Stride Simulations

Cedric De Schepper[1], Thanh Danh Le[2], Wannes Marynen[3], and Stijn Vissers[4]

[1] University of Antwerp, Antwerp, Belgium
Cedric.DeSchepper@student.uantwerpen.be
[2] University of Antwerp, Antwerp, Belgium
ThanhDanh.Le@student.uantwerpen.be
[3] University of Antwerp, Antwerp, Belgium
Wannes.Marynen@student.uantwerpen.be
[4] University of Antwerp, Antwerp, Belgium
Stijn.Vissers@student.uantwerpen.be

1 Introduction

The Simulator for the Transmission of Infectious Diseases (Stride) is a simulator for the transmission of infectious diseases. To get a better understanding of how Stride works and what it does specifically, a set of questions were answered using Stride. The results are discussed in more detail below.

2 Simulation

2.1 Stochastic variation

When running multiple simulation runs of the exact same situation, it's very easy to see that there exists a large difference between the number of infected cases observed during every run. Either the number of infected cases is minimal, or the amount rises to a fairly constant number.

This indicates that a form of chance is present during the simulation. This stochastic variation must be taken into account during our analysis or it might influence our findings greatly.

When analyzing the cumulative cases per time-step, we can distinguish two separate cases: one where an outbreak occurs, and one where an outbreak is averted. In the first case, the graphs start rising slowly, when at a certain point they start rising exponentially. After this huge rise, the graph tends to the maximum, rising much slower than before. In the second case, the graphs experience a rise, thus coinciding with each other at the bottom.

The number of new cases per time-step form a bell-like curve when an outbreak occur, otherwise it is a mostly flat line with maybe a couple of bumps, signifying the couple of transmissions that may happen before extinction occurs.

This means that the number of transmissions per day starts slowly. Then the disease starts spreading faster, until it reaches its full potential. After this point has been reached, it is not as easy as before to find susceptible victims, and thus the number of cases dwindles until the final amount of cases is reached.

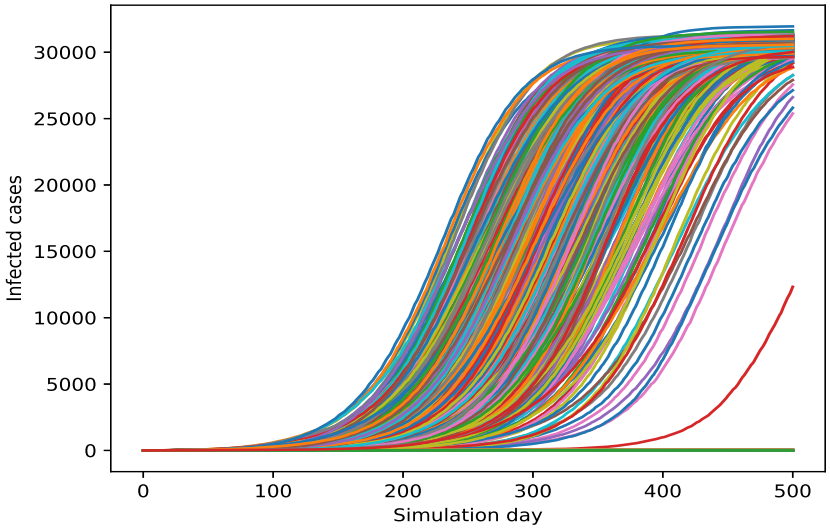

Fig. 1. Cumulative cases of 1024 simulations with a period of 500 days

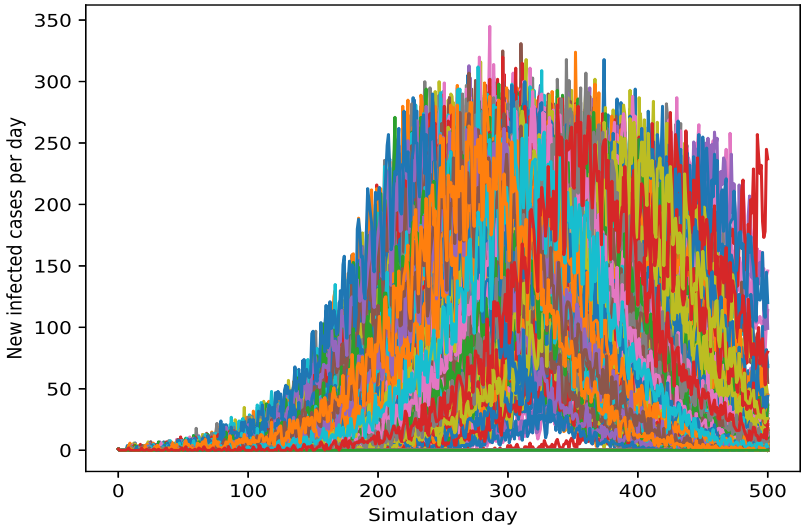

Fig. 2. Newly infected cases per day of 1024 simulations with a period of 500 days

