

EpiDEMExtended report

Introduction

EpiDEMExtended is a model created to simulate the spread of a pandemic in a population at the variation of different parameters. The model is implemented in NetLogo¹ and is an extension of epiDEM Basic², one of the library models that come pre-installed with NetLogo. EpiDEM Basic simulates an SIR (susceptible-infectious-recovered) epidemiologic model (once someone has recovered, he/she is completely immune from the disease) in which individuals move around randomly and infect each other when close enough. This base model allows to vary the size of the population, the probability of getting infected when close enough to an infected individual, the average time needed for recovery, the probability of recovering after that time (at each time unit following the end of the recovery time, the model keeps extracting a random number and checking whether it satisfies the probability. Only when it does the person has actually recovered). Regarding the output the model produces, three plots are generated: the cumulative infected and recovered (cumulative since it keeps track of everyone that ever got infected, even if now that someone has recovered), the number of infected and not infected in the population, the infection and recovery rates. It is also calculated an estimate of the basic reproduction number, R_0 , which is the number of people directly infected by a single individual inside a fully susceptible population. This number is rather important since if $R_0 > 1$ then the disease tends to become a pandemic, if $R_0 < 1$ then the disease tends to disappear.

EpiDEMExtended starts from here, adding various elements to the model and trying to simulate the spread of a specific recent disease, Covid-19, using real Italian data. In the next sections of this document these elements will be described, first in a high level, global way, and then more specifically regarding the implementation. However, in order to have a first-glance understanding of the work done, here are summed up the enhancements apported to the model:

- The population has been divided into age classes with different susceptibilities to the disease (that the user can modify if he wishes)
- The population has been divided into families (depending on user-defined parameters) and to each has been assigned a house, houses also being new additions to the model
- The population now does not move at random but follows a (user-defined) cycle to visit various new elements in the simulation, called activities
- The susceptibility to the disease now also depends on the current activity
- The disease transmission can now also be environmental, not only due to individuals closeness
- The possibility to enforce a quarantine (with different gravity levels) has been added, allowing to modify individuals behaviours
- To check the results of these additions, various new outputs have been inserted



High level model

EpiDEMExtended presents different types of agents who partake in the simulation:

- Activities -> immobile agents, comprising all various activities that individuals may do during the day, both dutiful (jobs, going to school) and recreational
- Houses -> immobile agents, each representing the home of a certain family
- People -> the individuals, (mostly) moving agents divided into age classes and grouped into families

Activities

Activities are furtherly divided in 4 sub-groups:

1. Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL

2. Yang, C. and Wilensky, U. (2011). NetLogo epiDEM Basic model. <http://ccl.northwestern.edu/netlogo/models/epiDEMBasic>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL

- Leisure activities -> all the recreational activities people will do after their duties. This includes parks, cinemas, restaurants etc. Note that each of these activities will also be a workplace for some people
- Education activities -> any teaching activity falls here. Schools and universities are part of this category. Again, note that for some people these activities will be their workplace
- Health activities -> hospitals, clinics, chemists' and such are part of this category
- Professional activities -> all other work-related activities that are not part of the other groups (banks, post offices, factories etc.)

Activities are defined in files, specifying their description (such as "restaurant", referred to as kind), its productive value (economical importance of the activity) and its capability of smart working (how possible it would be to work from home).

Individuals - structure

Regarding individuals, they are separated into 7 age classes, each of them presenting a different behaviour and susceptibility to the virus. The considered age classes are the following:

- 0-4 -> modelled to remain constantly home (may still get infected through the other family members)
- 5-14 -> attend the type of education activity in the simulation denoted as "primary school"
- 15-19 -> attend the type of education-activity in the simulation denoted as "secondary school"
- 20-24 -> attend the type of education activity in the simulation denoted as "university"
- 25-39 -> young workers, to each of them is assigned as a job one of the activities
- 40-64 -> old workers, to each of them is assigned as a job one of the activities
- 65 and older -> elders, who do not work but only enjoy recreational activities

The percentage of each class can be specified from file.

As already stated, people are grouped into families. By family is meant a group of people (of appropriate age classes) living together in the same house. The grouping in families and subsequent house assignment is done following some rules. First of all, families have at most 4 people, of which at most 2 are sons (these two rules may not always hold, [more details](#) later in the implementation description). People aged 0-4, 5-14, 15-19 always live with two parents (0-4 with 25-39 parents, the rest with 40-64 parents), while 20-24 people may live with parents 40-64 or with a roommate (another 20-24) or alone. The remaining age classes may live with another person of the same age class or alone. All these possibilities can be made more or less probable defining from file percentages for them to happen, along with the probability for any individual of one of the "son" age classes to have a sibling in the same age class.

