

Epidemic Simulation based on Dynamic Multi-Layer Network

Final Project

Course: Network Science 2020-21
Professor: Erseghe Tomaso

Ma Jiawei

February 11, 2021

1 Introduction

Epidemiology has a long history starting from the 17th century [1] and it's still a well-known field study, especially in this moment (2020 – 2021), when the entire world is facing a new pandemic emergency: COVID-19.

This field research involves several other scientific branches; for instance, biology, chemistry, medicine and also mathematics. Traditional mathematical epidemiology aims to develop proper models to explain observations or to predict outcomes. One of mainly used mathematical tools is the *differential equation theory*. Although sometimes mathematical theories are able to carry out an explicit solution to a epidemic model problem (temporal function), nevertheless we should pay attention to what we can conclude from the results, because the initial model, no matter how complicated, is *not perfect*.

The modern theoretical epidemiology relies on other available techniques and one of them is *network science*. There are many ways to exploit this young research area since it's quite intuitive how the *social interactions* are relevant in disease spreading, the most simple approach could be to perform *stochastic simulations*, it's a powerful way to study complex networks such as real-world social networks. In this work we implemented the *Temporal Gillespie Algorithm* to simulate a the so called *SIQRD* epidemic model and the results are compared with the real data collected at the very beginning of *COVID-19 situation in Italy*. Finally, having this simulator, it was interesting to investigate on the *robustness; immunization and attacks*.

2 SIQRD model

The SIQRD model is based on the notable SIR model and the derivation of this epidemic model can be found in [3]. The additional letters in the model's name have specific meaning: Q stands for *quarantine* and D stands for *death*. This new model is quite well-explained by the fig. 1 and below we describe each *state*:

- **Susceptible(S)**: it's the initial state of every subject (node of the network) and the susceptible subjects can be infected or quarantined if they got in touch with infected people.
- **Infected(I)**: the infected subjects can change the state to other three states: death, quarantine and recovery.
- **Quarantine(Q)**: this extra state is to consider many real situations, for instance, one person is quarantined whenever he/she had been in contact with an infected person or simply he/she is tested as

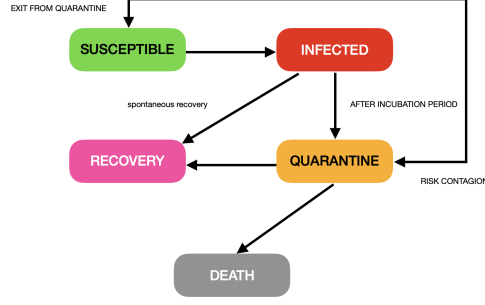


Figure 1: SIQRD model

infected.

- **Recovery(R)**: once the subject is recovered he/she becomes immune to the virus.
- **Death(D)**: we suppose that this state is followed only by the Q state because usually ill persons get cures before death.

3 Temporal Gillespie Algorithm

This method is flexible and applicable to many situations, so first we briefly describe the general framework and then we'll give a pseudocode of our implementation to the SIQRD model. The mathematical derivation details related to this algorithm are in [2].

3.1 Gillespie Algorithm

The Gillespie Algorithm (also known as *SSA* or *Gillespie's direct method*) made popular by *Daniel Gillespie* [4] for the simulation of coupled chemical reactions and then exploited by other scientific study (like economics or physics). It belongs to the big family of *stochastic simulations* and it's applicable to general *Poisson* processes with constant rate, namely, every process in which events occur *independently and randomly*.

It's a suitable simulation algorithm for the spreading phenomena since the infection events could be considered as random events and therefore they follow *Poisson distribution*. The main idea of this algorithm is to reasonable assume that *the waiting time (between state transitions) is exponentially distributed* [5] and that each change of state happens with a probability that is proportional to its rate.

It's a very intuitive method and we can describe it formally. Suppose that the possible events are finite set Ω and the waiting time τ for an event $m \in \Omega$ follows the exponential distribution:

$$p_m(\tau) \propto S_m(\tau) = \exp(-\lambda_m \tau),$$

where λ_m is the m event's transition rate. Due to the independence between events occurrences, if now λ is the waiting time for the first event, we can write:

$$p(\tau) = \Lambda \prod_m S_m = \Lambda \exp(-\Lambda \tau),$$

that is the joint probability distribution and where $\Lambda = \sum_m \lambda_m$. Thus, the waiting time probability for a specific event m is $p_m(\tau) = \lambda_m \exp(-\Lambda \tau)$. In practical terms, when we implement this algorithm, we consider the waiting time $\tau \sim \text{Exp}(\Lambda)$ for the next transition and the probability that m is the next event equal to $\pi_m = \lambda_m / \Lambda$.

3.2 Dynamic network

we haven't yet taken into account the dynamic structure of the real social networks and in doing so in [2] it was proposed an extension of Gillespie's algorithm: *Temporal Gillespie Algorithm*.