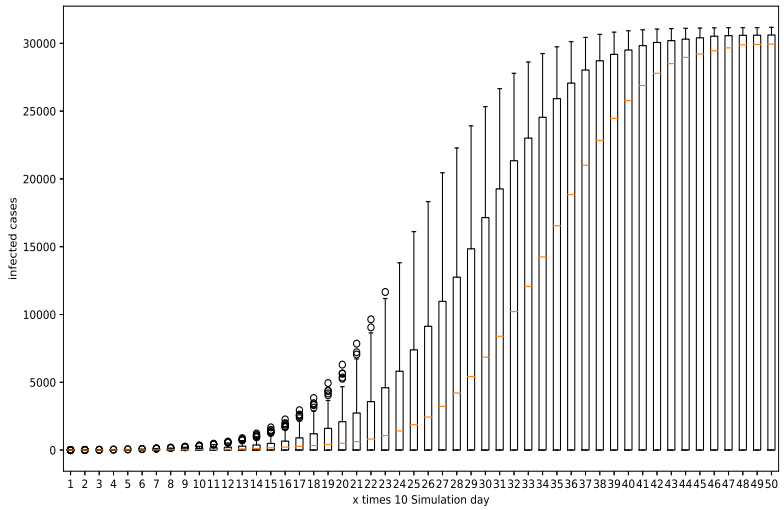

Fig. 3. Boxplot of cumulative cases of 1024 simulations with a difference of 10 days

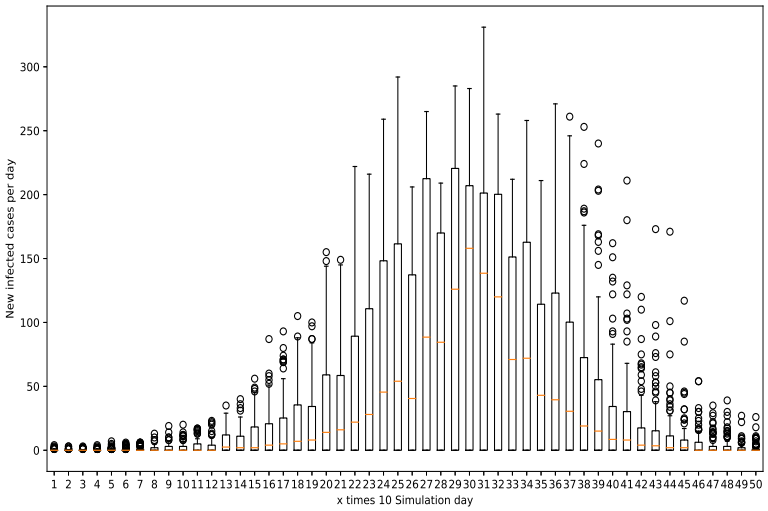

Fig. 4. Boxplot of newly infected cases peo ray df 1024 simulations with a difference of 10 days

2.2 Determining an extinction threshold

As mentioned before, there is a clear distinction in the simulations, that can be separated into two cases. In the first case, no outbreak occurs, and the disease stops spreading after infecting in between zero to thirty people. In the second case, an outbreak occurs. This results in the disease spreading among the population, infecting around thirty thousand people.

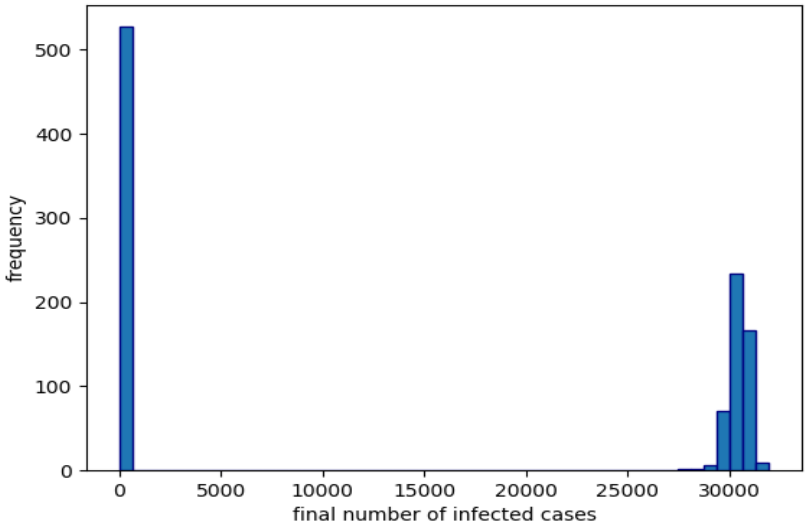

Fig. 5. Frequencies of the final number of infected cases displayed in several bins showing a clear distinction between outbreaks and extinctions

Thus, the distribution of cases per time-step is almost equal to zero when no outbreak occurs, or the number of cases increases until approximately thirty thousand people are infected during the event of an outbreak. This corresponds approximately to 5% of the total population, while 10% of the population was not vaccinated. this means that only 5% of the unvaccinated population is not infected.

Why do we see a 50-50 relation between extinctions and outbreaks? Here the stochastic variation influences the location of our "patient zero". Depending on his location, an outbreak either occurs or not, thus influencing the number of cases observed.

In real life, such a phenomenon can be observed, in particular when a community decides not to vaccinate themselves because of e.g. religious reasons.

This results in all of the susceptible people in the community being infected when a case is introduced. On the other hand, when a case in introduced in a community where everyone is vaccinated, the disease has no chance of spreading, and thus outbreaks are prevented.

