Modeling SARS-CoV-2 pandemic in NetLogo

Lorenzo Vainigli

lorenzo.vainigli@studio.unibo.it

matr. 0000842756

Course of Complex Systems and Network Science
Laurea Magistrale in Informatica
University of Bologna
A.Y. 2020-2021

Abstract. This is the paper's abstract . . .

1 Introduction

2 Previous work

In the literature there is some works that I used to plan this project. The concepts of model that I used derived from a book written by Brauer et al., that shows the models that can be used in a simulation of virus spreading in a population [1]. An article from Giordano et al. explain how to model the intervetions in the population to fight the COVID-19 outbreak [2], but it's too complex for the purposes of this project.

The Netlogo [3] models library offers a program that simulate the virus spreading in a population [4], it's very simple and general-purpose, but it was a good point to start the building of the model of this project.

3 Modeling

In this section I will explain the process of thinking which models are better to use for the aim of the project and how I modified the ones from [1].

3.1 Achievements

The final achievement of this project is to build and study a network model that simulates the outbreak of the new SARS-CoV-2 virus. To do that, we must consider a situation in which we have a crowd of people that have contacts with each other, where there are some infected individuals that cause the spreading of the infection. Also some kind of quarantine and isolations are needed for who become infected.

We need a model similar to the SEIQJR model. Let's start from the simplest model, the SIS, and then add new classes to make a more complex model.
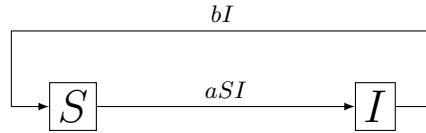
3.2 The SIS model

The SIS model is the simplest model in which there is a class of susceptible people (S) that, if one get the infection, goes to the infected class (I). Then, when infected people get recovered they returned to the susceptible class.

New classes

- S = Susceptible
- I = Infected

New variables

- a = infection rate;
- b = probability that an infected person get recovered.

Fig. 1. Diagram of the SIS model

$$\frac{dS}{dt} = -aSI + bI$$
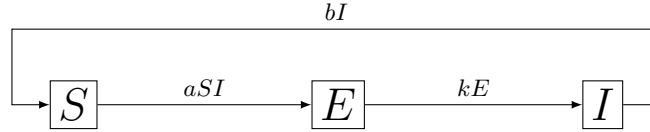
$$\frac{dI}{dt} = aSI - bI$$

3.3 The SEIS model

As we know, SARS-CoV-2 infection leads an incubation period during which the infected person don't have symptoms but can transmit the infection to other people. We introduce an exposed class E in which we put people that had have a contact with infected people and may be infected in an asymptomatic status.

New classes

 – E = Exposed

New variables

 – k = probability that exposed people become infective.

Fig. 2. Diagram of the SEIS model

$$\frac{dS}{dt} = -aSI + bI$$

$$\frac{dE}{dt} = aSI - kE$$

$$\frac{dI}{dt} = kE - bI$$

No immunity provided Beacuse of the unknown properties of immunity from SARS-CoV-2, we can assume that the S class and R class coincide, creating a cycle that causes the comeback of the recoverd individuals in the susceptible class.

3.4 The SEIQJS model

Due to the unavailability of a vaccine, to fight SARS-CoV-2 infected people are quarantined or isolated (or, also, hospitalized). As suggested from the SEIQJR model, two classes were added: Q is the class of quarantined people, i.e. the exposed people belong to the class E; J is the class for isolate infected people of class I. So basically Q is for E what J is for I, with the only difference that there is a flow from Q to J but not vice versa.

People isolated (i.e. belong to class J) can recover as people belong to class I.

