Report Homework 3

Chiara Scagliola
Politecnico di Torino
s303037@studenti.polito.it

Chiara Van der Putten
Politecnico di Torino
s306273@studenti.polito.it

Genuary 2022

1 Exercise

1.1 Epidemic on a known graph

In this first part, it's simulated an epidemic on a symmetric k-regular undirected graph with node set $V = \{1, \dots, n\}$ where every node is directly connected to the $k = 4$ nodes whose index is closest to their own modulo n. The possible node states are: susceptible (S), infected (I) and recovered (R). The parameters considered are the following:
• β: the probability of infection between an infected person and a susceptible one
• ρ: the probability of recovery once an agent has been infected
• m: number of infected neighbours for node i

Assuming that $\beta \in [0, 1]$, the transition probabilities are:

$$\mathbb{P}(X_i(t+1) = I | X_i(t) = S, \sum_{j \in V} W_{ij} \delta^I_{X_j(t)} = m) = 1 - (1 - \beta)^m$$

$$\mathbb{P}(X_i(t+1) = R | X_i(t) = I) = \rho$$

To simulate an epidemic on a symmetric k-regular graph $G = (V, E)$ with $|V| = 500$ nodes and $k = 4$, with parameter values $\beta = 0.3$ and $\rho = 0.7$, one week is considered to be one unit of time and the epidemic is simulated for 15 weeks. Initially, there are 10 infected nodes. The goal is to estimate the weekly evolution of the number of newly infected individuals and the average total number of susceptible, infected, and recovered individuals at each week. In Figure 1 are reported the results of the study.

In the second point of exercise 1, was required to create and implement a random graph according to the preferential attachment model. The goal is to have a randomly generated graph with average degree close to k.

1.2 Epidemic on a random graph with preferential attachment model

The second simulation that has been made regards the use of a preferential attachment random graph, constructed as previously explained. The number of nodes of the new graph is always 500 while the average degree now is k=6, the configuration of the initial ten infected has been chosen as before. The results for each week shown in the graphs (Figure 2) below were obtained on an average of 100 experiments.

As can be seen from the first graph in the upper right, the number of nodes that become infected in the first weeks is much higher than that found previously with the fixed graph. This is due to the fact that the average degree rose to 6 and that in the random graph, we have few nodes with a very high degree that become infected in the first weeks, this involves a propagation of the infection to all the other nodes.

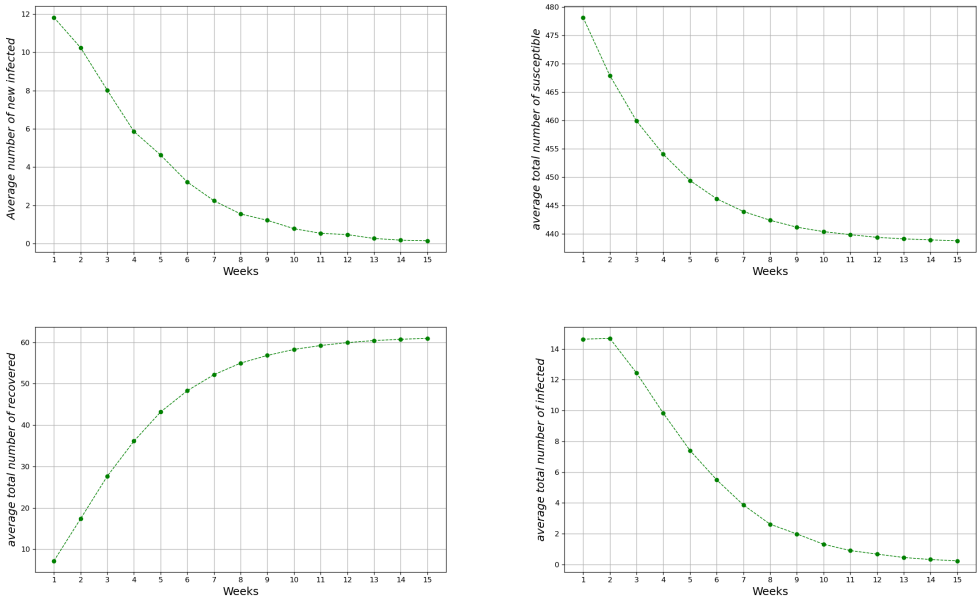

Figure 1: In the image are reported the average number of new infected (upper left), average total number of susceptible (upper right), average total number of recovered (lower left), average total number of infected (lower right)

