il soggetto contrae l'infezione)
- **Recovered** (stato in cui il soggetto è guarito dalla malattia)
- **Dead** (stato in cui il soggetto muore)

L'individuo suscettibile può essere contagiato, con una probabilità  $\alpha$  (tasso di trasmissione), da parte di un individuo infetto a lui connesso.

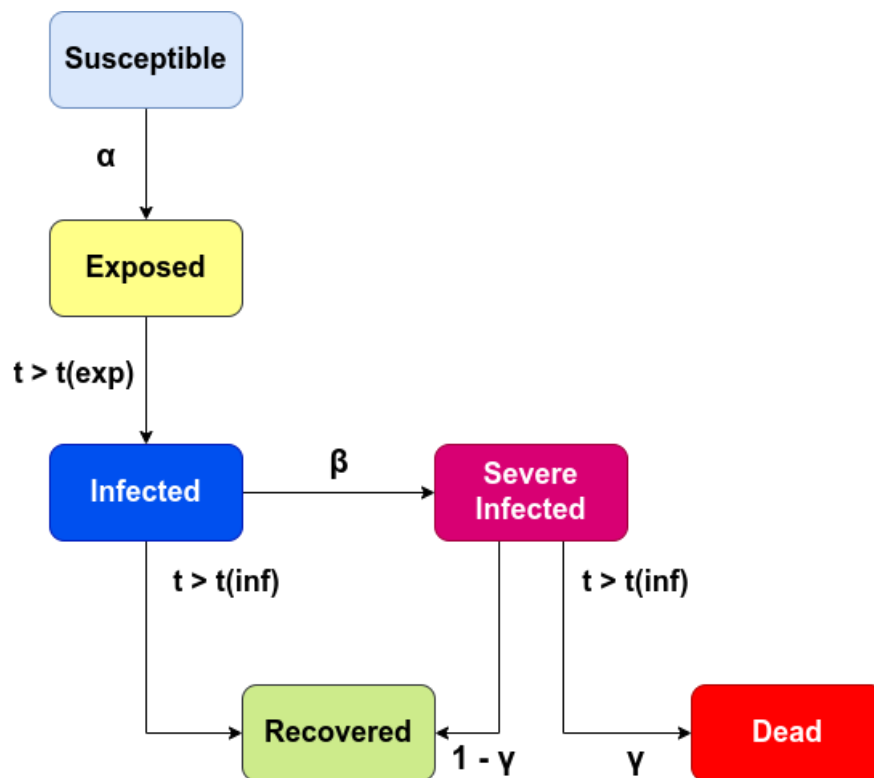
Lo stato *exposed* ha una durata di tempo  $t(exp)$ , dopo la quale avviene automaticamente il passaggio allo stato *infected*.

Il simulatore prevede un ulteriore stato intermedio, ovvero lo stato *severe infected*. La probabilità che un individuo passi dallo stato *infected* a *severe infected*, è indicata dal parametro  $\beta$ .

Si può fissare in input l'intervallo di tempo  $t(inf)$  (durata della malattia), dopo il quale il soggetto viene ritenuto *recovered*, cioè guarito.

Nello stato di *severe infected*, c'è la probabilità  $\gamma$  (tasso di mortalità) che un individuo passi allo stato *dead*, cioè che muoia.

Di seguito, viene illustrato il diagramma degli stati:



Il network scelto per simulare la popolazione è un grafo random, in cui la probabilità di connessione è uguale per tutti i nodi del network. Il numero totale di individui della popolazione (numero totale di nodi del network) non varia nel tempo. Quando un individuo entra nello stato *dead*, il nodo relativo non viene effettivamente rimosso, ma viene escluso dai successivi step iterativi, perchè trattandosi di uno stato finale, non può evolversi in altri stati successivi. In questo modello, anche lo stato *recovered* è considerato uno stato finale. Infatti non è contemplata la possibilità che un individuo guarito ritorni suscettibile.