First of all we make events space Ω to be dependent on time and we denote the new set $\Omega(t)$ and following a similar approach to what we did in the static network case we obtain the waiting time joint probability distribution

$$p(\tau; t^*) \propto S(\tau; t^*) = \exp\left(\int_{t^*}^{t^{**}} \Lambda(t) dt\right),$$

where $\Lambda(t) = \sum_{m \in \Omega(t)} \lambda_m$, t^* is the time in which the last transition took place, $t^{**} = t^* + \tau$ is the time when the next transition takes place. We note that now $\Lambda(t)$ depends on time since the network's structure changes and so we have no analytical solution because the dynamics of the network considered is generally unknown or complicated. Nevertheless, we can still implement the algorithm numerically. Algorithm 1 is a possible pseudocode for simulating the SIQRD model.

Input: μ_0 = initial recovery rate, γ_0 = initial death rate and q_0 = initial quarantine rate.

- 1: Initialize the rates $\mu \leftarrow \mu_0$, $\gamma \leftarrow \gamma_0$, $q \leftarrow q_0$.
- 2: New social contact network is created.
- 3: Draw a normalized waiting time until the first event from a standard exponential distribution, $\tau \sim \text{Exp}(1)$.
- 4: Find out the number of persons who are in contact with infected subjects and multiply this number by β_0 . So we obtain β .
- 5: Compute $\Lambda = \mu + \beta + \gamma + q$.
- 6: **if** $\Lambda < \tau$ **then**
- 7: No events occurred before new contact network was created. Compute the difference of time: $\tau \leftarrow \tau - \Lambda$.
- 8: Return to step 4.
- 9: **else**
- 10: A transition occurred. Initialize remaining fraction of time before the change of social contact network: $r \leftarrow 1$.
- 11: **while** $r \times \Lambda \geq \tau$ **do**
- 12: Randomly select the transition $m \in \{I, R, D, Q\}$ with probability $\beta/\Lambda, \mu/\Lambda, \gamma/\Lambda, q/\Lambda$ respectively.
- 13: Update values and parameters for transition m . For instance, if the transition I takes place a subject is infected and update network's parameters.
- 14: Update β, μ, γ, q according to which event occurred and remaining time: $r \leftarrow r - \tau/\Lambda$.
- 15: Recompute β and Λ . Draw a normalized waiting time $\tau \sim \text{Exp}(1)$.
- 16: **end while**
- 17: Time expired and new social contact network is coming up. Return to step 2.
- 18: **end if**

Algorithm 1: Temporal Gillespie Algorithm applied to SIQRD model.

4 Dynamic social network

Recently several research works about COVID-19 spreading have emerged and some of them leveraged the network structure of social interactions, generating a proper social network for simulations or even for theoretical purposes. In [6] a possible virus spreading network model is presented, such model is a *multi-layers network model* and each layer represents a particular social community such as workplaces and households. See fig. 3 taken from [7]. The model considered human-human interactions (*social interaction network*) and also *lockdown* strategies that many countries adopted during the pandemic. In the paper many lockdown situations are analysed and some of them are revealed to be more effective than others in mitigating the spreading.

Here, based on the work done in [6], we modified the original static network into a *dynamic network*;

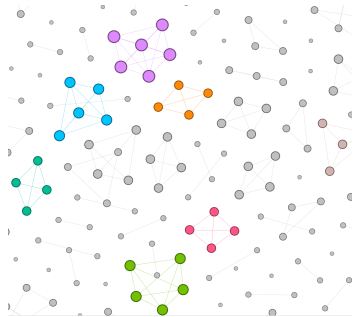


Figure 2: Household layer: two nodes are connected if they are family members.

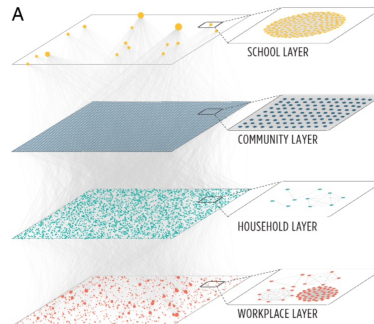


Figure 3: Example of a multilayer social network structure.

dynamics are due to the random interactions between the populations while we kept fixed the layers. We built this network model by taking into account italian social structure and then we applied Temporal Gillespie Algorithm to the SIQRD model. We want to stress that it's difficult to map every single citizen because it would require enormous computational power, we believe that it's not necessary for studying the *general situation* of the pandemic. Thus, it's sufficient to build a *mini-scale* network which simulates the real-world society and at the end analysing the results carefully, keeping in mind the differences in term of networks' size.

Table 1 summarizes the features of the layers included in the model. The workplace layer was also intended to resemble the school layer described by fig. 3.

Layer	Type	Parameters
<i>HOUSEHOLD</i>	<i>Fully connected</i> between family members	Families proportion = [0.333, 0.271, 0.193, 0.151, 0.04, 0.013]
<i>WORKPLACE</i>	<i>scale-free</i> (non-ess.), <i>random</i> (ess.)	ba-degree = 4, ess-prop = 0.2
<i>INTERACTIONS</i>	<i>random</i>	interaction-prob. (with ess. workers) = 0.2, social-prob. = 0.01

Table 1: Layer type summary.