New classes

- $Q = $ Quarantined
- $J = $ Isolated

New variables

- $\varepsilon_E = $ probability that a person exposed (in E) transmit the infection, exposed-transmission-factor;
- $\varepsilon_Q = $ probability that a person quarantined (in Q) make a contact, quarantined-contact-factor;
- $\varepsilon_J = $ probability that a person isolated (in J) transmit the infection, isolation-transmission-factor;
- $k_E = $ probability that exposed people become infective;
- $k_Q = $ probability that quarantined people are isolated;
- $c_Q = $ percentage of exposed people that are quarantined at each time step;
- $c_J = $ percentage of infected people that are isolated at each time step;
- $b_I = $ probability that an infected person get recovered;
- $b_J = $ probability that an isolated person get recovered;
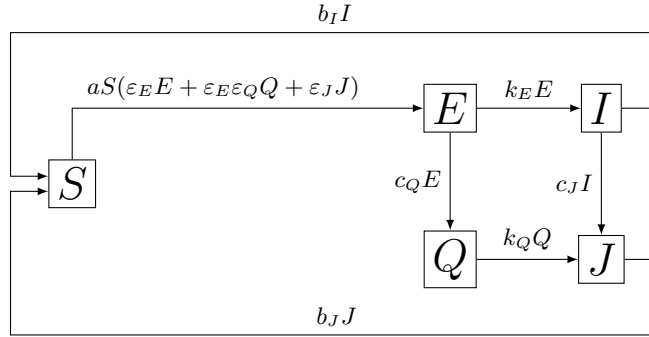
Fig. 3. Diagram of the SEIQJS model

$$\frac{dS}{dt} = -aS(\varepsilon_E E + \varepsilon_E \varepsilon_Q Q + \varepsilon_J J) + b_I I + b_J J$$

$$\frac{dE}{dt} = aS(\varepsilon_E E + \varepsilon_E \varepsilon_Q Q + \varepsilon_J J) - (k_E + c_Q)E$$

$$\frac{dQ}{dt} = c_Q E - k_Q Q$$

$$\frac{dI}{dt} = k_E E - (b_I + c_J)I$$

$$\frac{dJ}{dt} = k_Q Q + c_J I - b_J J$$

3.5   Action to limit the epidemic spread

In a context of epidemic spread some actions can carried on to protect the health of the population. Basically, because of the virus infection is transmitted during contacts person-to-person, these contacts must be reduced. Putting the exposed or infected people in quarantine or isolation is the best thing to do, but also methods of social distancing or mobility restrictions can be used.
In this project I implemented quarantine, isolation, social distance and lockdown.

3.6   Contact tracing and evaluation method

As we know that SARS-CoV-2 infection leads to an asymptomatic status, a contact tracing technique can be used to find people who can transmit the virus before that they become symptomatic. In this model I built a contact tracing technique not properly to find potential infected individuals, but to evaluate the amount of contacts in each simulation to have an indicator of the power of the spreading. What is important to notice is that the more contacts we have between people, the more powerfull the outbreak will be and thus the harder will be to control it.

4   Implementation

In this section I describe the way I implemented the concepts shown in the previous section.

4.1   The base model

The Virus model from the NetLogo library [4] is the point from which I started the development of my model. The Virus model is an implementation of the SIR model, with some peculiarities:

1. people can reproduce;
2. people can recover from the infection but they can also die;
3. recovered individuals have a phase of immunity before they come back to the susceptible class.

I conclude that these three characteristics have no interest to be in my final model, so I edited this model to have the following situation:

1. people cannot reproduce because the epidemic spread is as fast as we can ignore the machanism of reproduction in the population;
2. people can only recover from the infection and they cannot die;
3. recovered individuals have no phase of immunity beacuse, for SARS-CoV-2, we don't have clear informations.

Once I have a SIR model is easy to transform it to a SIS model: the only thing to do is put the flow that previously went to the recovered class (R) in the susceptible class (S).
After this editing I basically my final model will be an enrichment of this model.

4.2   Parameters

The parameters of the model are the ones described for the SEIQJS model, divided into fixed parameters, i.e. the ones that are specific to the virus, and the control parameters, which can be tuned in order to study attempts to manage the epidemic.

Fixed parameters

– `infection-rate` (a);
– `exposed-transmission-factor` (ε_E);
– `exposed-to-infected-rate` (k_E);
– `quarantined-to-isolated-rate` (k_Q);
– `infected-chance-recover` (b_I);
– `isolated-chance-recover` (b_J).

Control parameters

– `quarantine-perfection-rate` (ε_Q);
– `quarantine-perfection-rate` (ε_J);
– `quarantines-per-tick-rate` (c_Q);
– `isolations-per-tick-rate` (c_J).