Individuals - behaviour

Now that a description of the parts of the model has been addressed, let us move to the behaviour of agents during the simulation. As already told, the only moving agents are people. The individuals (except 0-4 people, who, again, just stay home) move depending on a cycle appropriate for the age class they belong to, cycle that can be specified from file (being cycles completely configurable, the responsibility for them being actually appropriate is left to the user). These cycles are made of activities that people will follow for a certain duration (defined as well from file). It is also possible to introduce variability of the activity performed for a certain time slice ([more details](#) in the implementation part). The defined cycles won't always be followed though: this due to the presence of a variable, modifiable at runtime, that simulates a quarantine with different levels of gravity that, consequently, may keep an increasing number of activities closed, forcing individuals who wanted to do them to instead stay home. This might not always happen: if the user decided to allow illegal behaviours, an individual may, with a probability of 5%, decide not to

follow the law and move towards a random leisure activity (allowing illegal behaviours will have some side effects that will be described [later](#) in the implementation part).

Infection – direct disease transmission

Now, regarding the disease transmission, individuals can get infected due to closeness with a sick person or through “environmental” infection (if activated by the user). Going in order, the infection due to closeness happens with a certain probability that differs for each age class and situation the person is in. The susceptibility (probability of getting infected) of a person of a certain age class in a certain situation is obtained by the mean number of contacts that age class experiences in that situation divided by the total mean number of contacts for that age class (data regarding the contacts has been obtained from a Covid-19 related [study](#) conducted by the Italian Comitato tecnico scientifico). The different situations considered are being at home, being at school, being at work, being at leisure and moving to any target (= being outside). Each of them has a respective susceptibility in the proper age classes (for example, a student does not need a work susceptibility).

Infection – environmental disease transmission

Moving to the other way of getting infected, by “environmental” infection is meant getting infected due to moving where an infected person has recently been, coming in contact with infected particles he left behind (by breathing, sneezing etc.). This happens only if the user wishes so, by activating a switch at any time before or during the simulation. There are two other related parameters that can be set: the infected particles decay time, since after a while those will evaporate, making passing through that particular point safe again, and the base environmental infection chance, the probability for a susceptible individual to become infected getting in contact with the infected particles. Note that this last probability is a base one, since it will be scaled down depending on how long the infected particles have been around (the “infectious strength” of the particles diminishes with time because of their evaporation).

Outputs

About the outputs, some were added: five counters for the number of infected people in each situation (while at home, while at school, while at work, while at leisure, while moving, plus a sixth total number of infected people counter and a seventh number of currently infected people counter) and an estimate of the global productivity, that changes as the quarantine level does, due to some activities being kept close (the percentage of closed activities is also shown).

Lastly, the graphical simulation employs different colours and shapes to distinguish the various agents:

- Circles are activities, that can be orange if leisure activities, yellow if education activities, cyan if health activities, blue if professional activities, while grey if activities currently closed due to quarantine
- Brown (slightly transparent) house-shaped agents are houses
- Person-shaped agents are white if susceptible, red if infected, green if recovered
- Patches (each of the squares composing the simulation grid) are coloured with shades of violet if environmentally infected (brighter = more infectious, darker = less infectious), otherwise remain black

Now that the model has been described at high level, let us move deeper into some implementational aspects.

[Activities](#)

General implementation

Implementationally speaking, each type of activity is a separate breed. These breeds all currently present the already mentioned attributes (kind, production-value, smart-working-capability). This is a bit redundant

(a single activity breed could have been employed), but will allow, if necessary, to specify a particular attribute only for a certain type of activity (for example it may be of interest in the future to keep track of the daily income of certain activities and it would not make much sense for education activities to have such a counter).

As stated before, activities are read from file. In particular, there is a file per type/breed of activity (“leisure-activities.txt”, “education-activities.txt”, “health-activities.txt”, “professional-activities.txt”) in which is specified an activity per row. To perform the reading, each file is opened (and closed) two times: the first one to count the number of activities and the second to do the actual reading. This could be avoided by simply specifying at the top of the file the number of activities present, but being all this still at setup time and thus not influencing the speed of the simulation at runtime (and also not being too computationally heavy, at least for the current simulation size), it has been decided against doing so. Once read, activities are created and initialized, setting their attributes, proper colour and location. It is not allowed for two activities to be on the same patch.

Quarantine effect

Activities may suffer the establishment of the quarantine law and some of them might be kept closed, depending on the quarantine level (variable quarantine-level in the simulation). In particular, it has been decided that:

- quarantine-level = 0 -> everything normal, no quarantine has been declared
- quarantine-level = 1 -> schools and universities are closed, everything else still open
- quarantine-level = 2 -> most of the activities closed, only the professional activities with kind = “factory” (the ones with highest production-value) and the health activities with kind = “hospital” or kind = “clinic” are kept open
- quarantine-level = 3 -> only health activities with kind = “hospital” or kind = “clinic” remain open

All activities will need to have workers assigned to them and schools/universities will need their students. This will be done while creating people, so it will be discussed [in that section](#).