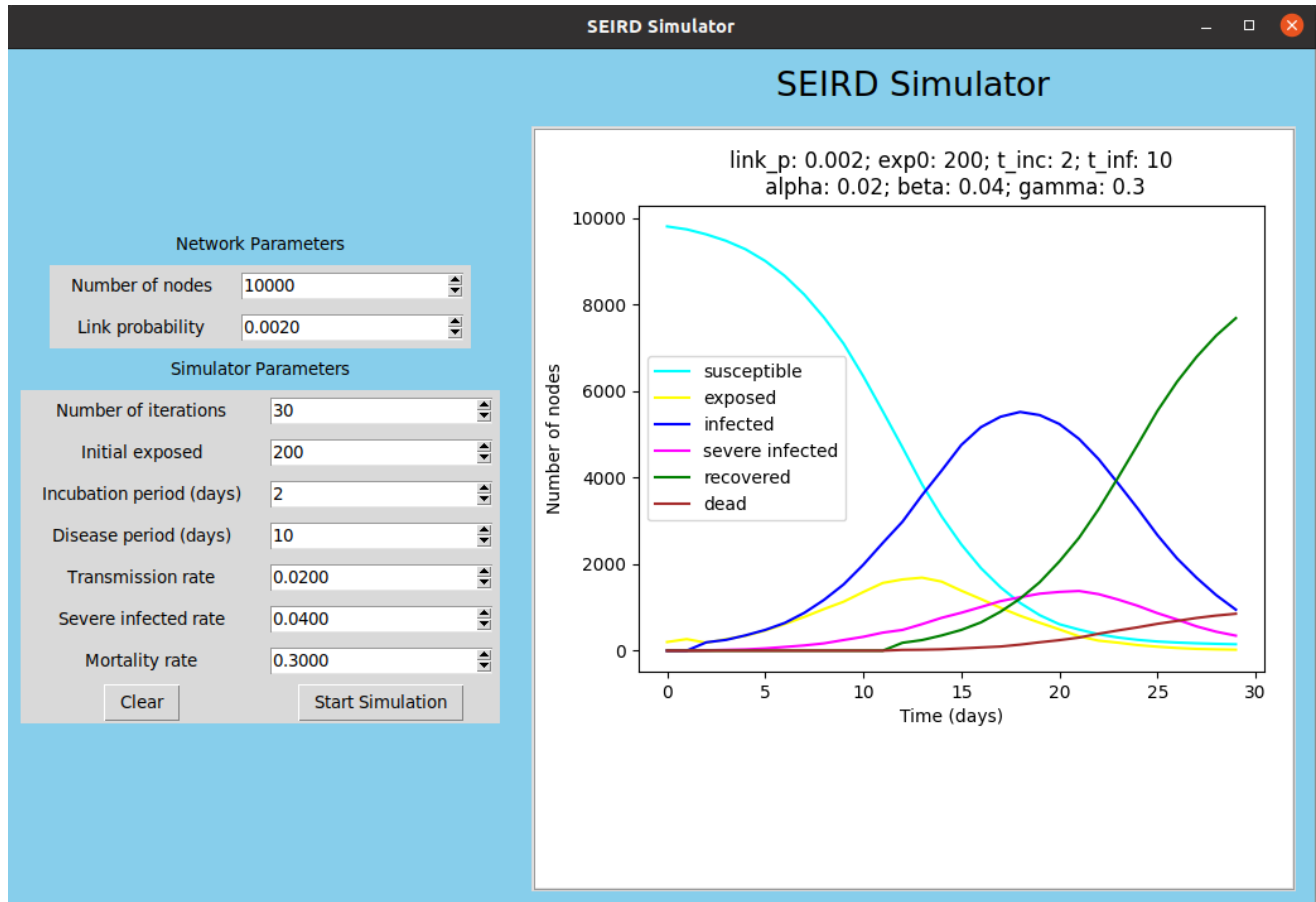
# Struttura del codice

Il simulatore è stato realizzato con il linguaggio Python. La struttura del programma è composta dai seguenti file di codice:

- model.py
- simulator.py
- main.py

Lo script principale è il main.py, che contiene l'interfaccia grafica e tutti i suoi oggetti.

