UNIVERSITAT ROVIRA I VIRGILI
Department of Computer Engineering and Mathematics
May 2, 2021

CN-MAI Assignment 4

Complex Networks
Spring 2021, 5 Credits

Epidemic Spreading On Complex Networks

Name	Bartosz Paulewicz Betty Törnkvist
E-mail	`bartosz.paulewicz@student.put.poznan.pl` `betty.tornkvist@gmail.com`

Teacher
Alejandro Arenas Moreno

Contents

1 Introduction

Epidemic spreading can be simulated with complex networks. In this paper, we create a Monte Carlo simulation of a Susceptible-Infected-Susceptible (SIS) epidemic spreading model. In the SIS-model, the nodes are either infected or susceptible.

2 Method

To be able to simulate and create comparable results, models needs to be generated. Thereafter, the Monte Carlo simulations can be applied. This section covers the generation and simulation method and setup.

2.1 Graph Generator

One model is already given, `Airports.net`. Using the python library `igraph`, we generate the following models: Watts–Strogatz (WS), Barabási–Albert (BA) and Erdős–Rényi (GNM).

Following the models generation, they are post-processed to ensure that all nodes belong to single component meaning that there is a path between any pair of given nodes.

If a node doesn't belong to the biggest component it gets connected by creating an edge between that node and random node from biggest component. Then to maintain intended number of edges random edges from biggest component are deleted. If during the process of deleting random edges the biggest component splits into multiple components whole process is called again until only one component remains.

The models are then generated in three different sizes:

1. **small** - 500 nodes and 1000 edges
2. **medium** - 500 nodes and 1500 edges
3. **large** - 3618 nodes and 14142 edges

2.2 Monte Carlo Simulation

Monte Carlo simulations can be quite time-consuming, therefore the simulation procedure was first prototyped in interpreted language (Python) and then implemented in the compiled language (C++). The program takes a path to network file as input, reads it and creates an internal representation of the nodes and its neighbours. Random nodes are assigned as infected according to the given fraction $\rho(0)$. The simulation is then run N_{rep} times, with different probability rates of infection β. A simplified overview of the simulation can be seen in Algorithm 1, and the parameter setup can be seen in Table 1.

Algorithm 1 Monte Carlo SIS

```
    Input: $\mu$ - spontaneous recovery probability, $\beta$ - infection probability

1: for node in nodes do
2:     if node infected then
3:         if random number $<\mu$ then
4:             Set node as healthy
5:     else
6:         for each neighbour of node do
7:             if neighbour is infected and random number $<\beta$ then
8:                 Set node as infected
```

Parameter	Description	Value
μ	Spontaneous recovery probability	0.5
β	Infection probability	[0,1] (step size 0.02)
N_{rep}	No of simulation repetitions.	50
$\rho(0)$	Initial fraction of infected nodes.	0.2
T_{max}	Max. number of time steps per simulation.	100
T_{trans}	Transitory no of steps.	90

Table 1: Key parameters for the Monte Carlo simulation.

3 Result

To compare and evaluate the infection spreading within the different setups, we calculate ρ, the average fraction of infected node in the network. The value ρ is calculated as an average over repetitions of averages over time steps. Initially, we try with different values to observe the curve for each network-size. The pattern of all those plots are the same, the smaller the value of the recover probability (μ), the faster the rise of average infected ρ, see Figure 1.