Now that we have determined that there are two possible outcomes,extinction & outbreak as seen in Fig 5, we can try to determine a threshold to distinguish them. By plotting the total infected cases of a large amount of simulation runs, it might be possible to find the threshold.

As see in Fig 6, the amount of cases suddenly increases noticeably, indicating an outbreak. One might deduce the threshold by taking the highest value before the outbreak occurs. However due to the stochasticity, it is impossible to deduce an exact value. One might overcome this issue by multiplying the found threshold by a factor. This doesn't guarantee an exact threshold but might be sufficient for certain needs.

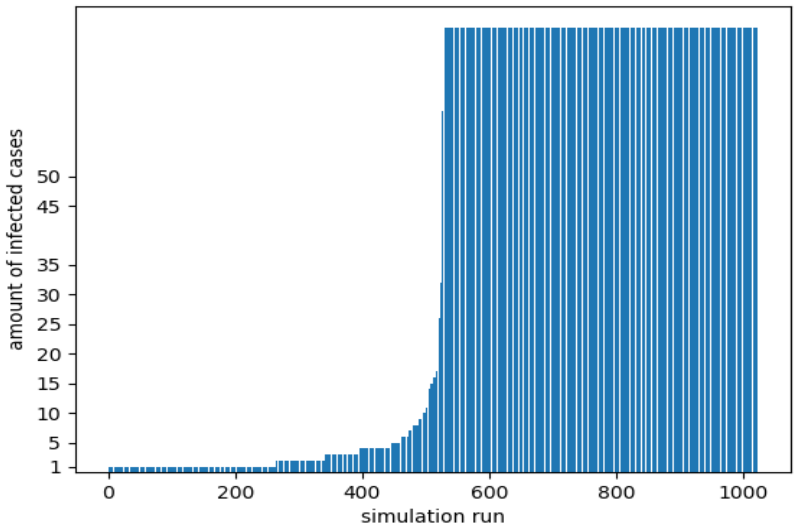

Fig. 6. Final number of infected cases for 1024 runs

2.3 Estimating the immunity level

We have to determine the immunity level based on the limited data provided in the picture. The main data we needed to use to determine the immunity level is the cases per day. After testing the complete range of immunity's we determined that the immunity of the real population will be around 70%.

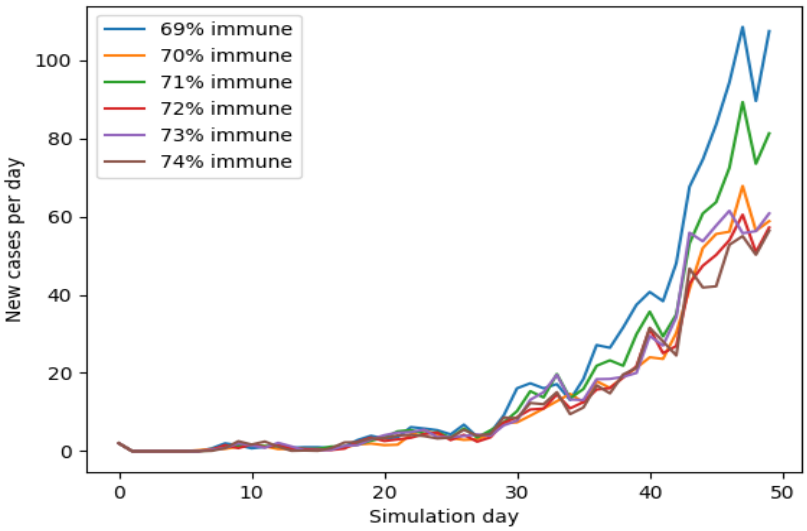

Fig. 7. new cases simulated data

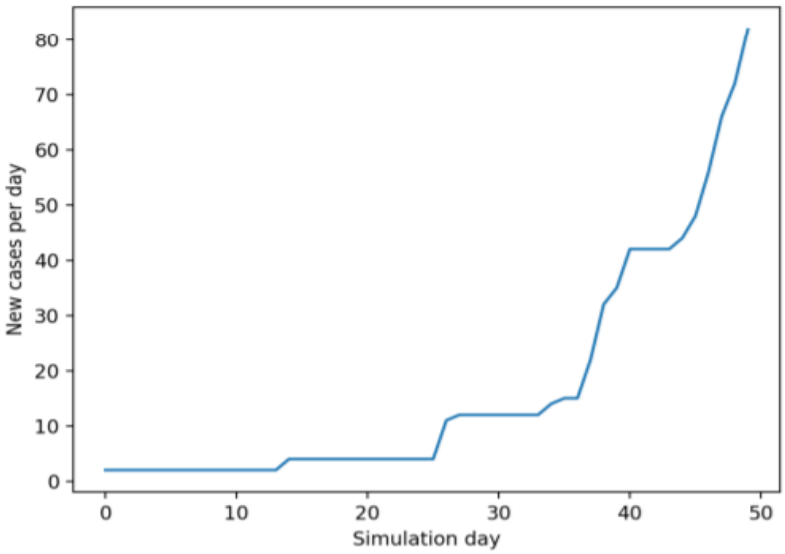

Fig. 8. new cases from original data

To estimate a closer percentage we use a smaller range, from 69% to 75%. The latter test shows us that the real percentage will be roughly 71%.