People

General implementation

People are represented by a single breed, the division into the different age classes is operated through an attribute (age-class) varying from 0 to 6 and indicating the classes in order (0 = 0-4 class, 6 = 65 and over class) and assigning to each class a percentage of the overall people, percentages that are taken from file (“ageClass%.txt”). Other important attributes of people are the susceptibilities per situation (home-risk, school-risk, work-risk, leisure-risk, transports-risk) and appropriate for the age class of the person (obtained by the file “riskPerAgeClass.txt”), the patch of their dutiful activity (school-patch, work-patch) if applicable, various booleans to keep track of the health state of the person and the patch of their home (home-patch), although this will be set later, when creating families (discussed in the next section).

The patches related to school and work are instead set now, at creation. The people of every age class are assigned to all activities to them available equally. For example people of age class 20-24, whose dutiful activity is going to university, will be equally split and assigned to all universities, while people of age classes 25-39 and 40-64, whose dutiful activity is working, will be equally split and assigned among all activities (it is assumed that every activity needs workers). This means that there must be coherence between number of people per age class and the number of activities that age class can be assigned to (or some activities will not be correctly visited).

Movement – general implementation

The most important action people do during the simulation (along with infecting) is moving. As written previously, people move depending on cycles defined in a file (“activities-durations.txt”). In particular, each cycle can be made of any number of activities and every activity can have any (positive) duration one

wishes. To specify a cycle in the file, this must be written one couple activity-duration per row, putting the couples in the preferred order and inside the proper age class section. The specifiable activities are “home”, “school”, “work”, “recreation”, corresponding respectively to the actions of people of going to their home-patch, going to their school-patch, going to their work-patch, going to the patch of a randomly determined leisure activity. It is also possible to introduce a degree of variability for a certain time slice. This can be done by defining an activity as composed by various (any number) activities separated by a hyphen (for example “school-recreation”). By doing so, the actual performed activity will be randomly chosen among them. Note that, as a side effect, this can be useful to make an activity be chosen more likely: by writing “school-school-school-recreation” the activity “school” will be chosen 75% of the times, while “recreation” only 25% of the times.

Regarding the implementation, activities and durations are kept in separated lists different per age class (for example, activities-list-20-24 and durations-list-20-24) where at the same index will be found the components of a single couple activity-duration (at index 0 will be the first activity in the activities list and its duration in the durations list and so on). People access at the proper index of these lists, depending on at which point of the cycle they are, to determine their next target of movement. It is at this point that the hypothetical choice between various activities for that time slice is performed, guaranteeing variability even for the same person in different moments during the simulation.

[Movement – related modifiers](#)

A single activities cycle is supposedly set as related to a single day, so the durations resemble this idea. Due to the fact that an “activity duration unit” corresponds to a tick in the simulation and the fact that each movement of a person from a patch to the adjacent one requires a tick as well, the durations of activities may be disproportionally short compared to the time it took to get to them. If one wishes to avoid this, a boolean observer variable (more-realistic-activities-durations?) can be set, adjusting the durations to be preponderant with respect to the total average travelling time. This, though, will also automatically modify accordingly other time-related variables, namely each of the recovery-time variables of the individuals and the variable infect-each-n-ticks (explained [later](#) in the infection section). These variables will be scaled by the same factor that the activities durations were scaled by (factor editable in the function adjust-time-related-elements).

Regarding the quarantine, its presence may be handled when determining the next target of movement. Its effect will result in going (or staying) home if the activity the person would have normally done has been closed or moving as usual if the quarantine did not close that particular activity. A third option is, however, possible if the user decided to allow illegal behaviours (by turning on the switch possible-illegal-behaviours?). In this case, if the 5% probability of infringing the quarantine law is satisfied and the current activity does not equal “home” (to avoid the individuals not being at home when they should have been, for example to have lunch/dinner or to sleep during the night), then the person will not follow the law and go to a random leisure activity. It is very important to underline, though, that enabling the just mentioned illegal behaviours switch will also automatically turn on the more-realistic-activities-durations? switch: this due to a problem (described [later](#)) that will occur with illegal behaviours if the durations of activities are not preponderant compared to the travelling times.