1.3 Epidemic on a random graph with vaccination

To try to slow down the pandemic in the random graph it was decided to use vaccinations. The total fraction of population vaccinated within 15 weeks shall be:

$Vacc(t) = [0, 5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60, 60]$

The people to be vaccinated in each week were chosen uniformly randomly among those who had not yet received the vaccine. It was also considered that a vaccinated person could no longer be infected and infect, the results obtained are shown in the Figure 3:

As can be seen from the chart, thanks to vaccinations, the number of new infected is halved, also the number of recovered has decreased significantly. This is due to the fact that thanks to vaccinations the number of susceptible decreases, this leads to a reduction in the probability that the node becomes infected and that it may infect its neighbors.

Algorithm 1 Generation of a random Graph

```
procedure CREATEGRAPH(k, n)        ▷ The initial(k) and the final(n) number of nodes
    Gprec = completegraph(k+1)
        for node in range(k+1,n) do
    | calculate the degree and the probability of each node to be chosen as neighbors
    | update the new graph
        if k is even then
    |       add k/2 links
        else
    |       add k/2 or k/2 + 1 links
        end
        add node to Gprec
        end
    add the edges and return the new graph
end procedure
```

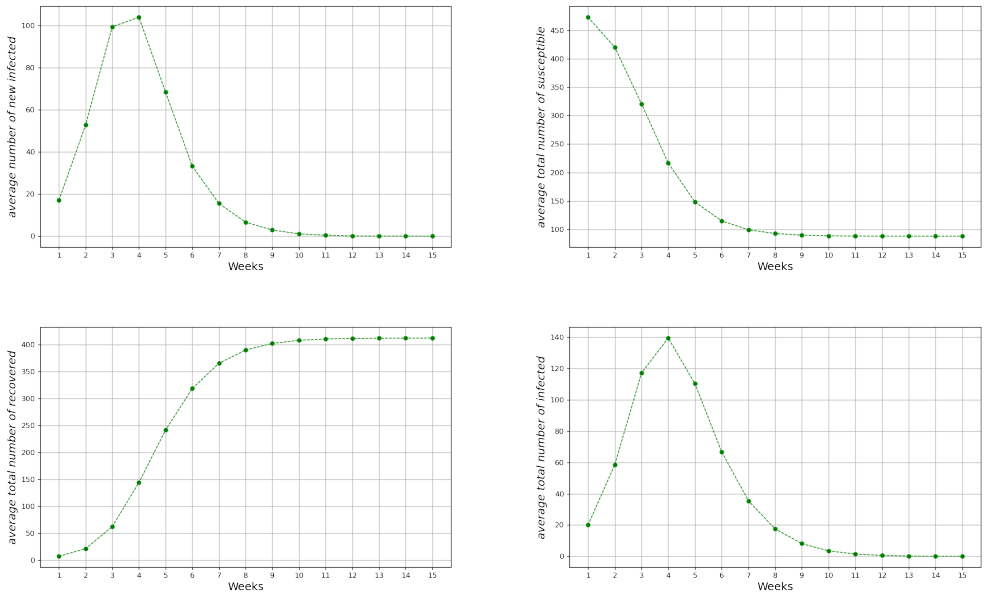

Figure 2: Average number of new infected (upper left), average total number of susceptible (upper right), average total number of recovered (lower left), average total number of infected (lower right)