Experiments with this model are made by trying different values for control parameters, while fixed parameters remain unchanged.

4.3 Class representation

The SIS, SEIS and other similar diagrams have a fundamental constraint: at each time step, a person of the population (i.e. an agent) must belong to one and only one class. This is done by simply add five properties to the agent: one of each indicates a different class, so at each time step, one of these property must be true and the others false.

4.4 Class transitions methods

To make possible that a person change class a series of methods were implemented. All of them share a similar structure:

1. check if the candidate that is going to go to the class actually belongs to a class from which a flow from the two classes is defined (throw an error if not);
2. put to *false* the properties of the person that could hold the value *true*;
3. put to *true* the property that indicates the new class.

These methods are `get-exposed`, `get-quarantined`, `get-infected`, `get-isolated` and `get-recovered`.

4.5 Core procedures: setup and go

Every NetLogo model is based on these two procedures:

- `setup` is the function that provides the initialization of the model before every simulation;
- `go` is the function that will be executed every time step (tick).

The procedure `setup` creates the turtles that represent the population and initialize their properties: every turtle is putted in the susceptible class. To create the beginning of virus spreading, after these actions some of the created turtles are picked at random and moved to the exposed class and to the infected class.

To understand the explaination of the following procedure `go` I need to underline first a design choice: I decided to let that some

actions occurs more often that others, below I explain which they are an why. To implement this behaviour I used a new variable that counts the days elapsed from the beginning of the simulation, every time `ticks` is a multiplier of `ticks-per-days`, days counter advances.

The procedure `go` calls various other procedures that carried on the simulation and the behaviour of the model, but they are divided into two groups:

– *once-per-tick* procedures that are called every time the `go` procedure is called, so every tick
  • people motion in the enviroment,
  • people transition to exposed class due to a contact with an infected person,
  • contract tracing update;
– *once-per-day* procedures that are called every time the days counter advanced:
  • every class transition except the one from susceptible to exposed, that, as I said previously, happens for every tick.

4.6 Virus propagation

The infection spreading of the virus in the population is implemented in the function `infect` that looks a little bit complicated but, essentialy, uses the parameters of the model call, in a proper way, the procedure `get_exposed`. It's the only procedure that makes call to `get_exposed`. Take a look to the comments in the code of the model to know more.

4.7 Contact tracing

Contact tracing is implemented using the construct `link` of NetLogo. Ar each time step, the program asks to each person to create a link with other people that share, in that moment, the same patch. Basically is the same way that the virus spreads and it has a sense beacuse the aim of contact tracing is to track the propagation of the virus. The result is a graph connecting people to each others.

4.8 Social distance and lockdown

Two methods were implemented to control people movement in order to reduce contacts between them: social distance and lockdown

Social distance can be enabled or disabled in this model and uses an auxiliry parameter called `social-distance-perfection-rate`. If social distance is disabled, people are free to move everywhere, while if it's enabled people avoid to move to a patch that is already occupied by others with an error probability of determined by $1 - $ `social-distance-perfection-rate`.

Lockdown is implemented taking into account a value that defines the probability that a person will move or not. If the variable `lockdown-strinctness` is set to zero, people are completely free to move, while if the value increases, people will make a move with less probability. If a person doesn't move, it remain in the same patch where it was previously.

Take a look to the procedure `move` to find the code about these control measures.