Un esempio è descritto nella seguente figura:



I parametri a sinistra, sono impostati con un determinato valore di default, nelle relative variabili del main.py, e possono essere modificati da input. Con il pulsante *Start Simulation* viene chiamata la funzione che avvia la simulazione e che contiene gli oggetti di classe Network e Simulator, definiti rispettivamente nei file *model.py* e *simulator.py*. La funzione è composta dalle seguenti righe di codice:

```
def start_sim():
    sim = simulator.Simulator()
    net = model.Network(sim)

    for i in range(sim.num_iter):
        net.update_states(i,sim)
        net.t_state = net.t_state + sim.dt_state
        new_cases = net.get_new_cases(i,sim)
        sim.s_to_e(net,new_cases)
        sim.e_to_i(net)
        sim.i_to_r(net)
        sim.i_to_ig(net,i)
        sim.ig_to_d(net,i)
        if i>=plot_freq and i%plot_freq==0:
            p = net.plot(i,sim)
```

Dopo la creazione degli oggetti Simulator e Network, viene eseguito un ciclo iterativo con un numero di iterazioni (numero di giorni) stabilito da input nell'interfaccia. All'interno di questo ciclo, vengono eseguite le seguenti funzioni:

1. **update\_states()**: funzione che si occupa di aggiornare la tabella degli stati epidemici per ogni iterazione.
2. **get\_new\_cases()**: funzione che calcola e restituisce i nuovi casi di esposizione alla malattia, in base alla lista di adiacenza del network: i nuovi nodi (individui) contagiati, sono quelli non morti e collegati ad altri nodi infetti.
3. **s\_to\_e()**: funzione che prende in input i nuovi casi, calcolati nella funzione precedente, e aggiorna lo stato epidemico exposed.
4. **e\_to\_i()**: funzione che aggiorna lo stato epidemico da exposed ad infected, dopo che è trascorso il tempo di incubazione del virus.
5. **i\_to\_r()**: funzione che aggiorna lo stato epidemico da infected a recovered, dopo che sono trascorsi i giorni della malattia.
6. **i\_to\_ig()**: funzione che stabilisce quali casi di infected passano allo stato severe infected.
7. **ig\_to\_d()**: funzione che stabilisce quali casi di severe infected passano allo stato dead.
8. **plot()**: funzione che stampa il grafico dell'andamento degli stati epidemici nel tempo.

## Analisi dei risultati

I parametri che si possono impostare dall'interfaccia grafica sono i seguenti:

- **Number of nodes**: il numero di nodi (degli individui) del network.
- **Link probability**: probabilità di connessione tra un nodo ed un altro.
- **Number of iterations**: numero di iterazioni, che corrisponde al numero di giorni analizzati.
- **Initial exposed**: numero iniziale di individui esposti alla malattia.
- **Incubation period**: periodo di incubazione del virus (in giorni).
- **Disease period**: periodo della malattia (in giorni).
- **Transmission rate**: tasso di trasmissione del virus ( $\alpha$ ).
- **Severe infected rate**: tasso di infezione grave ( $\beta$ ).
- **Mortality rate**: tasso di mortalità ( $\gamma$ ).

Impostando il numero di nodi del network e la probabilità di connessione tra due nodi, viene generato il grafo, che simula la popolazione con un numero fisso di individui ed una probabilità media di contatti tra essi.

Di seguito sono riportati due esempi di scenari che si possono considerare, nell'ipotesi di voler analizzare nel tempo la diffusione di una nuova epidemia, in una popolazione composta da 10000 individui.