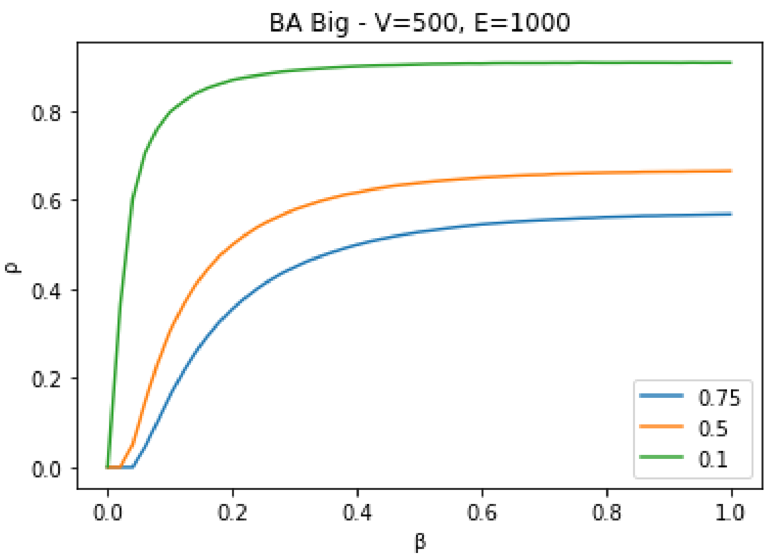
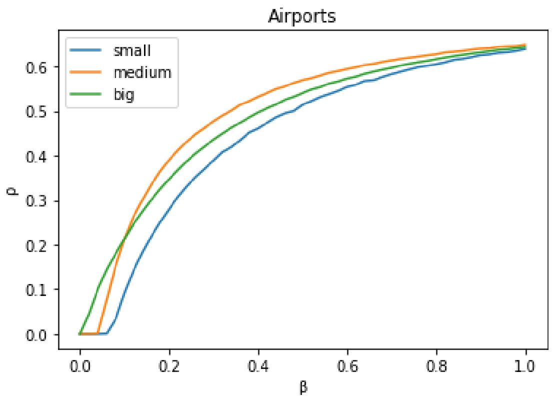
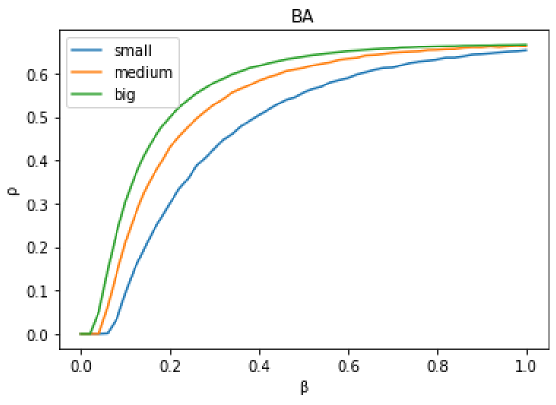


Figure 1: Big BA network with different recovery probability, μ

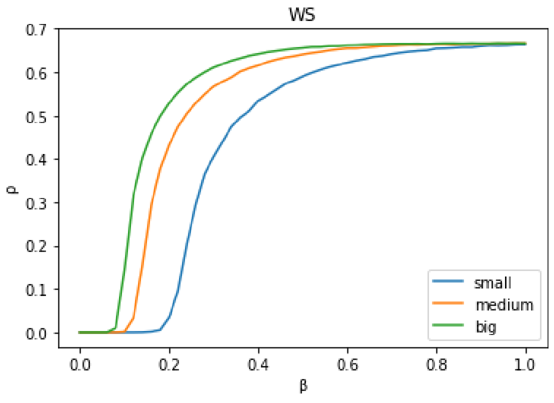
The following simulations are then run with recovery rate $\mu = 0.5$, i.e. a 50% chance of recovery. Figure 2 shows the different networks, where each size is plotted together for comparison. In Figure 3 the plots are grouped according to size, for comparison of the networks.



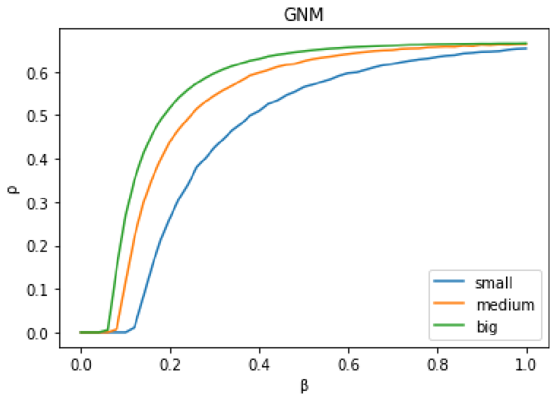
(a) Airports.



(b) BA.