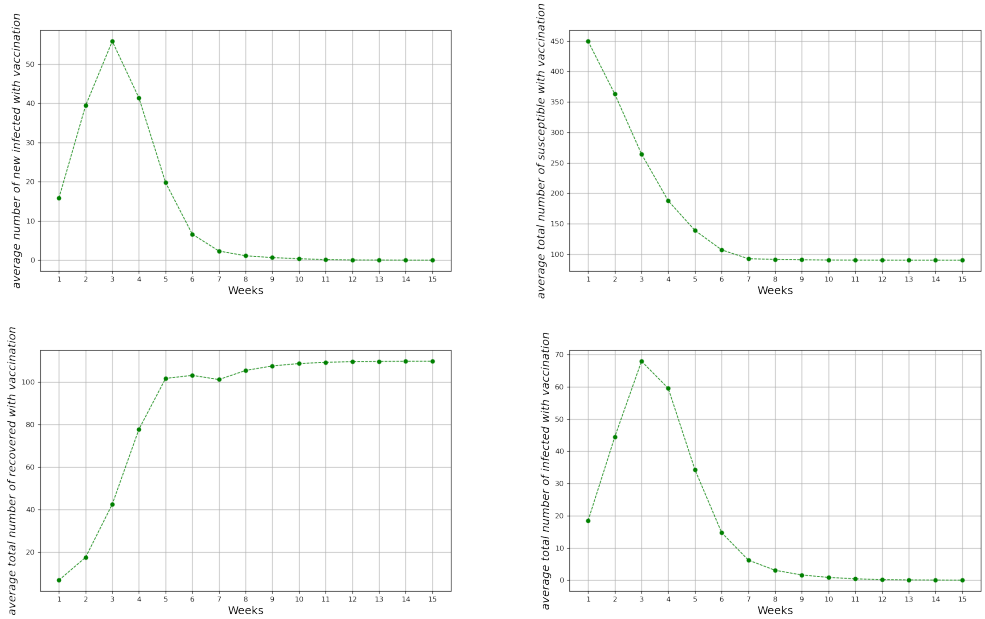

Figure 3: Average number of new infected with vaccination (upper left), average total number of susceptible with vaccination (upper right), average total number of recovered with vaccination (lower left), average total number of infected with vaccination (lower right)

1.4 The pandemic in Sweden 2009

In this part a pandemic will be simulated on the Swedish population in order to evaluate the evolution of the infections. In this case the percentage of population vaccinated in each week is: $Vacc(t) = [5, 9, 16, 24, 32, 40, 47, 54, 59, 60, 60, 60, 60, 60, 60, 60]$. The number of new infected per week is estimated to be: $I_0(t) = [1, 1, 3, 5, 9, 17, 32, 32, 17, 5, 2, 1, 0, 0, 0, 0]$. Since

the weeks on which the RMSE is calculated are 15 while the vectors of the percentage of vaccinated population and of infected animals for each week have 16 elements; it was decided that the first elements of these two arrays represent respectively the percentage of vaccinated population and the number of infected at time t=0 , that is, before the algorithm begins.

An algorithm has been constructed which, by means of a gradient-based search, estimates the best parameters of β, ρ and k which best approximate the values of the newly infected. The starting parameters are k_0= 10, β_0= 0.3, ρ_0 = 0.6 from which at each iteration are obtained new parameters to be tested within the range $k \in \{k_0 - \Delta k, k_0, k_0 + \Delta k\}$, $\beta \in \{\beta_0 - \Delta\beta, \beta_0, \beta_0 + \Delta\beta\}$, and $\rho \in \{\rho_0 - \Delta\rho, \rho_0, \rho_0 + \Delta\rho\}$. Through a grid-search of these parameters 27 combinations are obtained, between which the one that minimizes the RMSE is selected.

$$RMSE = \sqrt{\frac{1}{15}\sum_{t=1}^{15}(I(t) - I_0(t))}$$

Where I(t) is the average number of newly infected individuals each week after 10 iteration. In the case that the RMSE does not improve after an iteration it was decided to reduce the delta by half and move the parametric search around the combination that had returned the minor loss. The algorithm stops when twice consecutively the same set of parameters is selected as best. The found parameters are shown in the Table 1

β	ρ	k
0.2	0.6	9

Table 1: best set of parameters found by the algorithm

The RMSE obtained with these values is 3.15 and the difference between I_0 and $I(t)$ is represented in Figure 4