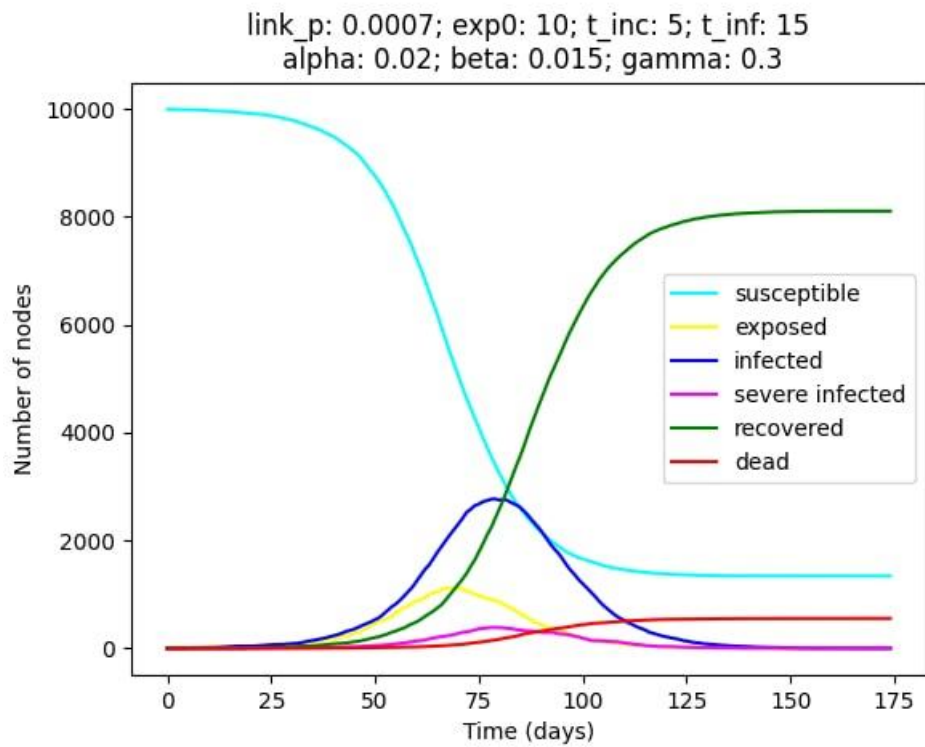
### Scenario 1:

La probabilità media di connessione è 0,0007, il che significa che su 10.000 individui, ogni individuo incontra una media di 7 persone nell'unità di tempo (al giorno).

Ci sono inizialmente 10 individui infetti all'interno del network della popolazione. Le caratteristiche del virus sono tali che il tasso di trasmissione è 0,02 (2%), il tasso di infezione grave è 0,015 (1,5%) e quello di mortalità, nel caso in cui ci si ammala gravemente, è 0,3 (30%).

Il periodo di incubazione del virus è di 5 giorni e quello dell'infezione è di 15 giorni.

Con queste condizioni, il grafico degli stati epidemici risulta così:

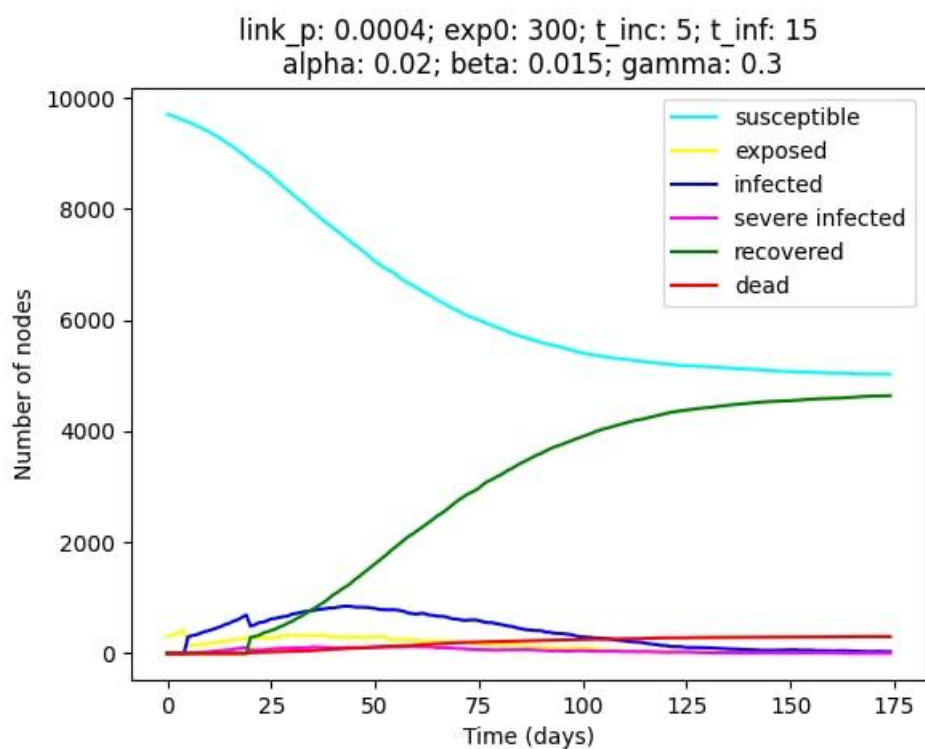


Dopo circa due mesi e mezzo, l'ondata delle infezioni raggiunge il suo picco, quando il numero di infetti raggiunge i 3000 individui. Alla fine dell'ondata epidemica, 8000 individui sono guariti dalla malattia, mentre poco più di 500 individui sono morti. Non vengono raggiunti dal virus, invece, quasi 2000 individui, che rimangono quindi nello stato *susceptible*.

## Scenario 2:

Supponiamo che, ad un certo punto della diffusione della malattia, il governo decida di adottare una misura di restrizione come quella del lockdown. Per simulare questo fenomeno, si può ridurre la probabilità media di connessione durante la costruzione del network. Quindi, se prima un individuo aveva una media di 7 contatti al giorno, si può ridurre la media per esempio a 4 contatti. Questa media è stimata col presupposto che i contatti all'interno del nucleo familiare siano in media di 4 persone.

Quindi, se si interviene dopo un mese, in cui numero di esposti è arrivato a circa 300, il nuovo grafico è il seguente:

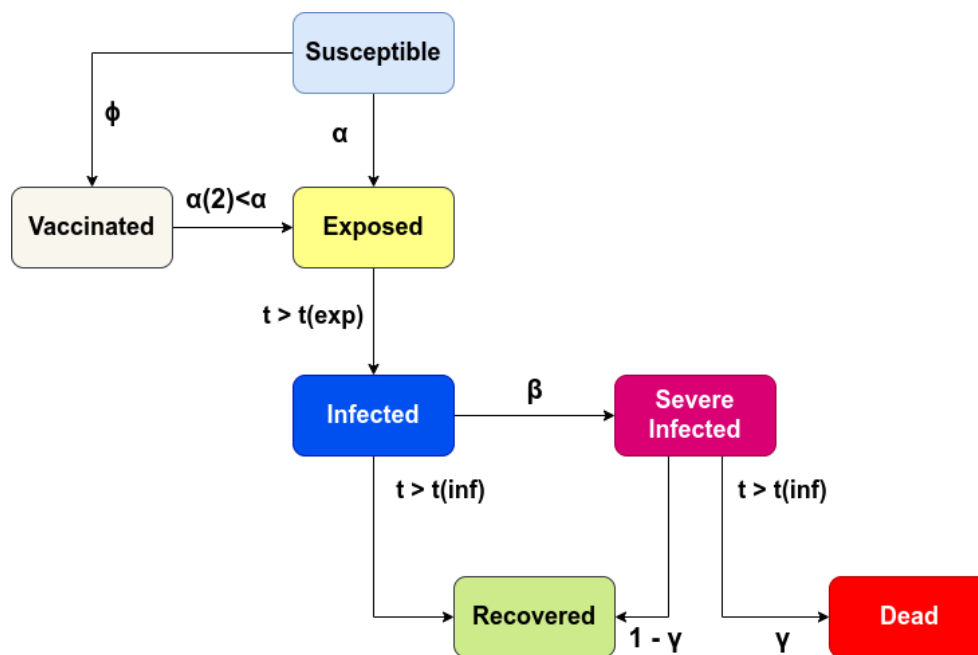


Si può constatare quanto sia efficace la misura del lockdown. Differentemente dall'andamento epidemico precedente, in questo caso la diffusione dell'epidemia risulta più lenta. La metà della popolazione non viene colpita dalla malattia e gran parte dell'altra metà guarisce. Il numero di morti è ridotto di qualche centinaio.