[Houses and families](#)

As said above, at the creation of each person his/her home-patch is not set. This is because it is necessary to have access to all people to be able to create families and then assign to each a house. The division of people into families happens following the [rules](#) already stated before and depending on percentages defined in a file (“families%.txt”). To be precise, one can define from file:

- parents-25-39-percentage -> the percentage of 25-39 being parents (living with children)
- parents-40-64-percentage -> the percentage of 40-64 being parents (living with children)

- siblings-percentage -> determines the percentage of siblings in appropriate age classes (applied to those age classes living with parents)
- people-20-24-with-parents-percentage -> the percentage of 20-24 living with parents, contains the siblings too
- people-20-24-house-sharing-percentage -> the percentage of 20-24 living with another 20-24 (roommates)
- people-25-39-in-a-couple-percentage -> the percentage of 25-39 living in a couple, not necessarily parents
- people-40-64-in-a-couple-percentage -> the percentage of 40-64 living in a couple, not necessarily parents
- people-65-and-over-in-a-couple-percentage -> the percentage of 65 and over living in a couple

By applying these percentages to the correct age classes are obtained various pools (note that individuals of proper age classes living alone are obtainable by subtraction) from which one can extract individuals in order to create families.

A further explanation is though needed about when the rules defining families may not be followed. This may happen due to lack of people of a certain age class in a pool: a person 20-24 who should have received a roommate may not find any (because all other 20-24 were already assigned) or a person 15-19 may not find both parents 40-64 (because all the available ones were already set as parents of other individuals). This happens when there is no full coherence between the percentages of each age class ("ageClass%.txt") and the ones describing each type of family ("families%.txt") or simply because this model does not take into account all possible types of families (even using real data may in part result in this situation). When such a case happens, the person who could not form a family is assigned to an already existent one, which is why it has been said that the rules regarding the maximum number of members per family being 4 and the maximum number of children per family being 2 may not always hold.

Infection

Direct disease transmission

Starting from the infection caused by closeness, it has been already said how this happens when a susceptible individual is close enough to an infected one. Specifically, close enough means in the neighbouring patches of the infected individual (the 8 patches around his/her one) if both the sane and sick people are moving, while it means on the same patch of the infected individual if both people are not moving. Note that not moving implies being inside an activity patch (= doing an activity) and this would make the infected person, in a real-life scenario, most likely separated from people not doing that activity (he/she would probably be inside a building) or at least not really close (if he/she is inside a park then will not be near people outside of it). Hence an infected moving person only infects other moving people (= not doing an activity) in the neighbouring patches, while a not moving infected person only infects other not moving people (= doing the same activity as he/she is) in the same patch.

Furthermore, the chance of getting infected always depends on what the infected and susceptible couple is doing (again, they will be doing the same action if in the first place the simulation got to check whether the susceptible individual will be infected): if moving the infection chance is the transports-risk of the susceptible person, otherwise the proper risk of the susceptible person will be selected (home-risk if at home, school-risk if at school, work-risk if at work, leisure-risk if at leisure).

Environmental disease transmission

Moving to the environmental infection, an infected person can infect the patch he currently is on (patches have a boolean called patch-infected? to keep track of this) only if moving (same reason as before, if not moving he/she is inside an activity patch and will infect only people in the same state due to closeness). When a patch gets infected any susceptible person passing through that will have a chance of getting infected as well. This chance varies depending on how long the patch has been infected for. In particular,

each patch also has a variable called patch-infection-time-left, that is set to patch-infection-decay-time (how long a patch will be infected for) as soon as the patch gets infected. This is used to scale down the base infection chance (called base-patch-infection-chance) by a factor equal to patch-infection-time-left / patch-infection-decay-time, making the infection more likely if the patch has just been infected. Both base-patch-infection-chance and patch-infection-decay-time can be set by the user before or during the simulation.

Infection – spreading speed modifier

The spreading speed of the infection is extremely high (even without the environmental component) with the currently set susceptibilities, to the point that the quarantine effects are hardly visible. This happens since the majority of people is infected during the execution of the first activity of their first activities cycle in the simulation and the quarantine effects are enforced only from the next activity determination after the quarantine-level variable has been changed. So, to make the effects of the quarantine more noticeable, one can vary the frequency at which people infect other people around them (affecting both the direct transmission and the environmental transmission). This can be done by modifying the variable infect-each-n-ticks (normally set to 1) so that, as the name says, infected people will only try to infect each specified number of ticks passed, slowing down the direct infection spread. The environmental spread, if activated, will be diminished too, since people will be checked to possibly get infected environmentally only every infect-each-n-ticks ticks. So, by modifying the infect-each-n-ticks variable, one can make more visible in the outputs the presence of the quarantine.