4.1 Household layer

The italian family composition data are reported by ISTAT¹ and we set up a household network in which the nodes belong to the same family are *fully connected*. The fig. 2 illustrates an example result. There are families with size 1, 2, 3, 4, 5, 6 and the proportions of these families are written in Table 1.

4.2 Workplace layer

In order to simulate the real-world social network we constructed two different workplace layers: *non-essential* workers' network and *essential* workers' network. We again rely on the informations come from ISTAT.

Non-essential workers' network

Non-essential workers are those people who don't work in a necessity good sector, some examples are

¹<https://www.istat.it>

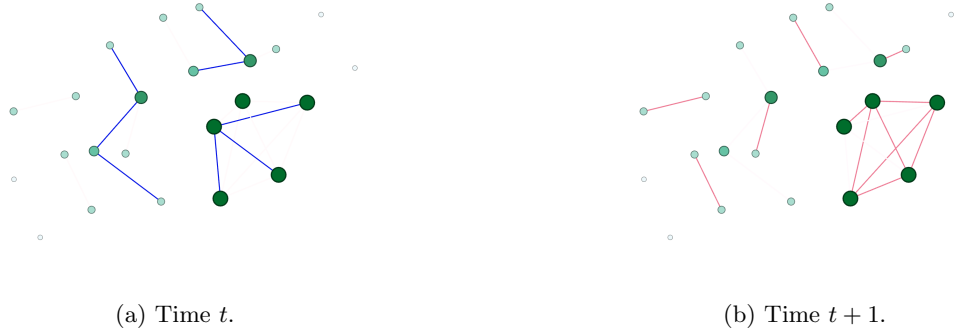


Figure 4: Interactions between nodes.

travel agencies and clothing stores this specification is relevant in our model because they are not allowed to work during the lockdown as happened in many countries. It's reasonable to assume that this network is a *scale-free network* and to use the *Barabasi-Albert model (preferential attachment)* for generating the network, since it resembles other scale-free networks such as internet, protein and actor networks [8]. The fig. 5 illustrates the nodes degree distribution of a sample network. As expected it has a evident scale-free trend and the maximum degree is about 100. Majority of the nodes are connected with few other nodes.

Essential workers' network

While scale-free property is suitable in case of non-essential workers networks here we decided to model the essential workers network as a *random network* and to use the *Erdős-Rényi model* for generating the network. The fig. 6 shows the nodes degree distribution of a sample network. The median degree is about 3 and it's a proper network to simulate the real essential workers' network.

4.3 Interactions

The network blocks we have just explained are static, the interactions among the people are the dynamic part of the whole network, this means that at time t we *add an edge between two nodes* if they are in contact at time t and this relationship can disappear at time $t + 1$, the fig.4 illustrate this situation. Here we simply fixed an interaction probability and randomly connected two nodes according to that probability. The choice of this latter probability is not trivial, we should redefine it if we obtain bad results.

We divided the interaction in two types: interaction between normal persons and interaction between normal persons and essential works. This distinction was to simulate the scenario in which during the lockdown period the citizen are allowed to exit the own habitation only for necessities.

5 Simulation and analysis

In this section we want to show the results of simulations performed and to compare these results with the italian real contagion data². As pointed out before we don't expect the model can 'predict' perfectly the contagion numbers, precisely the proportions w.r.t the real population, during the time but we hope that it can *capture the general trend* (increasing or decreasing) and the goal is actually achieved. Then we wanted to go deeper analysing how the nodes with different degree are involved during the spreading of the virus. The period of time examined started from 2020/02/21 to 2020/06/21 and we considered also the lockdown that the italian government imposed (2 months). In some cases in order to eliminate randomness we ran the algorithm many times and computed the average results.

²<https://lab24.ilsole24ore.com/coronavirus/>

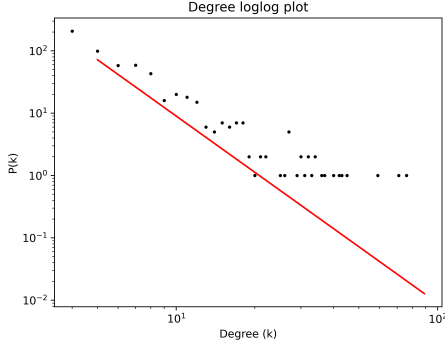


Figure 5: **Non-essential** workers degree distribution; simulated by 2500 nodes.

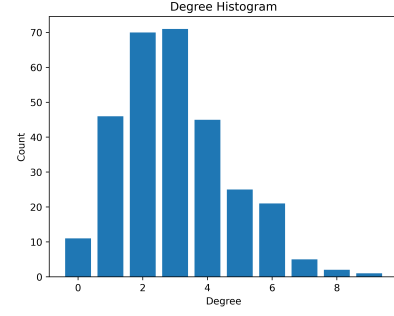
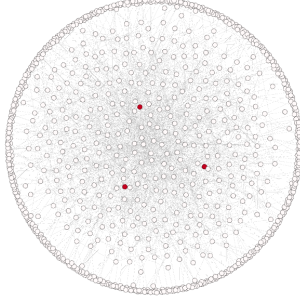
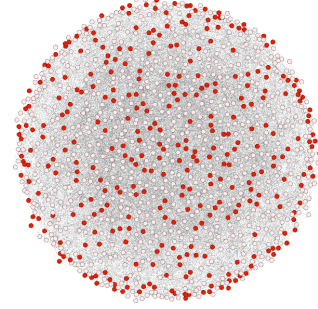


Figure 6: **essential** workers degree distribution; simulated by 2500 nodes.