4.9 Evaluation

In this model I consider to evaluate every situation analyzing the graph generate by the contact tracing method. The more connected is the graph, the more powerful is the virus spreading and so, in a real situation, the more stressed the health system will be.
I use the definition of *degree centrality* to have a clear indicator of the connections that each person create during simulation. Beacause of degree centrality is a node-based measure, I used the average of all the degree centrality to get a value to evaluate the entire graph. An auxiliary indicator of standard deviation for this average is provided. If a control action on the population reduces the value of *average degree centrality*, it means that the action is effective.
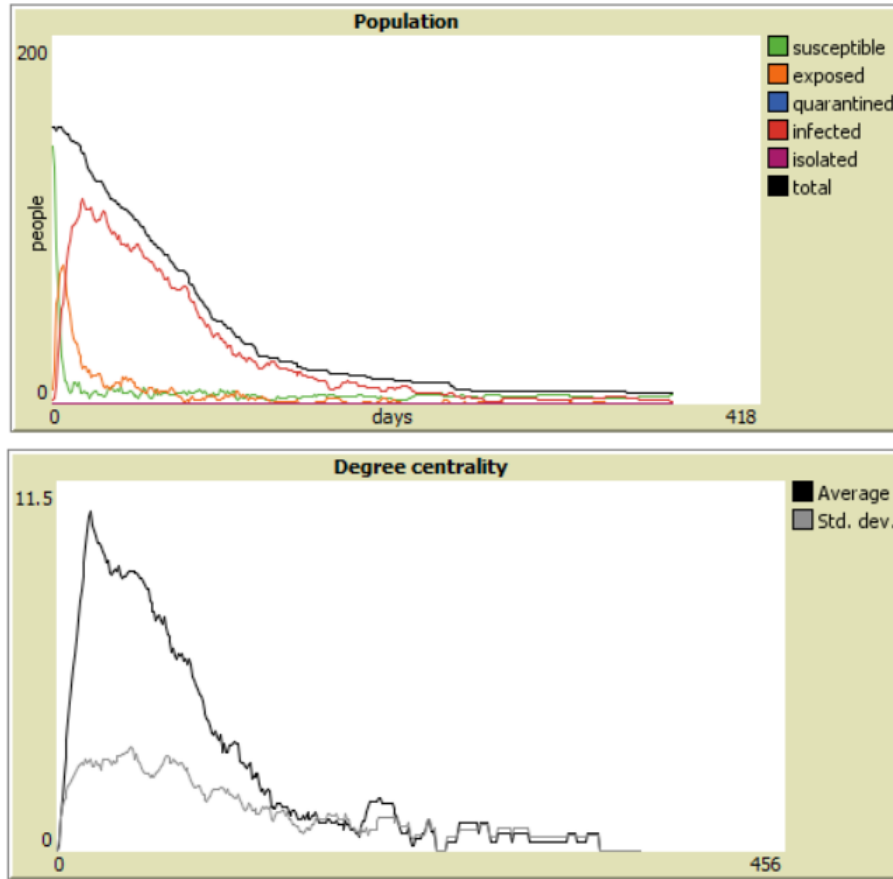
5 Results

In this section I describe the results obtained with simulations on this model. Differences between simulations are about managing control

10

parameters in order to watch how the simulation evolves with different actions.

The size of the population at the beginning of the simulation is fixed at 150 and the duration of the simulation is 365 days.

5.1 Virus diffusion without actions

Fig. 4. Result of the simulations without actions to control the diffusion of the virus. Population has been reduced by **96%** and **147** people were dead.

5.2 Using only quarantine and isolation

Simulation is carried out with these settings:

- quarantine-perfection-rate $= 70$
- quarantines-per-day-rate $= 10$
- isolation-perfection-rate $= 90$
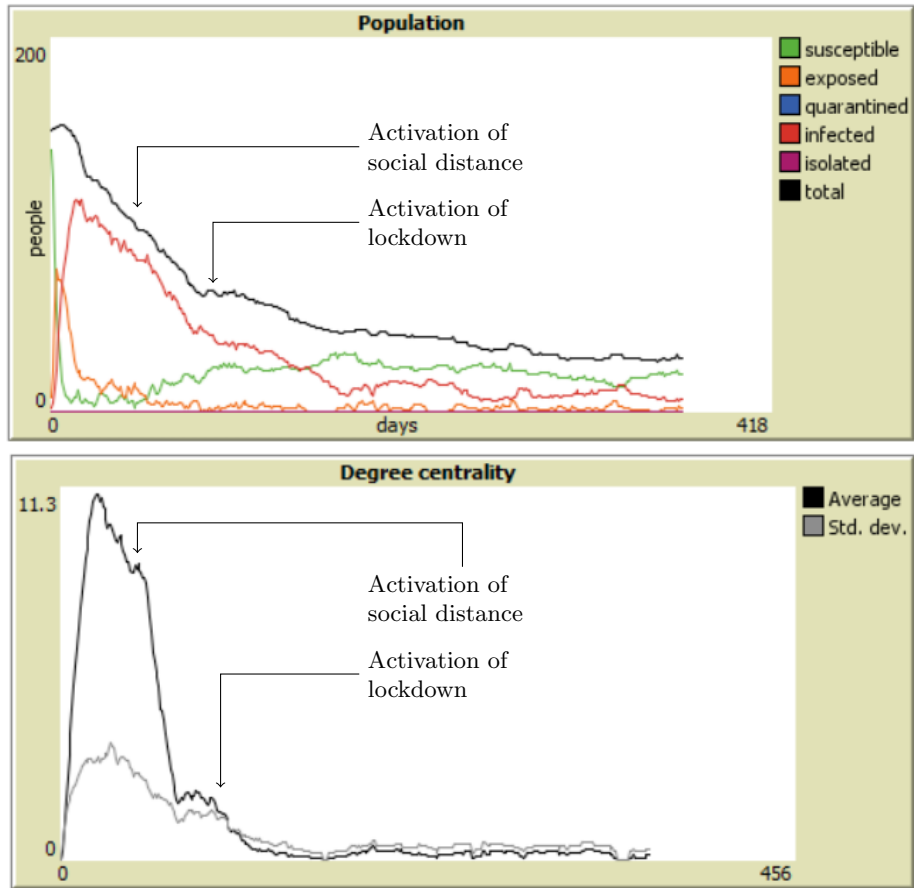- isolations-per-day-rate $= 10$
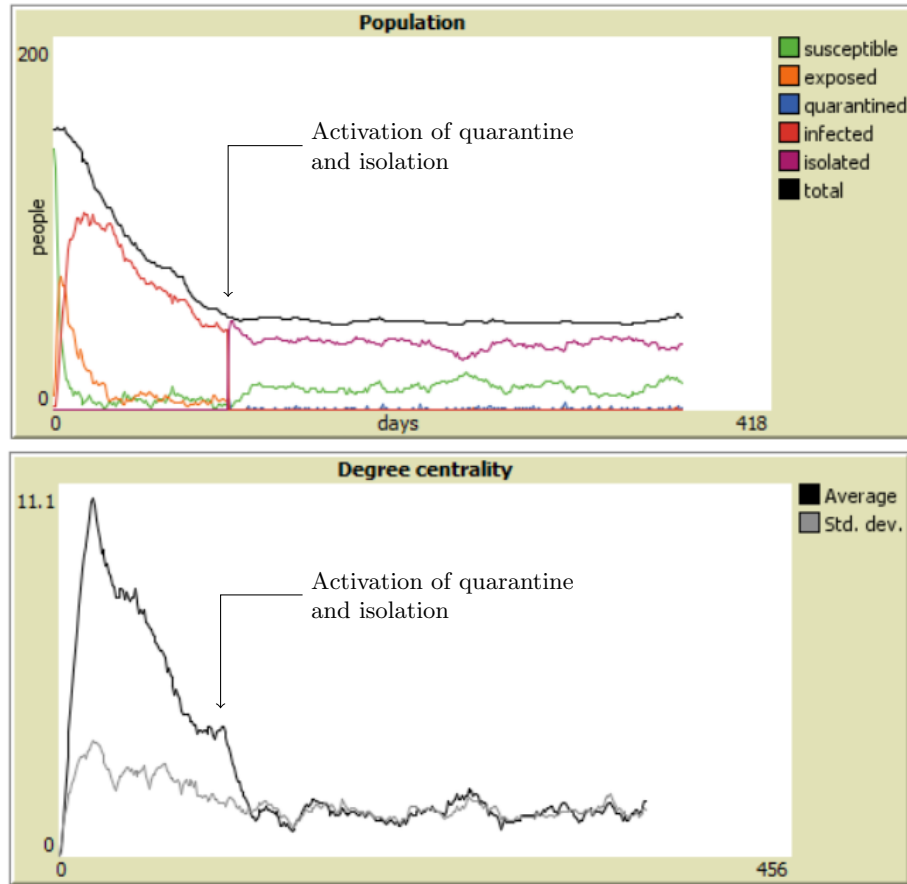
Fig. 5. Result of the simulations where on the 100th day quarantine and isolation has been activated for exposed and infected. Population has been reduced by **66.7%** and **122** people were dead.