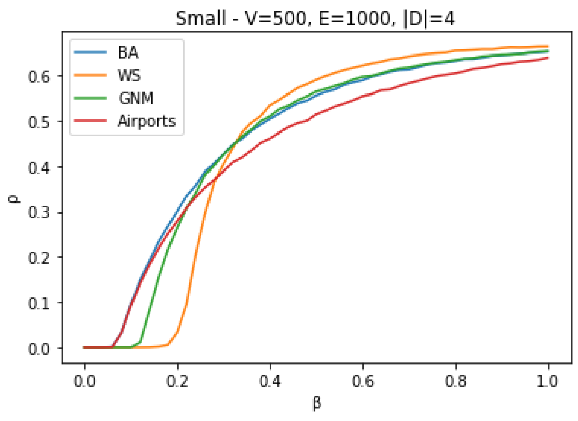


(c) WS.

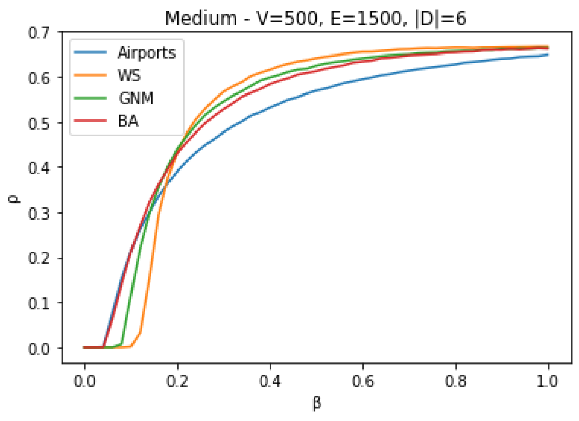


(d) GNM.

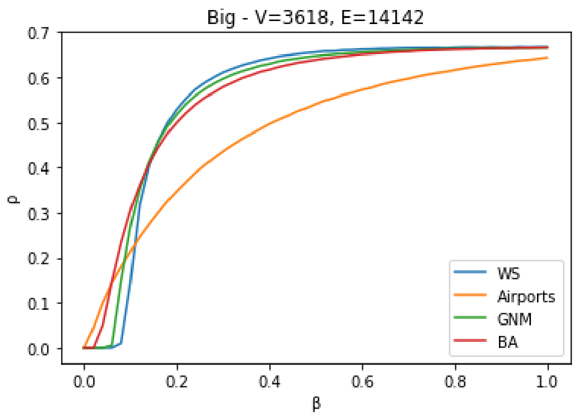
Figure 2: Plots grouped by network type, with recovery probability $\mu = 0.5$. The x axis represents β, the infection probability. The y axis represents ρ, the average fraction of infected nodes in the network.



(a) Small networks.



(b) Medium networks.



(c) Large networks.

Figure 3: Plots grouped by network size, with recovery probability $\mu = 0.5$. V shows the number of vertices, E the number of edges, and D is the average degree of nodes in the network. The x axis represents β, the infection probability. The y axis represents ρ, the average fraction of infected nodes in the network.

4 Conclusion

When comparing the different network versions in Figure 2, it can be observed that all networks except Airports follows a certain curve. The larger versions of the networks grows faster in terms of average infected nodes, followed by medium and lastly smaller. This implies that the in smaller network the start of an infection spreading is slightly slower and demands a slightly higher infection probability. However, when the infection `hits`, all the curves rises exponentially rather than linearly. As mentioned, the Airport network differs. The curves are not as steep as the other networks, and while the big network starts to increase before the medium network, the medium network crosses the big.

The same values were then plotted by grouping the sizes of networks and comparing different models. Here it can be observed in all plots that the WS requires the highest infection probability to start spreading, followed by GNM. BA and Airport starts increasing at the same point in the smaller and medium networks, while in the big plot Airport increases at the first time step. However, the growth of the Airport network is significantly lower compared to the other models. A possible explanation for this could be that Airport has an uneven degree of edges, meaning that a few nodes are well connected. If those well-connected nodes were part of the initially infected fraction, the chance of spreading could initially increase rapidly and slow down when nodes with a lower degree are encountered.