(a) Time t_0 .



(b) Time $t_1 > t_0$.

Figure 7: Example of virus spreading.

5.1 Simulation method

The fig. 7 illustrates how the virus spread at the beginning during one of the simulations. We want to stress also the fact that one node can be 'quarantined' only after a certain interval of time: *incubation period*, that has as mean value equal to 14 days. Finally a subject exit from quarantine status only if it's passed some time: *observation period*. The complete algorithm implementation can be found in the attached files.

It's interesting to observe how often the events take place; the fig. 8. It's quite intuitive that few days after the lockdown started we had the maximum number of events occurred due to the infection peak. It's also clear that during the lockdown period events occurrence rate is lower immediately after the lockdown and then it increases mainly due to non-infection events (such as quarantine or recovery). Finally we remark that the most probable event occurrence rate is about 60, it's what we can understand from the density plot.

5.2 Results

The fig. 9 contains what we obtained from 12 simulations. There is no perfect match between the real data and the simulation data but it's an expected outcome since the real contagion numbers could not be *real-time*; at the beginning of the pandemic there was no enough resources to test any people having the COVID symptoms (some patients were even asymptomatic) and so the number reported were related to how many tests were executed. In fact, as we can observe the contagion number had no drastic drop but it decreased over the time when more people were tested. The trend could be affected by other factors, for instance how the infection numbers were counted or how the data were collected. That's said, we can conclude that our model is *satisfactory* because it was able to capture the *increasing/decreasing trend* of

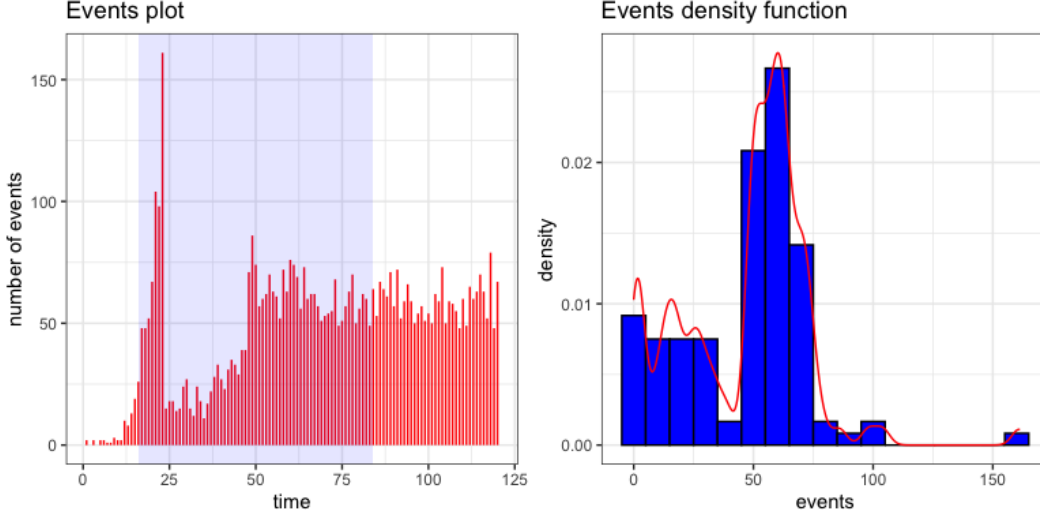


Figure 8: Events occurrences. The data are drawn from one of simulations performed.

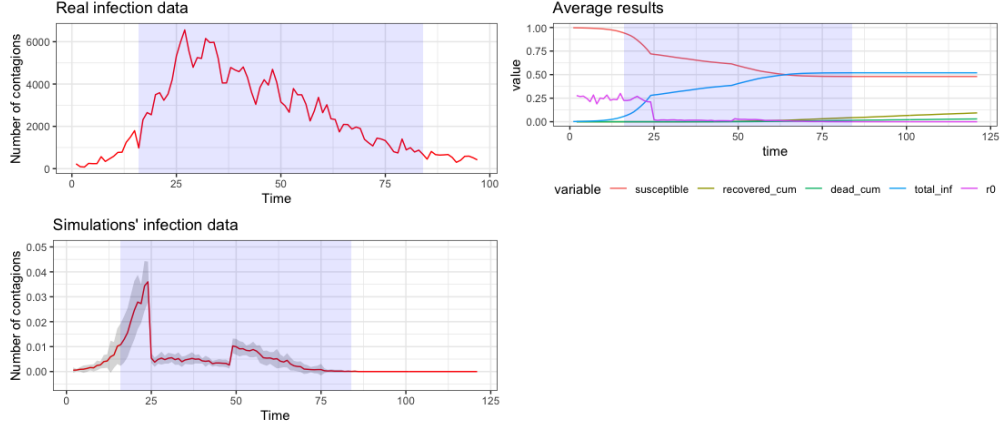


Figure 9: Average results from the performed simulations. with 1500 nodes. The simulation values are the percentages w.r.t the overall nodes. Shadow area indicates the lockdown period.

the time series, especially the *contagion peak*. The model can even simulate the virus spreading during the lockdown. Indeed, the infection events can take place due to the interactions between people and essential workers or subjects within same family.