To improve the accuracy of the simulations we ran each simulation 20 times and then we used the mean value of the new cases per day of the 20 simulations.

2.4 Estimating R_0

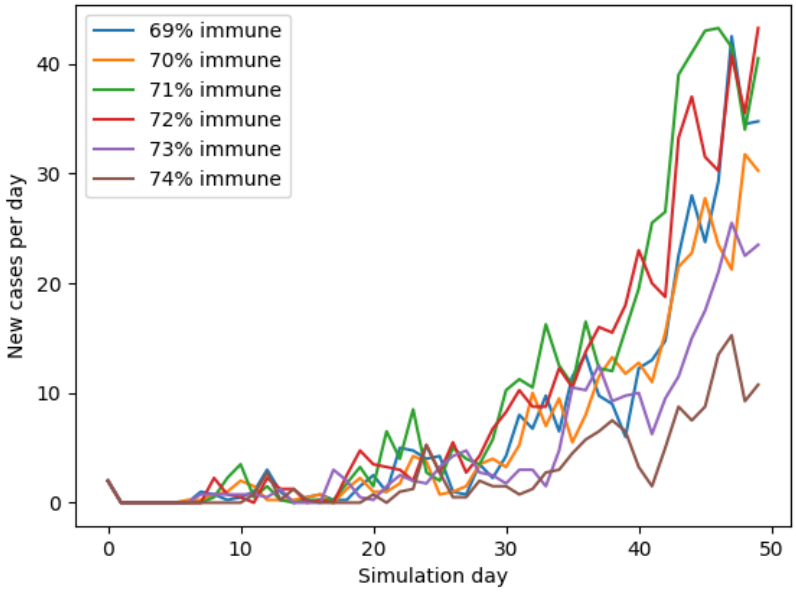

Fig. 9. new cases per day with an R0=12

When we check the conclusion of the last question with other R0's we see that it only holds for values around 14 which was the R0 that was used in the simulation for the previous assignment.

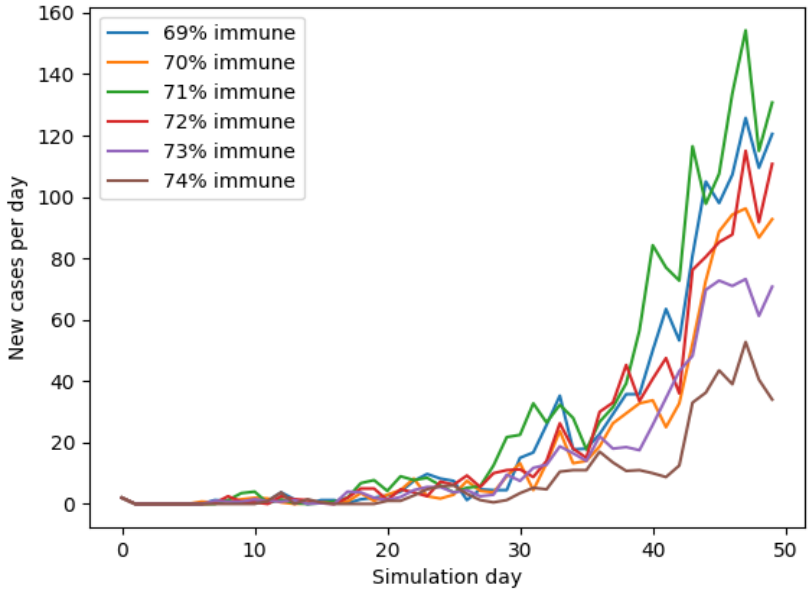

Fig. 10. new cases per day with an R0=18

For an R0 value of 16 it is clear that the value of the immunity level will be around 74%. This increase is logical because the disease is more infectious so with the higher number of infectious contacts per day a smaller percentage of people need to be infected to keep the same amount of new cases per day.

3 Population generation

3.1 Investigating the influence of demography on epidemics

Given two populations (see Fig. 11) where there is only a significant difference in the age distribution, along with the fact that an outbreak won't affect all age groups equally. Following assumption can be made.

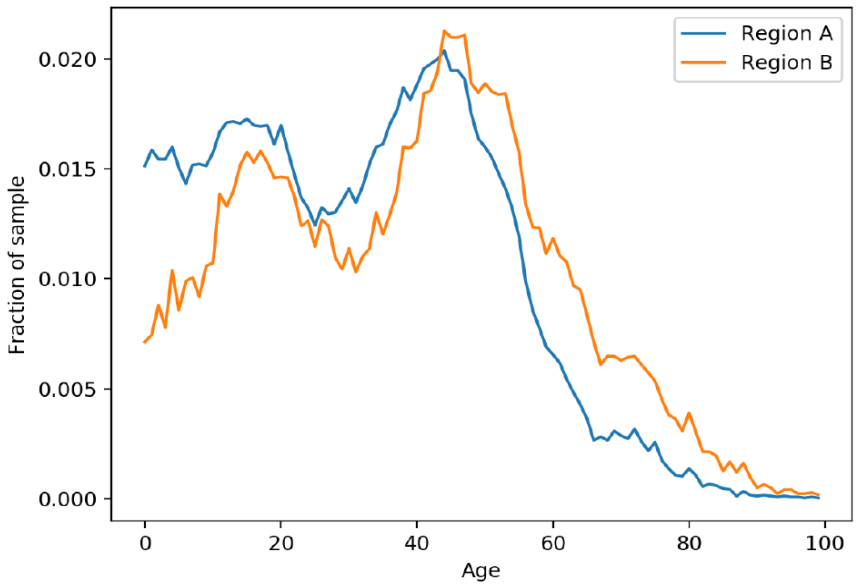

Fig. 11. Comparison of age distributions in household samples from region A and region B.