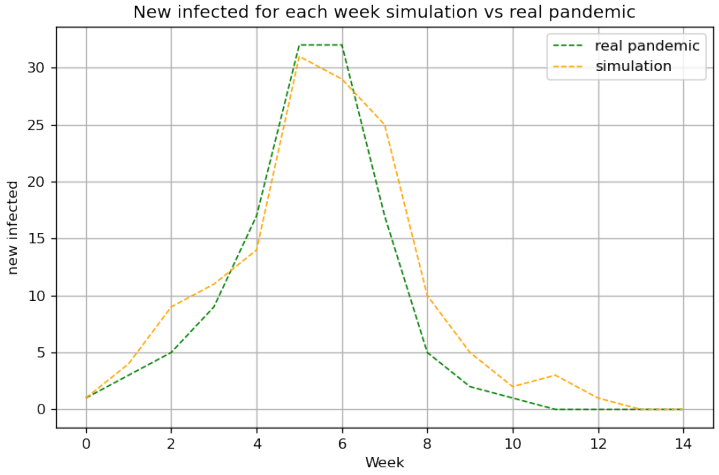

Figure 4: in orange is shown the number of new infected according to the simulation with preferential attachment model during the weeks and in green the actual number of new infected.

The average total number of susceptible and recovered individuals at each week according to the model with the above parameter are shown in Figure 5.

1.5 The pandemic in Sweden with small word graph

A new run to search for the best parameters is conducted on the Newman-Watts-Strogatz small-world model. It depends now on n and other two parameters, k and p:

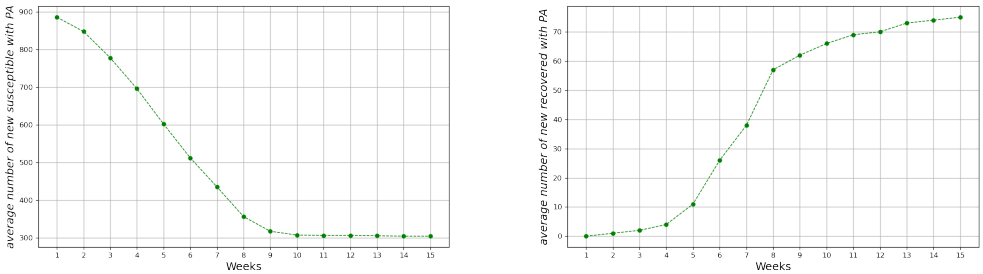

Figure 5: average total number of new susceptible(left) and recovered (right) individuals at each week

- n is the number of nodes
- k is the number of closest nodes each node is connected to;
- $p \in [0, 1]$ is the probability that a long-range connection between two random nodes i and j in the graph exists.

The Newman-Watts-Strogatz, first create a ring over n. Then each node in the ring is connected with its k-nearest neighbors (or k-1 neighbors if k is odd). For each edge in the ring with probability p add a new edge with randomly-chosen existing node, no edges are removed. The result obtained, by simulating with the same algorithm previously explained, is shown in Figure 6. The minimum MRSE obtained with this new model is 5.71 with the parameters in Table 2

β	ρ	k	p
0.35	0.5	10	0.05

Table 2: best set of parameters found by the algorithm

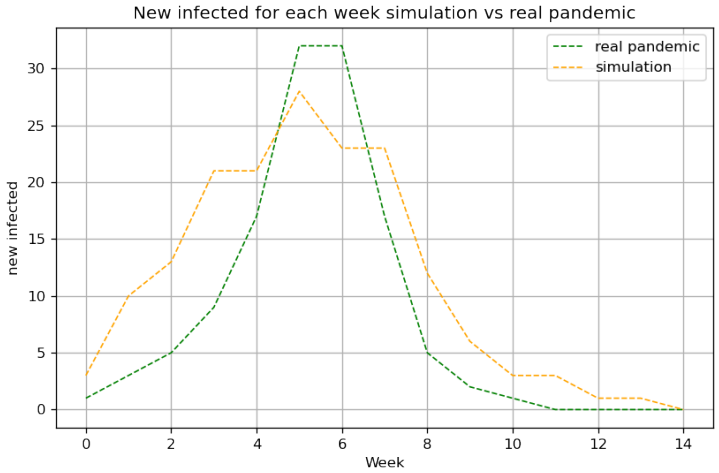

Figure 6: in orange is shown the number of new infected according to our simulation with the small world model during the weeks and in green the actual number of new infected

2 Exercise

2.1 Coloring on a linear graph

Now consider a linear graph composed of 10 nodes as in Figure 7, and suppose we have two possible states { red, green }. At the beginning, all nodes are initialized as red and at each iteration, a node is randomly selected to decide whether to change the color.