5.3 Using only social distance and lockdown

Simulation is carried out with these settings:

- social-distance-perfection-rate $= 70$

– `lockdown-strictness` $= 90$

Fig. 6. Result of the simulations where on the 50th day social distance has been activated for exposed and infected and on the 100th day lockdown has been activated. Population has been reduced by **80.7%** and **163** people were dead.

5.4 Combining quarantine and isolation with social distance and lockdown
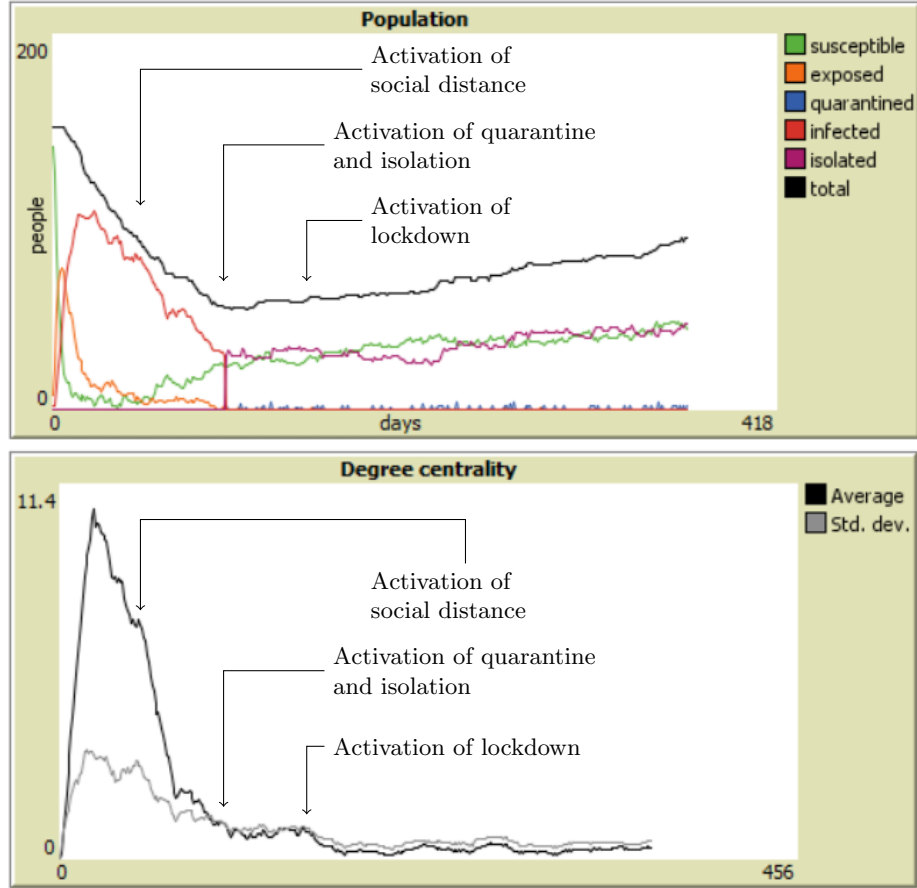
Fig. 7. Result of the simulations where on the 50th day social distance has been activated, on the 100th day quarantine and isolations for exposed and infected has been activated and on the 150th day lockdown has been activated. Population has been reduced by **39.3%** and **116** people were dead.

6 Conclusions

We worked hard, and achieved very little.

14

References

[1] Fred Brauer, Carlos Castillo-Chavez, and Zhilan Feng. *Mathematical Models in Epidemiology*. Springer New York, 2019. URL: https://www.springer.com/gp/book/9781493998265.

[2] Giulia Giordano et al. "Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy". In: *Nature Medicine* 26 (June 2020), pp. 1–6. DOI: 10.1038/s41591-020-0883-7.

[3] U. Wilensky. *NetLogo*. 1998. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. URL: https://ccl.northwestern.edu/netlogo/.

[4] U. Wilensky. *NetLogo Virus model*. 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. URL: https://ccl.northwestern.edu/netlogo/models/Virus.