As last comments about the results, we can verify that the lockdown emergency protocol can be really effective as showed in [6] and by the real data, we can also note that at the end of the simulation there were almost no infected nodes, nevertheless we cannot conclude the pandemic is over since the model considered *only a closed system* in which there are no external accesses to the system, this may cause a second contagion wave as really happened.

5.3 Spreading speed on different nodes

In [8] there is mathematically derived, in the simple SI epidemic framework and network model (such as random and scale-free network model), a formula describing the virus spreading speed on nodes with different degrees, the formula just validated our intuition, that is the nodes with high degree have strong likelihood to be infected. Unfortunately, the model SIQRD considered is a quite complex model and so it's very difficult to carry out a mathematical formula of spreading speed. Nevertheless we still use our algorithm to take a look to the simulation results.

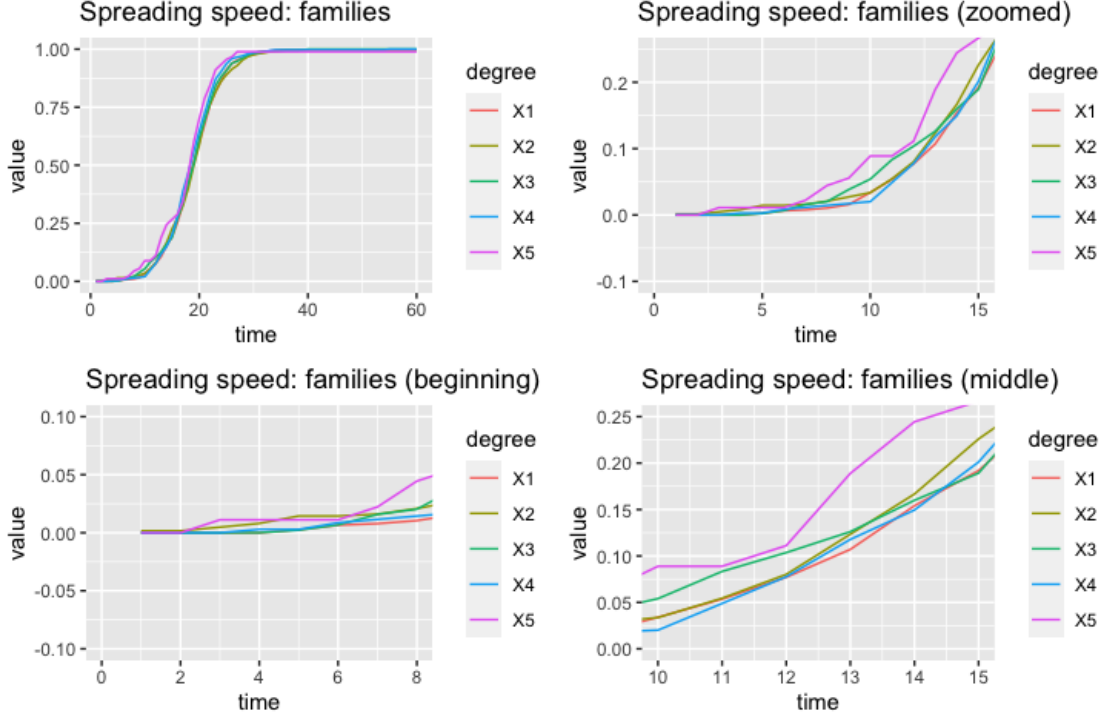


Figure 10: The virus spreading speed within household layer; 2500 nodes. No lockdown was imposed.

In fig. 10, 11 are reported the outcomes from a *no lockdown* simulation with 2500 nodes. The fig. 10 is about infections within families with different size. We can note that among the households with 5 members the virus spread more rapidly than others, but we cannot see a clear differences between the rest; at the first days the curves gone up with almost same speed, during the middle days we observe that the households with 3 members are more affected by the virus, however there were not big differences. We can certainly improve the results by increasing the number of the nodes, but it'll cost too much in terms of computational power. On the other hand the fig. 11 visualizes similar data drawn from the simulation but the degree now is computed dynamically, it means that the degree of a node can change according to the dynamic structure of the network (interaction). Here is more complicated to analyse the results. Nevertheless, we can still distinct the curves related to nodes with large degree difference; the curves associated to the nodes with degree 30 saturated (all the nodes with that degree are infected) before the others as we highlighted in the figure. Also in this case we may need more nodes in order to differentiate all the curves.