Outputs

Regarding the added outputs, the counters of how many people were infected in each situation are easily obtained: people have a boolean per situation that is set to true if infected in that situation so all that is left to output every counter is just to count the number of people with the corresponding boolean set to true. About the productivity-related outputs, instead, the global productivity is calculated keeping in mind the effect of the quarantine. The productivity for open activities is their production-value, while for closed activities it is the product of the production-value by the smart-working-capability. Since the smart-working-capability is a normalized value, this results in scaling the productivity of closed activities depending on how well those can be executed from home. The global productivity is obtained as the sum of the productivities of open and closed activities.

Finally, to find the number of closed activities it is sufficient to count the ones whose colour is grey.

Illegal behaviours problem and future developments

As stated before, if the user decides to activate the switch possible-illegal-behaviours?, allowing the individuals to have a possibility to move illegally, the switch more-realistic-activities-durations will be activated as well and the already mentioned time-related variables (activities durations, recovery times and infect-each-n-ticks) will be made longer. This is done in order to avoid people moving illegally more frequently than expected (more than the usual 5%) at the quarantine levels where the movements are more limited (the higher ones). Let us see why: as said before, the check to decide whether an individual will behave illegally or not is done when determining the next target of movement, the next activity to reach. At higher quarantine levels, however, most of the activities are closed and people are forced to stay home. Being the duration of activities generally much shorter than the time it takes to get to them, when people don't have to spend time in reaching their next activity (because it most likely will have been substituted with "home", where they already are), the overall time necessary to finish that activity will shrink a lot. This means that the determination of successive activities will happen more frequently and, consequently, the check to behave illegally will be carried out much more, so that the probability to act against the law will indirectly increase.

To solve this problem activities durations must be made preponderant compared to the travelling times, maintaining a more constant illegal behaviours frequency through all quarantine levels, since the lack of

travelling times at higher levels will not influence that much the overall time necessary to finish activities, keeping a new activity determination frequency similar to the one of lower quarantine levels, so that the illegal behaviours check will be done more uniformly at all levels. So this is why the switch more-realistic-activities-durations is enabled automatically.

Obtaining proper invariance, though, is not easy since the factor by which multiply the time-related elements (that can be seen and modified in the function `adjust-time-related-elements`) must be high enough that the probability of illegal behaviours will fall around 5% at all levels, but low enough that the simulation will have an acceptable duration (or another solution could be maintaining really high activities durations and employing parallelization of the simulation). In the next section, where some simulation results are shown, will be described the used parameter configuration, configuration that maintains acceptable runs durations but only limits the illegal behaviours problem, without completely solving it: the fine-tuning of the simulation falls under the future developments.

Other hypothetical future developments may be:

- The addition of the possibility of dying due to the disease, if the individual could not recover after a certain period of time, thus shifting to an SIRD (susceptible-infectious-recovered-deceased) epidemiologic model
- The addition of possible mutations of the virus, that would make cured people susceptible again

Results

Results – description

As just explained, the parameter configuration used for these result runs does not completely solve the illegal behaviours problem. However it reduces it, allowing to obtain outputs that are coherent with what one would expect. In particular, it has been found that, by using the parameters shown in the images (transcribing only the ones useful for these first runs: `initial-people = 5000`, `recovery-chance = 30`, `average-recovery-time = 500`, `infect-each-n-ticks = 20`, `possible-illegal-behaviours = true`, `more-realistic-activities-durations = true`) and setting the tuning factor inside the function `adjust-time-related-elements` so that activities durations (and the other time-related variables) would have been scaled up 15 times, the illegal behaviours problem was for the most part contained.

In *Figure 1* are shown the results of the run with the described parameters and, in addition, the quarantine-level variable maintained to 0. We can see how not enforcing any kind of quarantine allowed the virus to spread freely, nearly infecting the totality of the population. Due to the increased duration of activities, people are mostly infected while doing them, in fact around 50% of the population was infected while performing their dutiful activity (school or work). This is coherent with reality, since there will be higher density of people inside school and workplaces, favouring the infection spread.

In *Figure 2* are instead shown the results of a run in which the quarantine level was set to 1 at the 10th tick. Reminding the effects of the quarantine level 1, those are not very limiting: in fact this level consists in only closing all education activities, keeping though still open every leisure activity, that can potentially be visited by any moving individual. This is why the results of this run are similar (graphs-wise at least) to the one with `quarantine-level = 0`. There are, however, differences: first of all there are obviously no infected people at school, since these were closed. People infected at work diminished too, since all teachers could not go to work anymore. This, though resulted in an increased number of people infected while doing recreational activities and while at home. This probably due to teachers and students who did not get infected respectively at work or school but while doing recreational activities and then were forced to stay home during their dutiful activity time slice, increasing the probability to meet another family member to infect. We can also notice how closing schools slightly worsened the productivity and how, on the bright side, R_0 decreased.