Assumption. *An outbreak in two almost identical populations where there is only a distinction in age distribution, will not behave in the same way.*

Using Stride, a number of simulations can be run for each population. Before running a simulations, a few parameters must be set accordingly to produce adequate results.
The duration of a simulation will be set to 365 days which equals a year. This gives room to observe what happens after the peak of an outbreak. Next, the initial number of infected cases is set to equal 1 person. This is done by setting the seed to 0.00000167. An outbreak must start with someone after all.

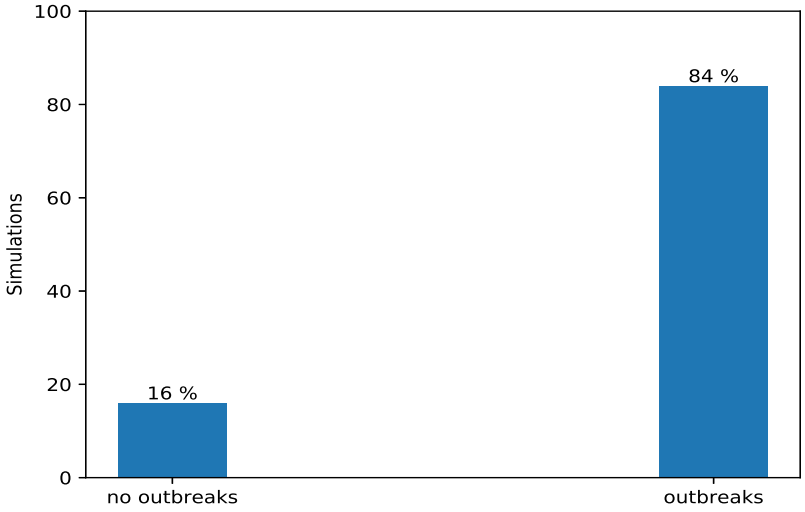

Fig. 12. Percentage of outbreaks in region A after 100 simulations

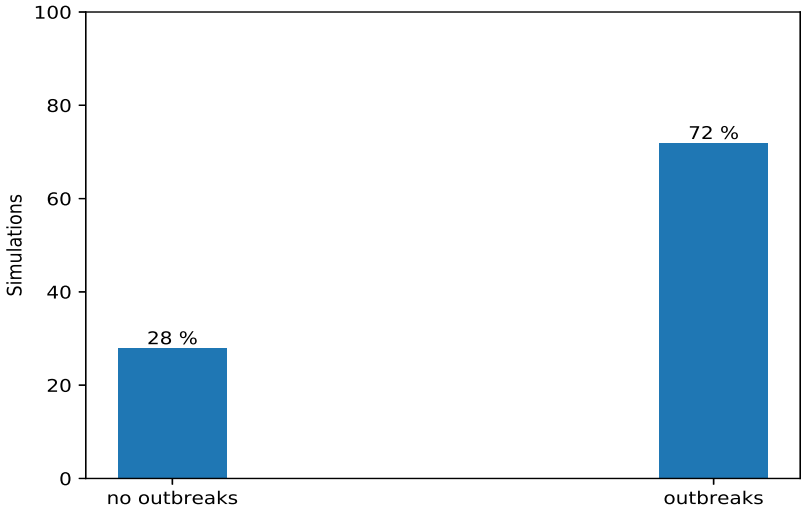

Fig. 13. Percentage of outbreaks in region B after 100 simulations

As seen in Fig 12 and Fig 13, region A will have a higher chance of outbreaks. Although the difference is roughly 10% between the two populations, this does not mean that the chances of outbreaks in region B are low. However these two graphs don't really show how an outbreak transpire in either of the populations.

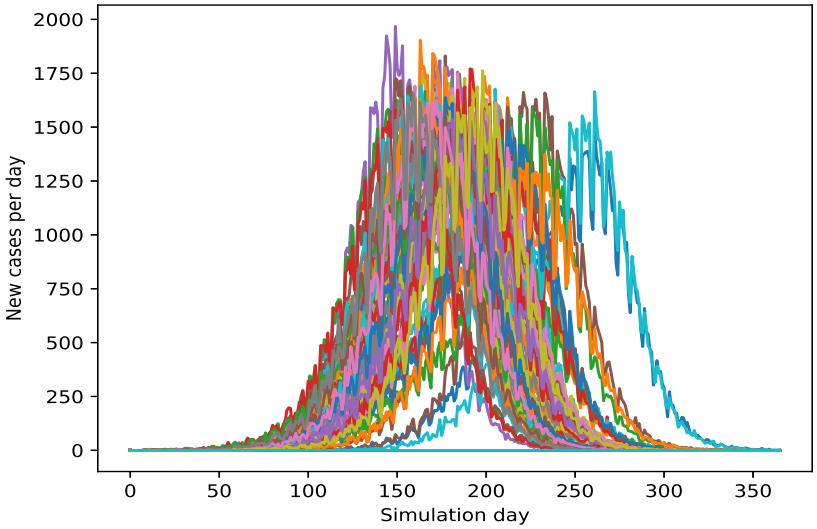

Fig. 14. New cases of infected per day for region A of 100 simulations over a period of 365 days