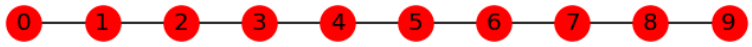

Figure 7: initial linear graph

The probability that the selected node changes color is given by the following formula:

$$P(X_i(t+1) = a | X(t), I(t) = i) = \frac{e^{-\eta(t)\sum_j W_{ij} c(a, X_j(t))}}{\sum_{s \in C} e^{-\eta(t)\sum_j W_{ij} c(s, X_j(t))}}$$

where c represent the cost function and is equal to:

$$c(s, X_j(t)) = \begin{cases} 1 & if \;\; X_j(t) = s \\ 0 & otherwise \end{cases}$$

while $\eta(t)$ is the inverse of the noise and is equal to: $\frac{t}{100}$. The algorithm stops when all nodes have as neighbors nodes with a color different from theirs, that is when the potential function: $U(t) = \frac{1}{2}\sum_{i,j \in V} W_{ij} c(X_i(t), X_j(t))$ is equal to 0. A brief description of how the Algorithm 2 works is illustrated below:

Algorithm 2 Coloring with two possible state

```
while potential function != 0 do
    choose a node i randomly;
    take a uniformly random v value between 0 and 1;
    calculates the probability p that node i will turn green according to the color of the
    neighbors;
    if v < p then
        change/leave the color of i to green;
    else
        change/leave the color of i to red;
    end
end
```

Figure 8 shows the trend of the potential function over time.

As can be seen in this case the potential after a certain number of iterations reaches 0, this is possible because the type of linear graph allows it, this is not always possible as we will see for the next graph. The configuration of the graph when the potential reaches 0 is illustrated in Figure 9 :

2.2 Coloring on the router network

The same process as before is now applied to the network illustrated in Figure 10. As before, all the nodes have been initialized to the red color, while the possible states are { red, green, blue, yellow, magenta, cyan, white, black }. The cost function is now equal to:

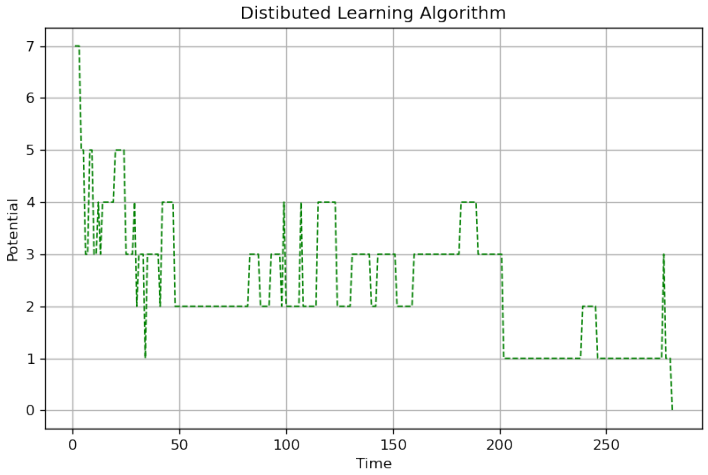

Figure 8: trend of the potential function over time

Figure 9: configuration of the graph when the potential reaches 0

$$c(s, X_j(t)) = \begin{cases} 2 \ if \ \ X_j(t) = s \\ 1 \ if \ \ |X_j(t) - s| = 1 \\ 0 \quad otherwise \end{cases}$$

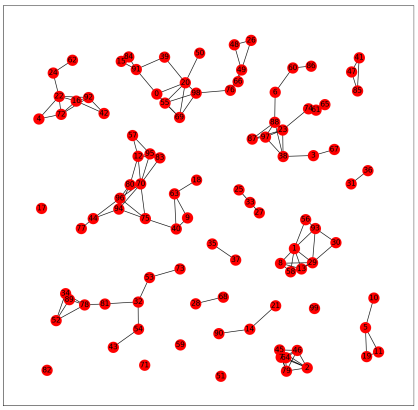

Figure 10: wi-fi network

The Figure 11 shows the trend of the potential function over time for three simulations. In this case, the algorithm stops after 1000 iterations since, as can be seen on the chart, the potential function never reaches 0. This is due to the fact that the colors are not enough

to ensure the absence of conflict between routers, to achieve the potential of 0 three more colours are required. The final configuration obtained is represented in Figure 12.

