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. The SARS-CoV-2 outbreak has forced the world population to face a new disease for which there was not a vaccine and governments had to use some alternative actions to limit the virus spreading. Predicting the impact of an action on the population is very important before making it effective and we need mathematical models to do that. This paper shows the implementation of a virus outbreak in NetLogo and it provides actions to limit the spreading: quarantine, isolation, social distance and lockdown. Every action can be enable or disable and it has parameters that can be set. A study on how these actions can help to fight and slow down the virus propagation is described and results that prove the effectiveness of these containment measures are shown.

1 Introduction

This paper describes the process of building and studying a network model that simulates the outbreak of the new SARS-CoV-2 virus using NetLogo. In addition, the model provides methods to control the spreading. To achieve this challenge, we must consider a situation in which we have a crowd of people, each of which have contacts with others and 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. Finally we need some population-size actions like social distancing and lockdown, beacause these are the most used and effective actions that have been used by governments to reduce the impact of the virus on the population.
Our model provides all the features mentioned above with different paramaters that can be changed in order to set the strenghtness of the containment measures.
Once we have found a setting of parameters that causes, more or less, the same effects on the population as the real virus did, we focus our experiments in studying the effects of the containment measures on the virus impact on the population. Results we got clearly show that these actions are effective and can save lives.

2 Preliminaries

The literature about virus diffusion is very rich because 2020 has been a year in which the researchers need to discover everything

about the new SARS-CoV-2 virus and the COVID-19 disease [1]. Models to represent situations like this or other diseases like flu were developed by Brauer et al. [2]; one of these, the *SEIQRJ* model, is suitable to study the SARS-CoV-2 pandemic. An article from Giordano et al. explains how to model the intervetions in the population to fight the COVID-19 outbreak [3], but it's too complex for the purposes of this project. The Netlogo [4] models library offers an implementation of the *SIR* model [5]: it's very simple and it's a good point to start the implementation of our model.

In this project, the modelling phase started from the *SEIQJR* model shown by by Brauer et al. and the implementation phase started from the Virus model from the NetLogo library.

3   Modeling

In this section we will explain the process of building the model that will be implemented to realize this project.

We start from the scheme of the *SEIQJS* model, explaining why this model is the one we need for our purpose and then we conclude this section explaining how to model contact tracing, social-distance and lockdown.

3.1   The SEIQJSD model

Before showing the diagram of the model that we will use, let's see how it was constructed.

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. 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. 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. Finally, a class D is added where dead people go.

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, if they don't die.

Classes

- S = Susceptible
- E = Exposed
- I = Infected
- Q = Quarantined
- J = Isolated
- D = Dead

Variables

- a = infection rate;
- ε_E = probability that a person exposed (in E) transmit the infection, exposed-transmission-factor;
- ε_Q = probability that a person quarantined (in Q) make a contact, quarantined-contact-factor;
- ε_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 left the infected class;
- b_J = probability that an isolated person left the isolated class;
- r_I = probability that an infected person get recovered, otherwise he die;
- r_J = probability that an isolated person get recovered, otherwise he die.

The diagram of the model is shown in fig. 1 and the equations are shown in eq. 1.
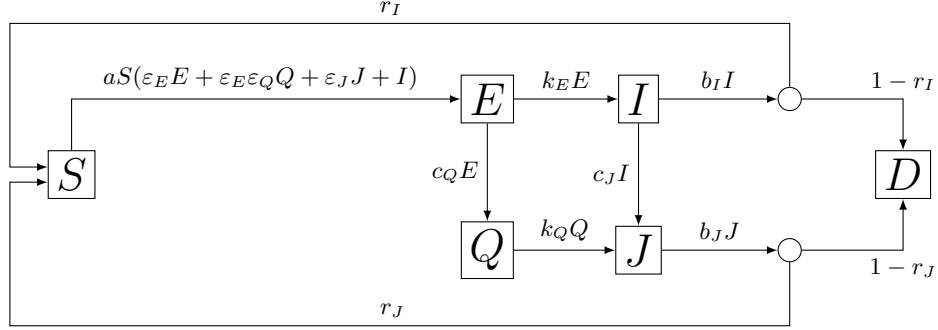
Fig. 1: Diagram of the *SEIQJSD* model.

$$
\begin{aligned}
\frac{dS}{dt} &= -aS(\varepsilon_E E + \varepsilon_E \varepsilon_Q Q + \varepsilon_J J + I) + b_I r_I I + b_J r_J J \\
\frac{dE}{dt} &= aS(\varepsilon_E E + \varepsilon_E \varepsilon_Q Q + \varepsilon_J J + I) - (k_E + c_Q)E \\
\frac{dQ}{dt} &= c_Q E - k_Q Q \\
\frac{dI}{dt} &= k_E E - (b_I + c_J)I \\
\frac{dJ}{dt} &= k_Q Q + c_J I - b_J J \\
\frac{dD}{dt} &= b_I(1 - r_I)I + b_J(1 - r_J)J
\end{aligned}
\tag{1}
$$

3.2 Action to limit the epidemic spread

In a context of epidemic spread some actions can carried out 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 we implemented quarantine, isolation, social distance and lockdown.