Fig 14 and Fig 15 display the newly infected cases per day over a simulation period. Here, it is noticeable that the peak of an outbreak in region A can be reached earlier than in region B. Thus, in region A an outbreak can die out earlier too. Next, it can be noted that the peaks in region A can reach a higher number of newly infected cases per day. A Hypothesis could be that because region A has a younger population, There would be more commuting which means more people come into contact with each other. This could then potentially lead to more infected cases per day.

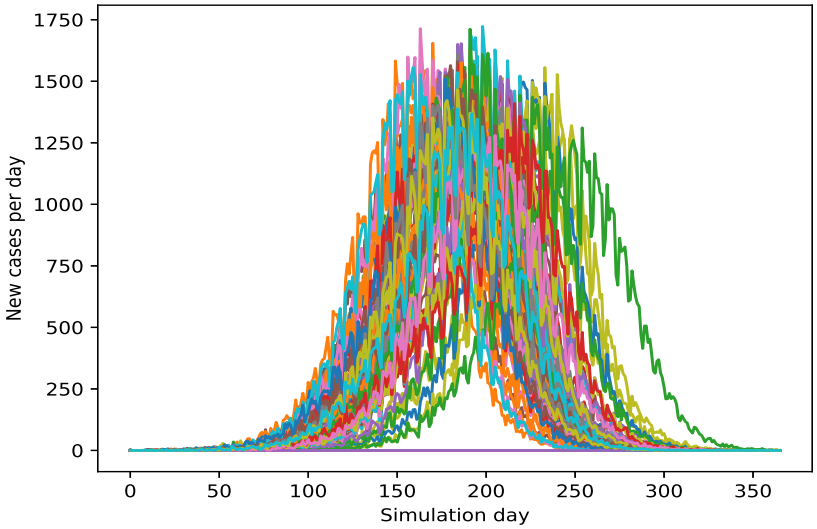

Fig. 15. New cases of infected per day for region B of 100 simulations over a period of 365 days.

Remark. *When using PyStride, a generated population (protobuf file) could not be imported using the import parameter in the config file. Therefore, the data was generated with the STAN controller.*

3.2 Vaccinating on campus

When there's no vaccination in general, it is expected that the amount of infected will increase day by day depending on the disease. Eventually reaching a peak.

Now as seen in Fig 17, when only college students between 18 and 26 are vaccinated, It can be noted that vaccinating does have a clear effect. Looking at Fig 17, the effect of vaccinating after a week can be clearly seen after day 20. From then the period between new infected cases, becomes longer and the amount of new infected cases per day does not seem to go higher than when no vaccination took place.

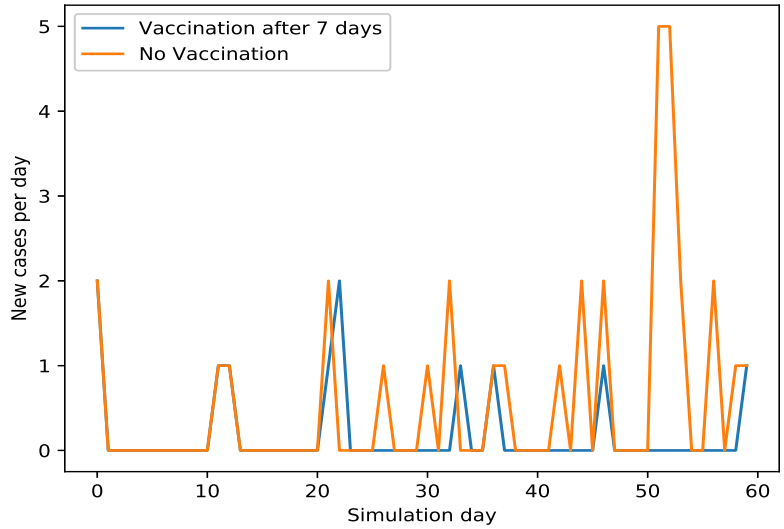

Fig. 16. Newly infected cases per day when vaccination was done after 7 days and when not.