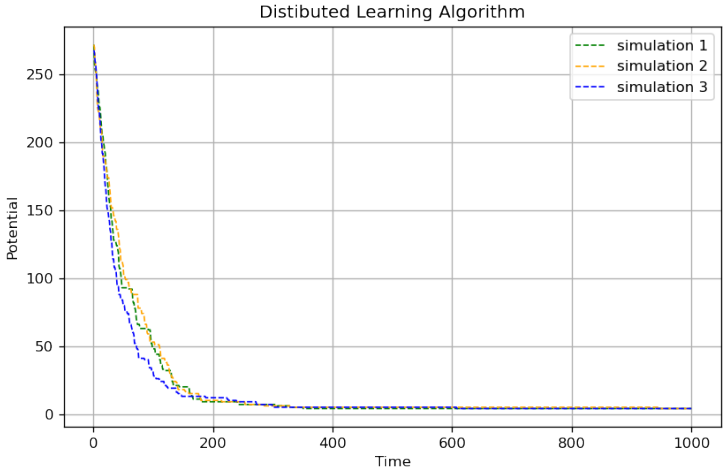

Figure 11: behaviour of the potential function in three different simulations

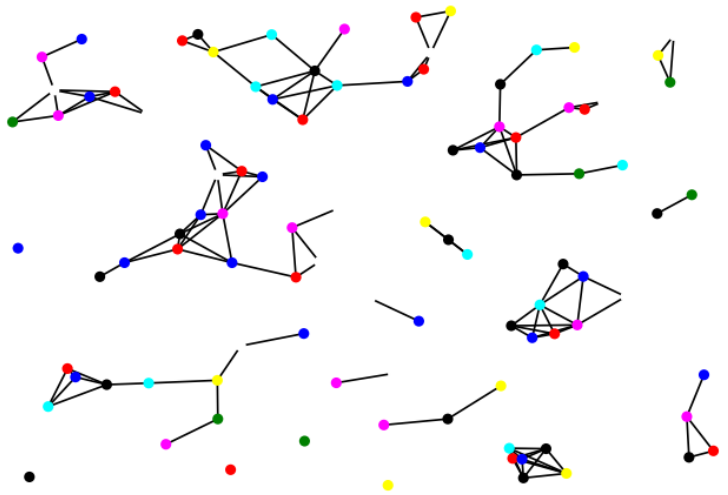

Figure 12: wifi network with the minimum number of colours required to achieve zero potential

2.3 Coloring changing $\eta(t)$

For different choices of $\eta(t)$ the graphs represented in Figure 13 are obtained. As can be seen from the image choosing a very low $\eta(t)$ the potential function never converges to four and has a random trend, this is due to the fact that by choosing very low values the probability of changing one color rather than another will all be equal. While choosing a larger n as can be seen $U(t)$ converges to 4. In the case of fixed n at a higher value $U(t)$ converges, the number of iterations in which it converges depends on the chosen value, for example for fixed $\eta(t)$ at 300 the potential function converges first with respect to the logarithmic function.

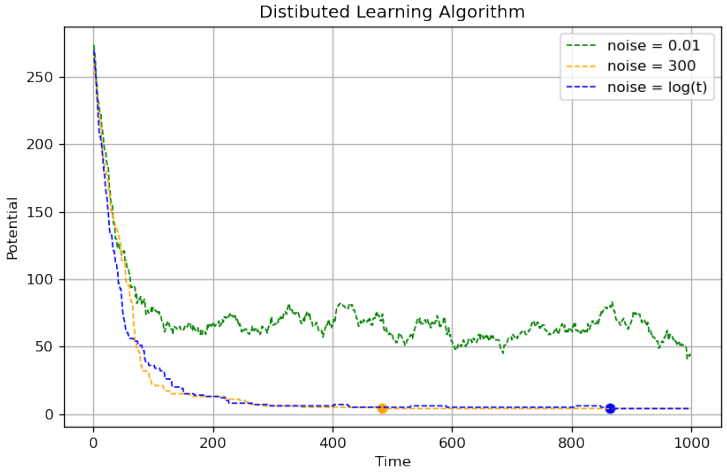

Figure 13: representation of the potential with a very small noise (0.01, in green), with a large noise (300, in yellow) and a growing function(log(t), in blue). The dots in the image indicate the first moment when the minimum potential is reached.