## Possibili sviluppi futuri

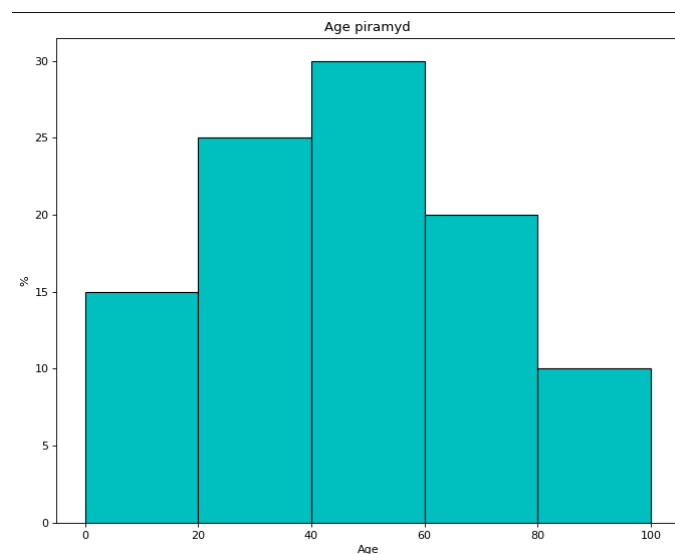
Questo simulatore può essere arricchito con ulteriori elementi, per rendere più verosimile la simulazione della diffusione epidemica.

Si può introdurre un altro stato, quello dei vaccinati. In tal caso lo schema descritto precedentemente diventerebbe:



Supposto che si vaccinino soltanto gli individui suscettibili con un tasso di vaccinazione  $\phi$  fissato in input, chi entra nello stato *Vaccinated*, ha una probabilità di esporsi al virus molto più bassa rispetto a chi non entra in questo stato. La stessa cosa vale per la probabilità di morte per chi proviene da uno stato vaccinato e si è infettato.

Si può inoltre raffinare la simulazione implementando una distribuzione in fasce di età della popolazione, come quella descritta in figura:



In questo modo, una fascia di età avrebbe un tasso di infezione grave e di mortalità diversi da un'altra. Quindi, beta e gamma non avrebbero un valore unico ma sarebbero dei vettori con tanti valori quante sono le fasce di età.

## Riferimenti bibliografici

---

Erdos-Renyi network model, LT03 - Spread of epidemics models, LT11 - Spread of epidemics networked models, LT12. Prof. Salvatore Micciché - Complex Networks 2021/2022. Master in Informatics, University of Palermo.

A. Kuzdeuov, A. Karabay, D. Baimukashev, B. Ibragimov and H. A. Varol, "A Particle-Based COVID-19 Simulator With Contact Tracing and Testing," in *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, pp. 111-117, 2021, doi: 10.1109/OJEMB.2021.3064506.

A. Karabay, A. Kuzdeuov, S. Ospanova, M. Lewis and H. A. Varol, "A Vaccination Simulator for COVID-19: Effective and Sterilizing Immunization Cases," in *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 12, pp. 4317-4327, Dec. 2021, doi: 10.1109/JBHI.2021.3114180.

A. Kuzdeuov *et al.*, "A Network-Based Stochastic Epidemic Simulator: Controlling COVID-19 With Region-Specific Policies," in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2743-2754, Oct. 2020, doi: 10.1109/JBHI.2020.3005160.