6 Robustness

We have before analysed the virus spreading speed on different nodes and the metric or *centrality measure* we used to group the nodes was the simple nodes' degree, it's of course interesting also to understand which roles other centrality measures play in our epidemic model, for instance, they can help to study the network's *robustness*. In our case study there are two concepts related to the robustness; the first is *immunization* and the second is *infected nodes attacks*. Thus, in this section we show the results of the experiments performed in which we 'attacked' the network in some different ways mainly based on three distinct but related centrality measures; *betweenness*, *pagerank* and *closeness*. Obviously, our goal was to carry out a strategy that is able to stop or at least mitigate the contagion curve. For computational issues we considered only 1000 nodes and 60 days simulation in our experiments and without lockdown restriction.

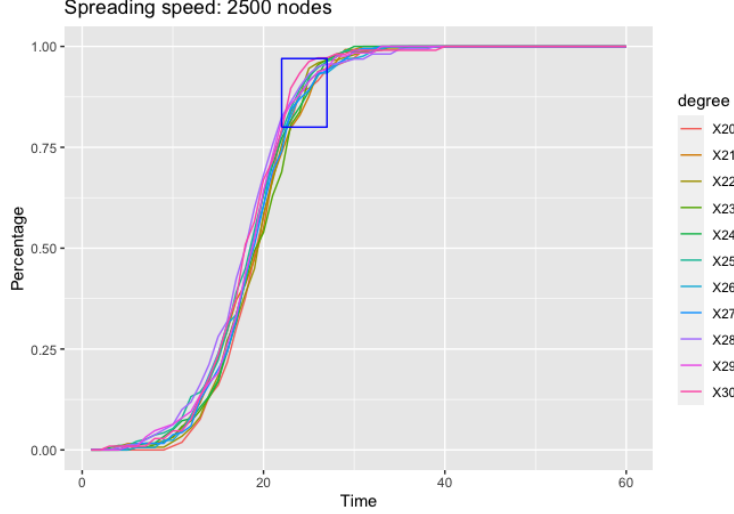


Figure 11: The virus spreading speed across the entire network; 2500 nodes. Note that here the nodes degree are not fixed due to the dynamic structure of the network and the values are computed by dividing the data with the respective saturation values. No lockdown was imposed.

6.1 Centrality measures

From a computational point of view it's very expansive keeping track of every infection events and compute the desired centrality measures. Fortunately, only recording the first infection we can recognize some patterns. The centrality measures of our interest are the following:

- **Betweenness:** it measures the centrality of a node computing all the *shortest paths* and denoting those across through the node. It can be seen as a vital node connecting two communities. The betweenness of a node v is calculated as:

$$B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma(st)},$$

where σ_{st} is the sum of shortest paths from node s to the node t and $\sigma_{st}(v)$ is the number of those paths that pass through the node v .

- **Pagerank:** it's a very popular centrality measures used also by Google research engine and it's a *eigenvalue based measure*. The underlying assumption is that more important nodes are likely to receive more links from other nodes and in a undirected we simply consider an undirected edge as two directed edges.

The basic formulation of this measure is

$$r = cMr + (1 - c)q,$$

where r is probability vector (sum of it's elements is equal to one) indicating score of each node, c is called damping vector, M is the normalized adjacency matrix and q is called *teleportation vector*.

- **Closeness:** also this measure is based on shortest paths and it's normalized and fully connected form is

$$C(v) = \frac{N - 1}{\sum_t d(v, t)},$$

where $d(v, t)$ is the *distance* (length of the shortest paths) between v and t . When the sum of distances is small then $C(v)$ is high as we desire.

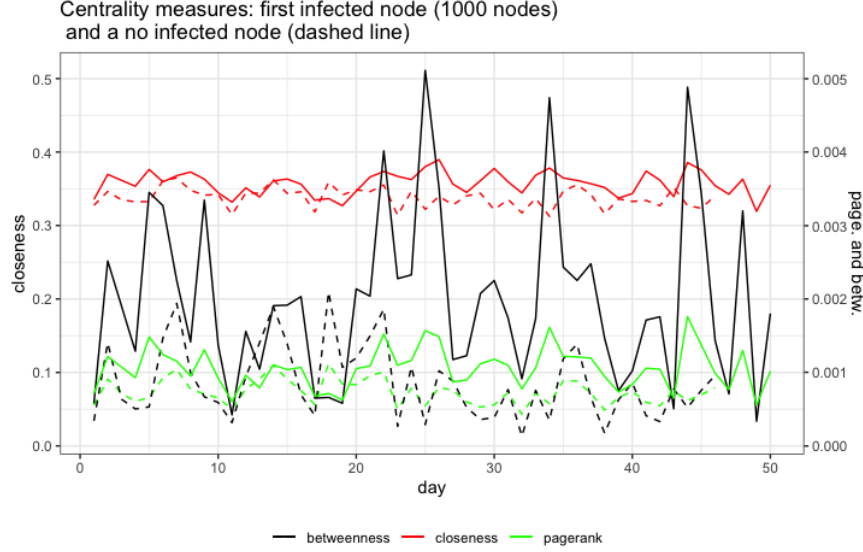


Figure 12: Centrality scores of the first infected node at each time step and a random no infected node scores.

When the network is not fully connected we can use the convention $1/\infty = 0$ and write:

$$C(v) = \sum_{v \neq t} \frac{1}{d(v, t)}.$$

The fig. 12 illustrates the data we obtained and the Table 2 the related statistics.