Moving to *Figure 3*, here is represented a run where the quarantine-level was set to 2 at the 10th tick. Since level 2 has effects of a certain gravity (all activities closed aside from factories, hospitals and clinics), we can see a significant improvement in the results: now only 1452 people out of 5000 got infected before the disease disappeared. People infected while moving are nearly not present since movements are very limited and people tend to stay home, although this means a relatively high presence of people infected while at their house. There still are infected people while at work since some activities are still open and we can see how the presence of illegal behaviours made so that some people got infected while being at leisure activities, although closed. It is probably due to these two causes that the number of infected people is not extremely low. We can also notice how R_0 diminished a lot (making by definition this virus not a pandemic, since $R_0 < 1$), even though these good results are counterbalanced by a heavy decrease in productivity.

In *Figure 4* we can see a run where the quarantine-level was set to 3 at the 10th tick. The results are generally similar to the ones of level 2, due to level 3 having only the additional effect of closing factories too, leaving open only hospitals and clinics. Despite only closing a small number of activities, these activities are factories, which are modelled as the most productive activities among all. This, paired with the fact that factories have an extremely low smart-working capability, determines a significant decrease in productivity. We can though appreciate how the overall infected people got lower, as well as R_0 , the infected people while at work and the infected people while at home. The infected people while moving and while at leisure activities did not nearly change at all. This can be explained by how similarly levels 2 and 3 limit movements, resulting in a similar number of illegal behaviours and, consequently, of the underlined outputs.

Finally, in *Figure 5* are shown the results of a quarantine-level 2 run where at the 10th tick was also activated the environmental infection (with related parameters patch-infection-decay-time = 5 and base-patch-infection-chance = 75). As stated before, the environmental infection too is affected by the influence of infect-each-n-ticks, which is why the results do not change much. It can be noticed though an increased number of people infected while moving (which is when the environmental infection acts) that is reflected in a generally higher number of infected people around all outputs, as well as in a slightly larger R_0 .

All these result runs refer to the current simulation settings, comprising, aside from the already described parameter configuration, a 100x100 patches grid (each patch being of size 10 pixels) and an age-class and families people division, a set of susceptibilities, a set of activities cycles, a set of activities that can be found in the corresponding .txt configuration files.

Results – figures

All figures can be also seen in the folder “results” on the Github repository

(<https://github.com/gpisanelli/EpiDEMExtended>)