3.3 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 we 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 virus 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 we describe the way we implemented the concepts shown in the previous section.

4.1 The base model

The Virus model from the NetLogo library [5] is the point from which we started the development of our 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.

As previously stated, we conclude that immunity is not a characteristic that we want to have in our final model, so we remove the class of immune people, but we keep the behavior of reproduction and death.

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 that we cannot change, 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-leave-rate` (b_I);
- `isolated-leave-rate` (b_J);
- `infected-chance-recover` (r_I);
- `isolated-chance-recover` (r_J).

Control parameters

- `quarantine-perfection` (ε_Q);
- `isolation-perfection` (ε_J);
- `quarantines-per-day-rate` (c_Q);
- `isolations-per-day-rate` (c_J);
- `social-distance-perfection`;
- `lockdown-strictness`.

The last two parameters, even if they are not part of the *SEIQJS* model, are used to manage the epidemic.
Experiments with this model are made by trying different values for control parameters, while fixed parameters remain unchanged.

4.3 Class representation

The *SEIS*, *SEIQJR* 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 properities 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`. This fact does not apply to the class of dead people, beacuse it is a class without exit: if a person dies, the agent is killed and the deadth counter increased by one.

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 (If p want to go from A to B a transition $A \to B$ must be defined). Throw an error if is not possibile;
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` we need to underline first a design choice: we decided to let that some actions occurs more often that others, below I explain which they are and why. To implement this behaviour we 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 by one.
Why this design choice? Every tick people make a single move, so it's strange that the mechanism of quarantine and isolate people occurs with the same frequency. Distinction between `ticks` and `days` makes the simulation more realistic.

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 day counter advanced:
  - every class transition except the one from susceptible to exposed, that, as we stated 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 to 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. At 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`. 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`.

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 we considered to evaluate every situation analyzing the graph generated 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.

We use the definition of *degree centrality* to have a clear indicator of the connections that each person creates during simulation. Beacause of degree centrality is a node-based measure, we used the average of all the degree centralities 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.
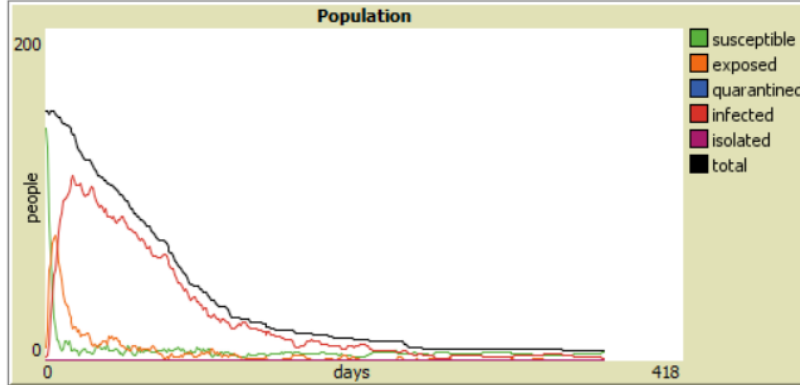
5 Results

In this section we describe the results obtained making simulations with this model. Differences between simulations are about watch how the simulation evolves with different actions.

The size of the population at the beginning of the simulations is fixed at 150 and the duration of the simulations is 365 days.
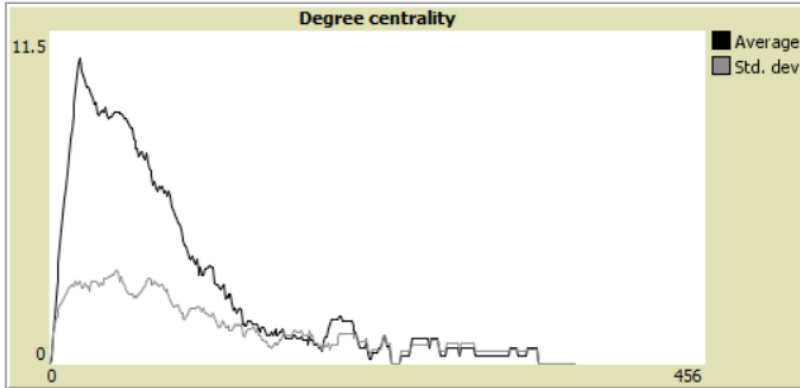
5.1 Virus diffusion without actions

The first thing to do to have an idea of what happens is to watch the behavior of the model if nothing action to fight the virus is taken. As we can see in fig. 2a the lethality of the virus makes an unacceptable impact on the population, reducing it to few individuals. The

reduction of the *average degree centrality* (fig. 2b) is only due to the decrease of the population.