We can deduce that pagerank and closeness measures are less variable and this may imply that the nodes with a certain centrality score are more affected by the disease than others. We can see also that the closeness measure had lowest standard deviation and this may mean that 'attacking' the nodes having the higher or equal closeness score can slow down the spreading process, same for the pagerank score. Moreover, we verified that a no infected node, in all three cases, was less central than the first infected nodes, especially for the pagerank score, implying that the values detected were significant.

Centrality measures	Min.	Max.	S.D.	Mean
<i>Betweenness</i>	0.00034	0.005	0.001	0.002
<i>Pagerank</i>	0.00056	0.0018	0.00028	0.001
<i>Closeness</i>	0.32	0.39	0.016	0.36

Table 2: Statistics related to first infected nodes.

6.2 Immunization

The vaccination is the main strategy adopted in the real world to tackle an epidemics and how to effectively vaccinate the population is not a easy task. In our computer simulations we can do different experiments and we firstly decided to simulate the immunization process based on the results we found about centrality measure of infected nodes. Specifically, at beginning of each time step we randomly move 7 nodes to the status 'recovery' according to their centrality score; those having the score above the average. We want to remark that the centrality scores are carried out with the static connections; on one side it saved a lot of computational resources and on the other side it's reasonable to select the subject to be vaccinated according to their stable relationships with other people (how many contacts s/he has in average).

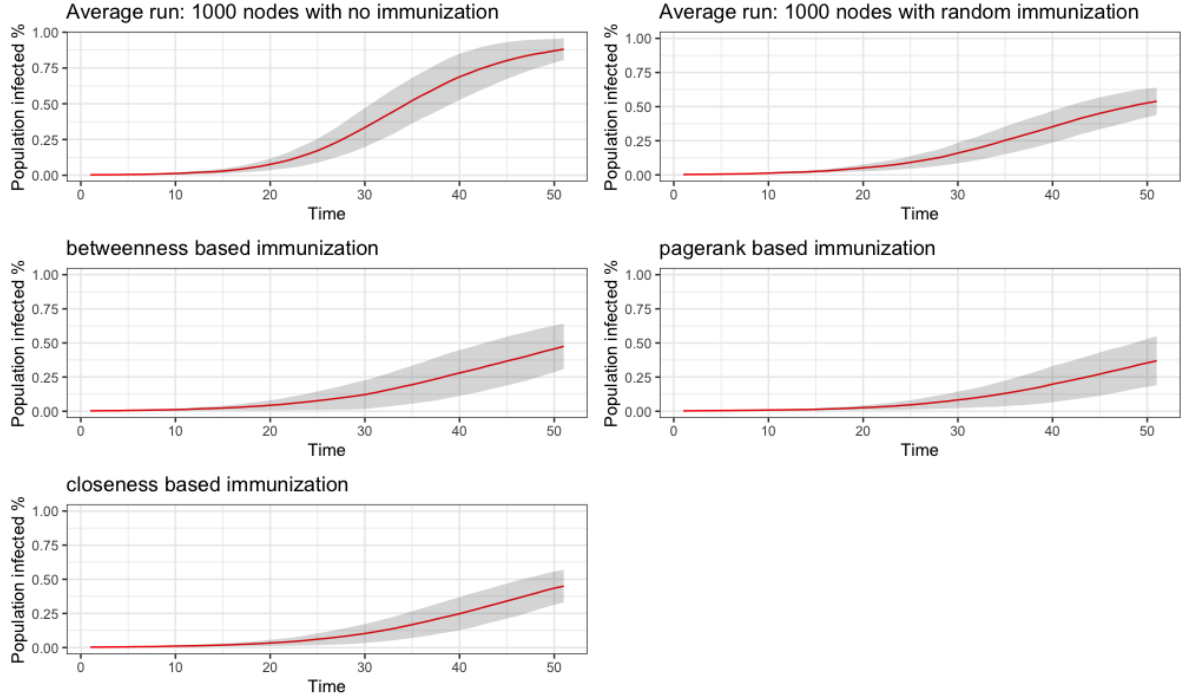


Figure 13: Immunization simulations' results. At each time stop we randomly move **7 nodes** to the status 'recovery' according to their centrality score.

We report the data drawn from the simulations in fig. 13. It's clear that all three methods are effective for mitigating the virus spreading and there is no big differences between them, as expected they worked better than random immunization. The pagerank based strategy seems to be the best since it has lowest percentage population infected and this is justified by the fig. 12 where a susceptible node's pagerank measure was quite different from the first nodes' pagerank score. Surprisingly, we discovered that closeness based method, not conforming to our initial beliefs, had no better outcomes than the betweenness based one. Lastly, the spreading process didn't stop completely; the virus was still in circulation since we had strict increasing curves. So, unfortunately, vaccinating was useful only to prevent total infection of the population and slow down the infection curves.

6.3 Attacks to infected nodes

The second strategy we proposed was, instead of removing a generic node having certain centrality score, removing only infected nodes. In particular, at each time step we changed the status of the top 7 nodes to 'recovery', ranked according to the centrality measures, and we did so only after the network had at least 70 infected nodes. Also here, for computational issues, we didn't consider the social interactions of the nodes.