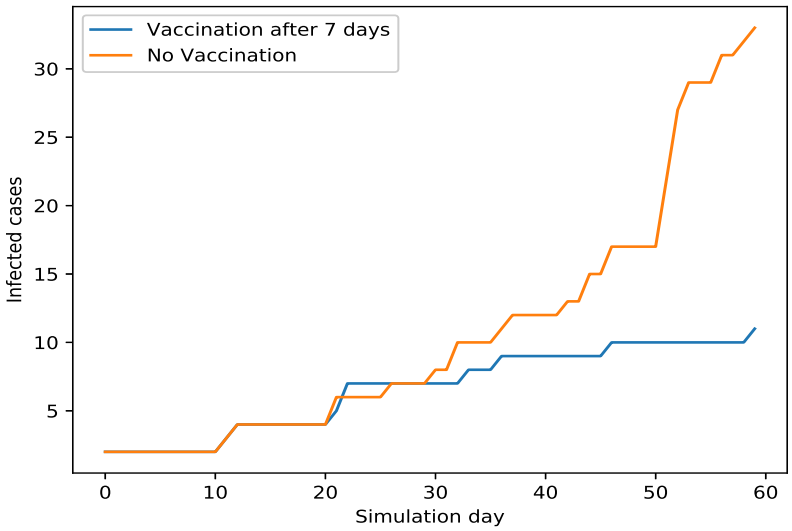

Fig. 17. Cumulative cases when vaccination was done after 7 days and when not.

Remark. *For this simulation PyStride had to be used to vaccinate after 7 days, however when running multiple simulations only 1 seed was used repeatedly. Also setting the seeding_rate to 0,00000167 which corresponds to 1 initially infected person, gave unrealistic results.*

3.3 Is commuting to work important for disease spread?

Assumption. *The percentage of the commuters will have an effect on how fast a virus spreads.*

Using stride we tested this assumption and ran simulations for commuter percentages from 0% to a 100%. These simulations confirm our assumptions as it shows that the 0% commuting population takes longer to reach its peak and has a longer period where it is infecting a large amount of people. While the 100% commuting population peaks quicker and is shorter as infectious.

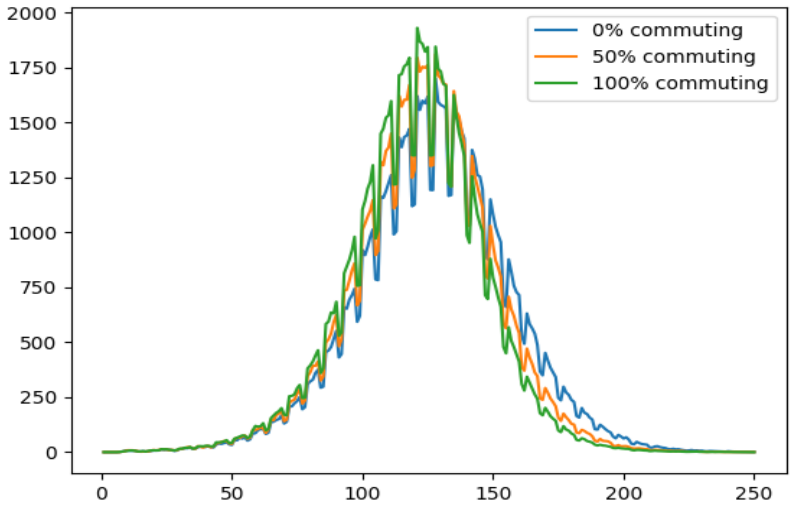

Fig. 18. The means new cases per day of 16 simulation runs per commuting level

4 Performance profiling of sequential code

For the last part of this paper, we analyzed the running times of stride using the *GProf* tool.
To be able to do this, we first had to recompile the stride project with the extra `-pg` flag for the `CXX` compiler. After doing so, our stride executable will now dump some output when executed. Afterwards, the *GProf* tool uses this output to analyse the performance.
This output is not readable at all due to the size of the stride project. Because of this, we used another tool called *gprof2dot*, which is able to create a dot representation of the call graph.
This representation is much easier to analyse, and the results of our analysis can be found below under the form of bar charts. In these charts, the time (in seconds) of each procedure will be plotted and compared to the default case.

Assumption. *Different extreme values for parameters will have an effect on the total runtime and the percentage of time spent creating the population versus actually running the simulation.*

4.1 Default case

Before we start varying the different parameters, we first established a default scenario to compare our results with.
In Figure 19 the results of our default scenario can be seen. On the left side of the figure the time it took to create the population and run the simulation, when reading the population from a file, is shown. On the right hand side, you can see the data when the population is generated instead of read from a file.

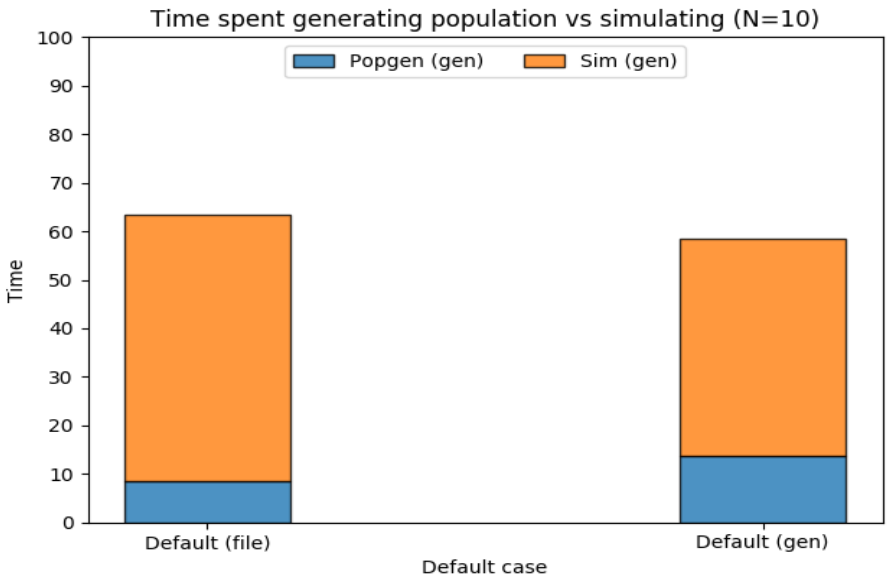

Fig. 19. Analysis of the default scenario.