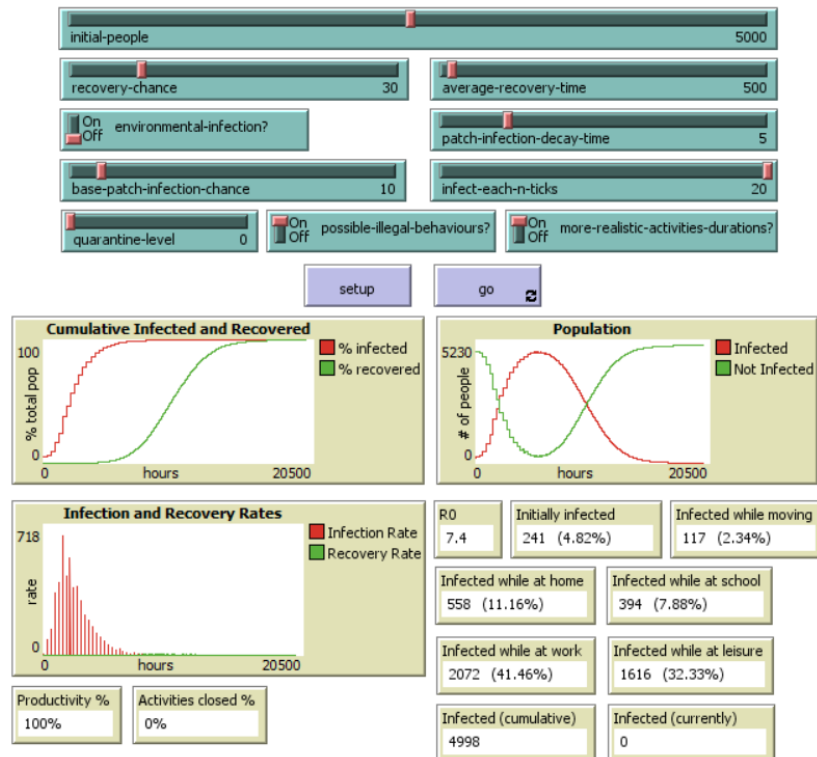


Figure 1 - results of a quarantine 0 run

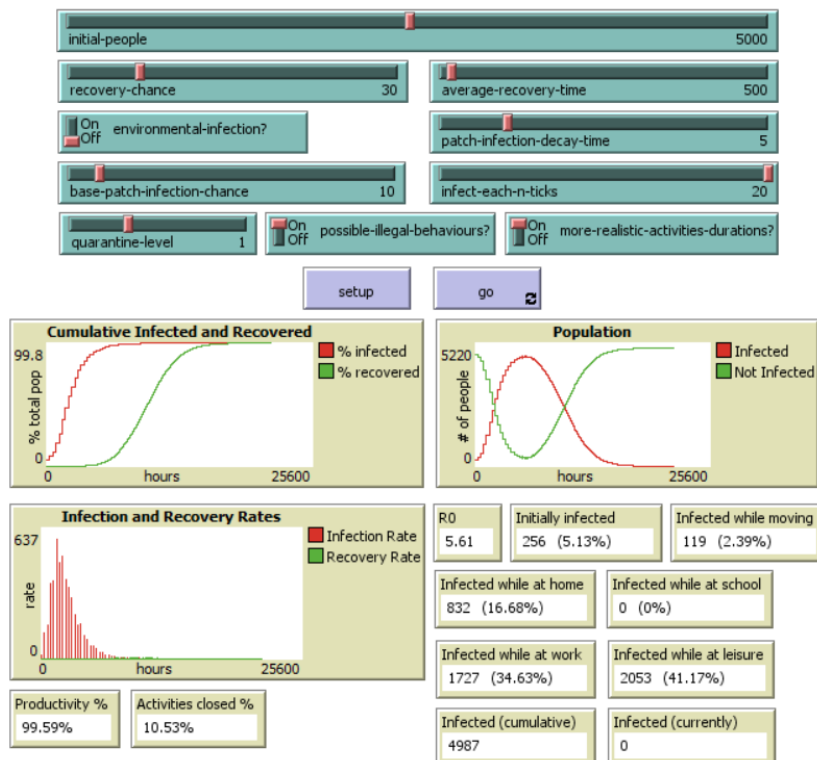


Figure 2 - results of quarantine 1 run

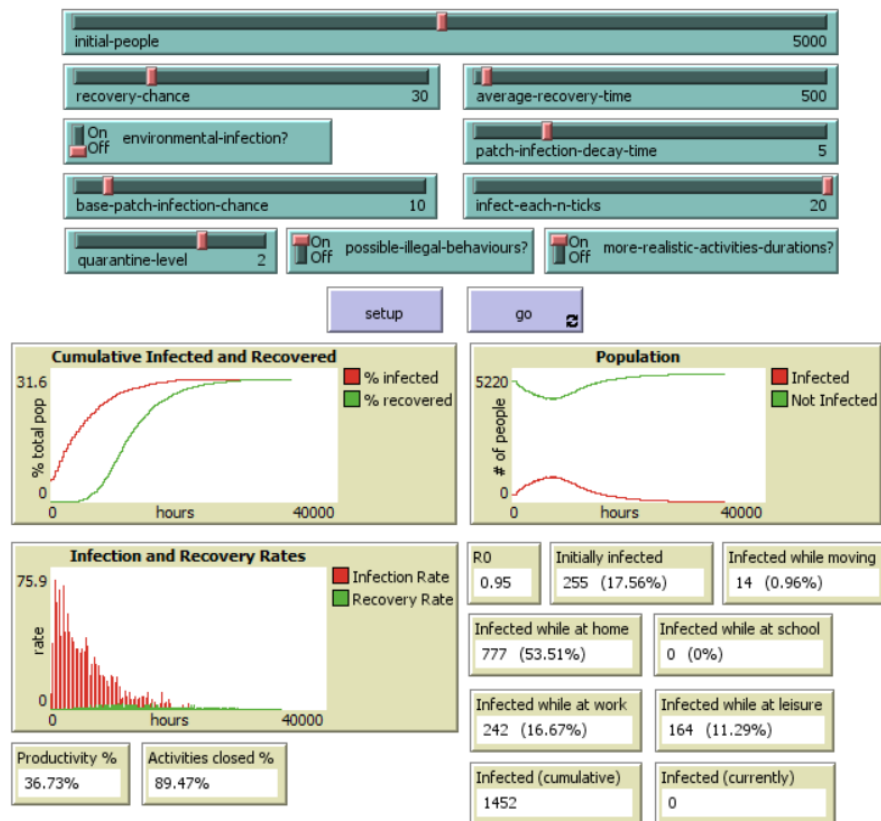