(a) Evolution of the population divided by class.

(b) Evolution of the values of degree centrality.

Fig. 2: 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

Let's see how using quarantine and isolation will change the virus diffusion and thus the results after a simulation. These measure prevent the people to move but not to making contacts, even if with a

11

lower probability. We can see in fig. 3a the huge impact that these measures make in the simulation, saving the 29,3% of the population with respect to the previous situation. In fig. 2b we can se a visible reduction of contacts between people. Simulation is carried out with these settings:

– `quarantine-perfection` = 70
– `quarantines-per-day-rate` = 10
– `isolation-perfection` = 90
– `isolations-per-day-rate` = 10


5.3 Using only social distance and lockdown

Before combining all the control measures available, let's see the impact of social distance and lockdown. In the graph of population in fig. 4a the effect is not particularly noticeable, but as we can see in fig. 4b there is an immediate drop of the number of contacts when social distance is activated. A similar thing happend with the activation of lockdown. Simulation is carried out with these settings:

– `social-distance-perfection` = 80
– `lockdown-strictness` = 90

Result shows that if there is the need to choose to use quarantine and isolation or social distance and lockdown, the first option is the best beacause is able to save more lives.

5.4 Combining quarantine and isolation with social distance and lockdown

Finally, let's see what happens if all these measures are activated progressively. Fig. 5 shows the final results: after the year elapsed in the simulation, the population number is higher and the number of deaths is lower with respect to the previous experiments. We can notice in this case something that didn't happens before: the population is able to come back to grow.
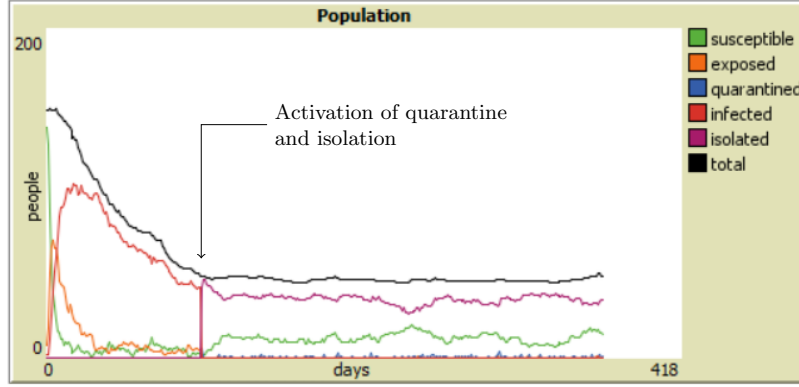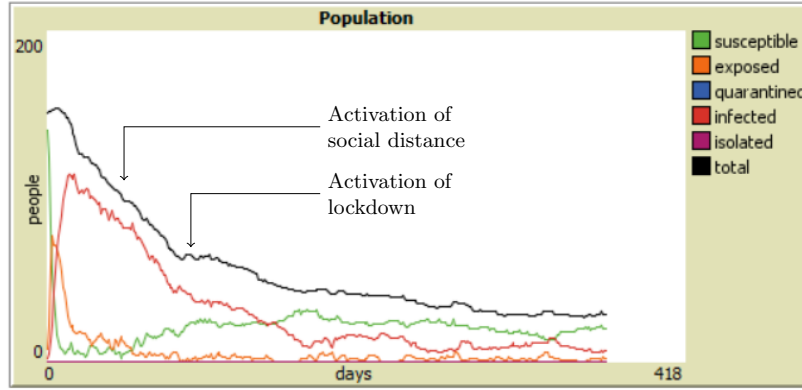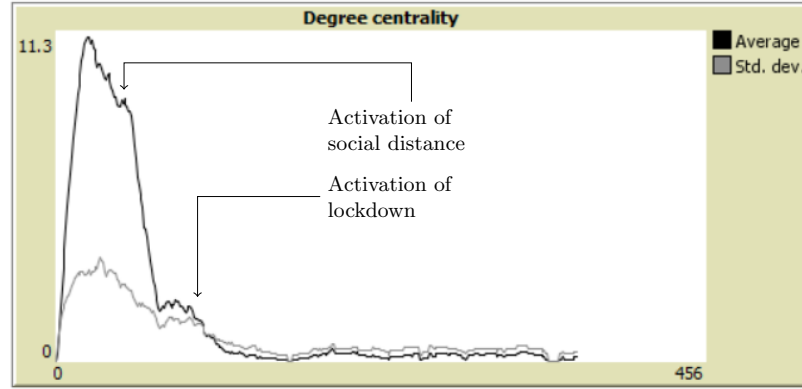
(a) Evolution of the population divided by class.
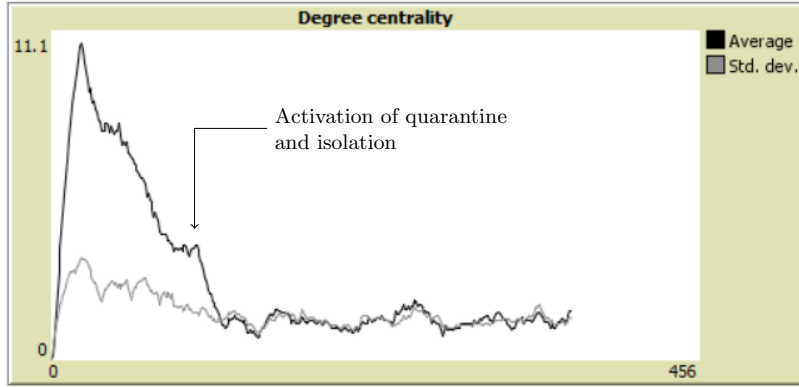
(b) Evolution of the population divided by class.