In fig. 14 are shown the results. Unexpectedly, this method had worse outcomes with respect to the first one, only the closeness based approach had no improvements, it's counter-intuitive since directly 'attacking' the infected nodes appears to be the best way to stop the virus outbreak. Once again, centrality measure based strategies are more effective than random attacks. In this case it was also interesting to record the actual number infected nodes at each time step and we can see that only the closeness based method was able to break off the infection number's increasing trend.

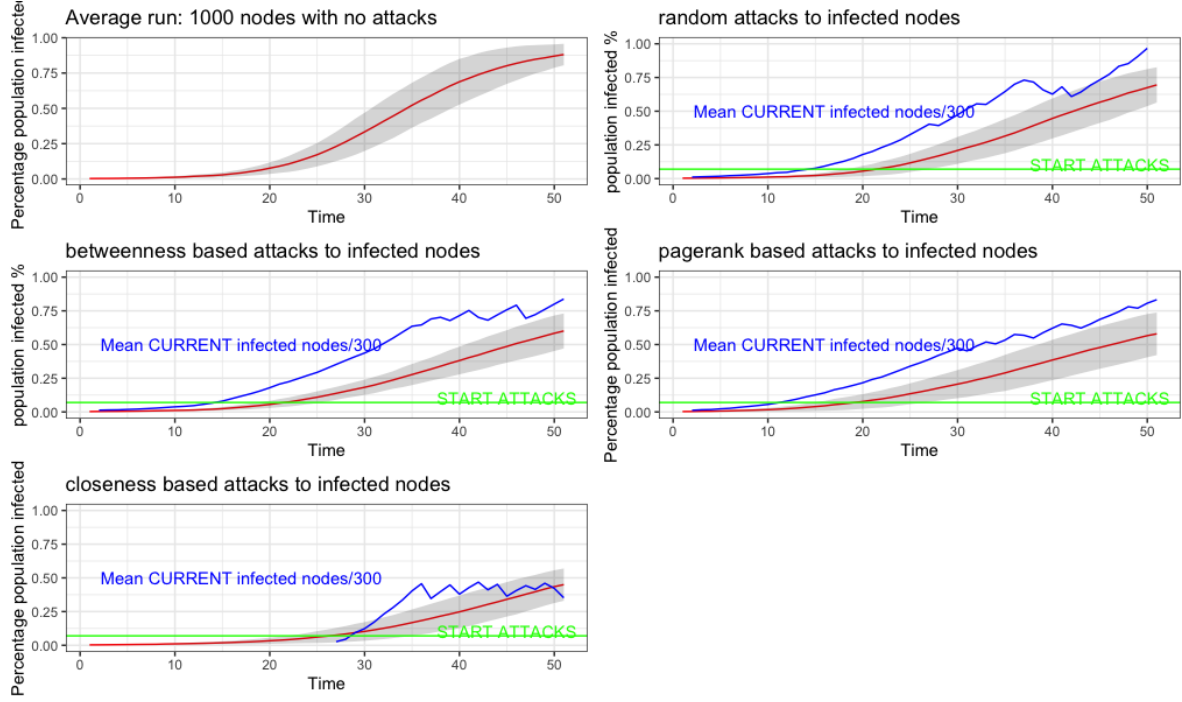


Figure 14: Results of the attack simulations. We started attack after the network had at least **70 nodes** (green line) and it consisted in changing the status of the **top 7 infected nodes** to 'recovery', ranked according to their centrality measures.

7 Conclusion

In this work we experimented with Temporal Gillespie's Algorithm and we found that the algorithm is a good simulation method since it returned satisfying results comparing with real data. We need investigate more on the parameters that we defined for the algorithm and the underlying network, maybe the more complex was necessary.

Finally, we studied centrality measures of the infected nodes in a non lockdown restriction context and using these informations we verified the network's robustness during the virus spreading adopting different strategies and we discovered that the spreading couldn't be completely stopped, so the network was robust.

References

- [1] Wikipedia contributors, “Epidemiology — Wikipedia, the free encyclopedia,” 2021.
- [2] C. L. Vestergaard and M. Géniois, “Temporal gillespie algorithm: Fast simulation of contagion processes on time-varying networks,” *PLOS Computational Biology*, vol. 11, pp. 1–28, 10 2015.
- [3] I. Z. Kiss, J. C. Miller, P. L. Simon, *et al.*, *Mathematics of epidemics on networks*. Springer, 2017.
- [4] D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
- [5] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science & Business Media, 2013.
- [6] P. Maheshwari and R. Albert, “Network model and analysis of the spread of covid-19 with social distancing,” *Applied network science*, vol. 5, no. 1, pp. 1–13, 2020.
- [7] Q.-H. Liu, M. Ajelli, A. Aleta, S. Merler, Y. Moreno, and A. Vespignani, “Measurability of the epidemic reproduction number in data-driven contact networks,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 50, pp. 12680–12685, 2018.
- [8] A. Barabasi and M. Posfai, *Network Science*. Cambridge University Press, 2016.