Figure 3 - results of a quarantine 2 run

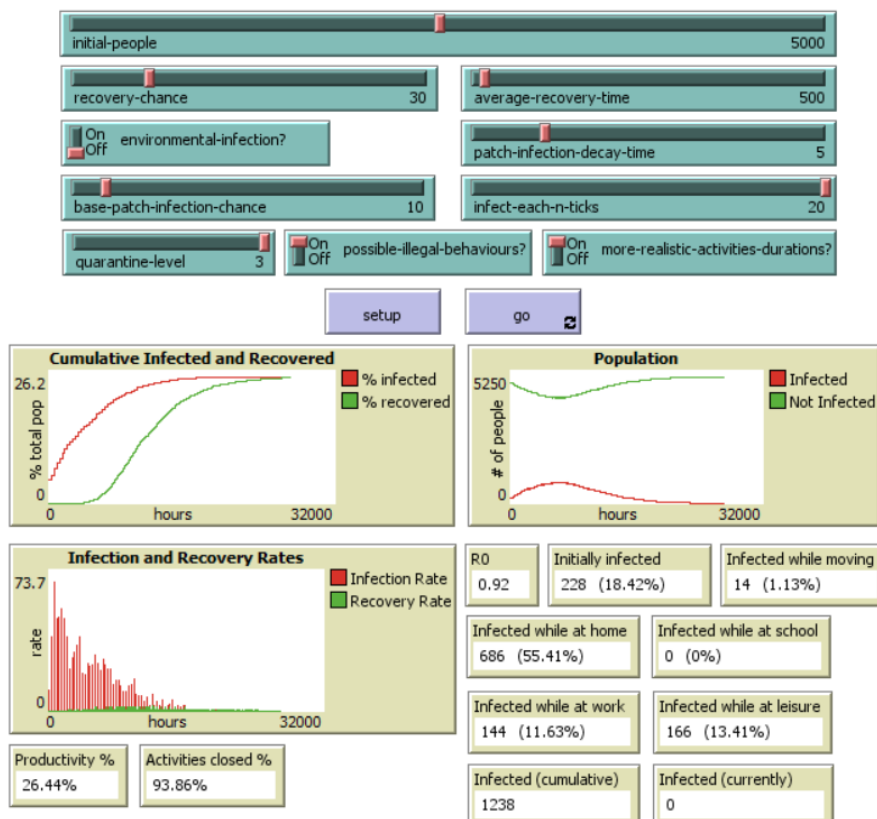


Figure 4 - results of a quarantine 3 run

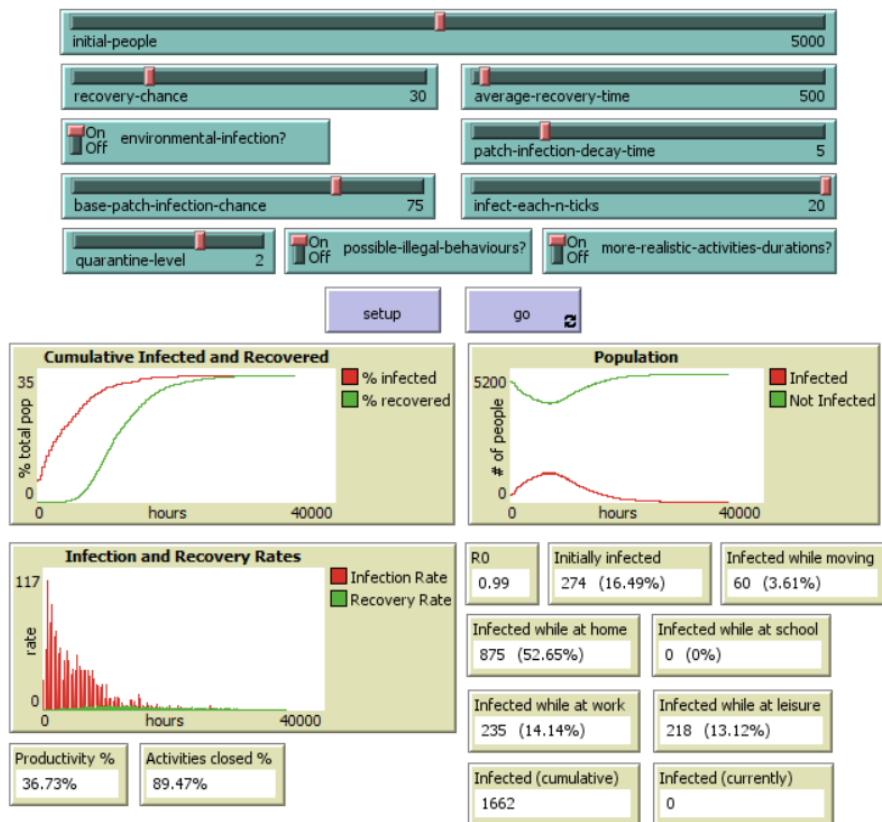


Figure 5 - results of a quarantine 2 run with environmental infection