Fig. 3: 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.
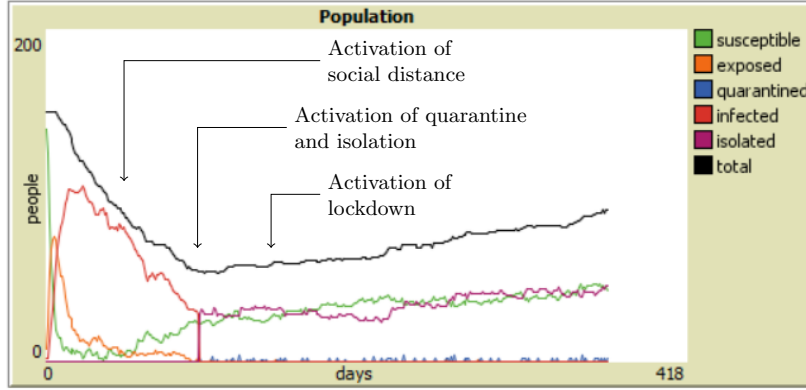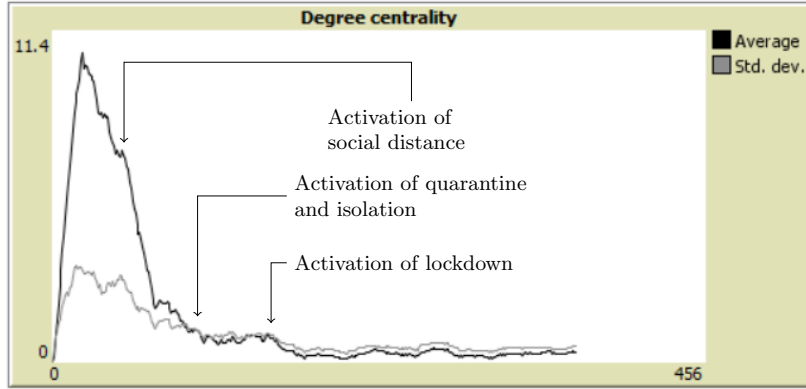
(a) Evolution of the population divided by class.

(b) Evolution of the population divided by class.

Fig. 4: Result of the simulations where on the 50$^{\text{th}}$ day social distance has been activated for exposed and infected and on the 100$^{\text{th}}$ day lockdown has been activated. Population has been reduced by **80.7%** and **163** people were dead.

(a) Evolution of the population divided by class.

(b) Evolution of the population divided by class.

Fig. 5: 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.

15

6 Conclusions

The model shown in this paper represents a pretty accurate simulation to reproduce the SARS-CoV-2 outbreak and results shows that the actions that were carried out in the real world to fight this virus are effective also in this model. Each of the possibile action that can be used (quarantine, isolation, social distance, lockdown) bring benefits to the final results and, if all of them are used toghether, the results are even better.

Further directions

Two interesting things can be done to carry on this work. The first if to try different setting of the parameters to watch how a change in a parameter value impacts on the behavior of the model; the second is related to the first and is about to find possible tipping points, i.e. when a little change in a parameter caused a big change in the simulation.

References

[1] Tanu Singhal. "A Review of Coronavirus Disease-2019 (COVID-19)". In: *Indian journal of pediatrics* 87 (Apr. 2020), pp. 281–286. DOI: 10.1007/s12098-020-03263-6.

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

[3] 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.

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

[5] 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.