4.2 Number of days

As you can see in Figure 20, the number of simulation days clearly play a role in the distribution of the weight of the program. The mean of the running times of the simulator are surprisingly enough higher than in the default case, both when there are less, and much more days than the default. We can also see, that the percentage of the time used for generating the population, or performing the simulation varies greatly, depending on the amount of days.

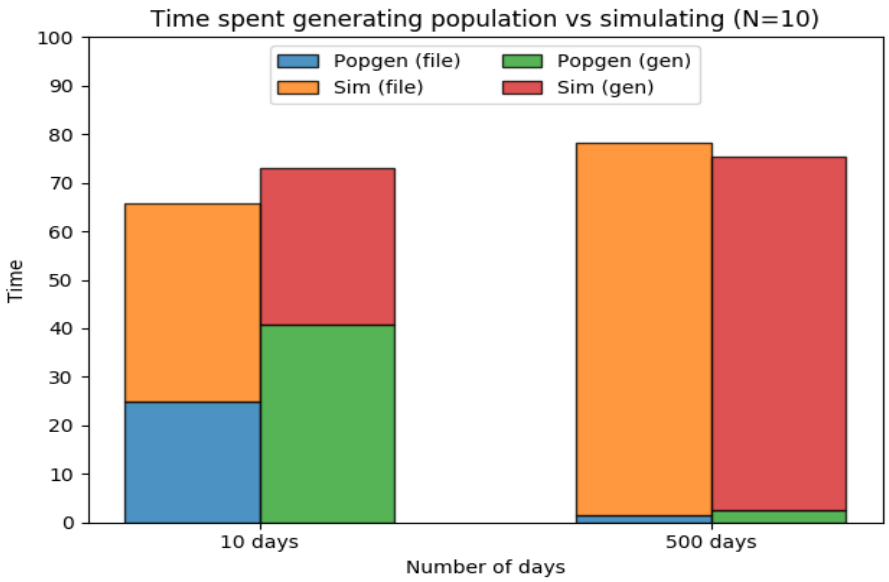

Fig. 20. Analysis of the influence of number of days.

4.3 Population size

For this parameter, we tested the simulator with 10x as much, and 10x less people as in the default case. All in all, this doesn't affect the simulator that much, as no great differences can be observed in Figure 21. For this part, we only generated the population, and did not read it from a file. This is due to the fact that there are no files for a population of 60K or 6M and because the generate and import settings are not working in the current version of stride.

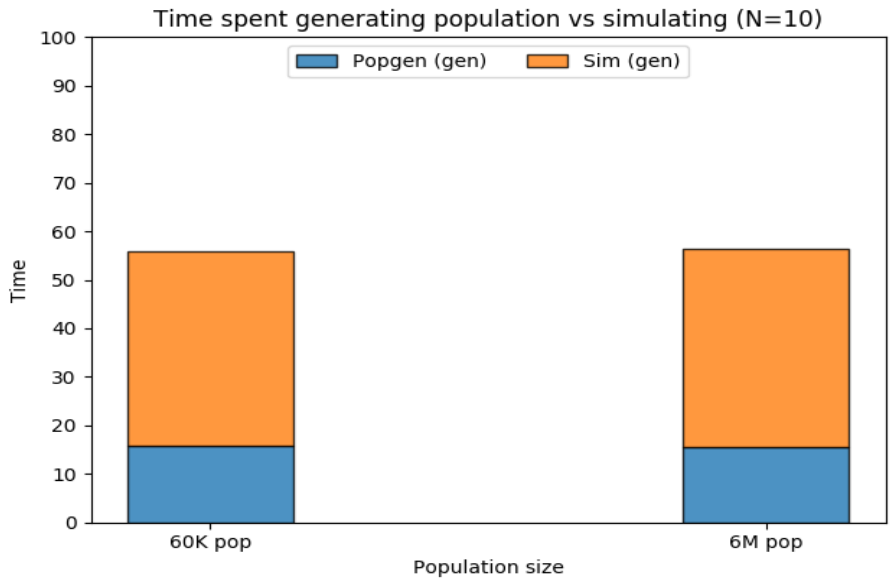

Fig. 21. Analysis of the influence of the population size.

4.4 Immunity rate

Concerning the immunity rate, the same can be observed as before. No big differences can be seen when varying this parameter. The results of our runs can be found in Figure 22

4.5 Seeding rate

According to our results, visible in Figure 23, the seeding rate does not affect the distribution of time used in the simulator.

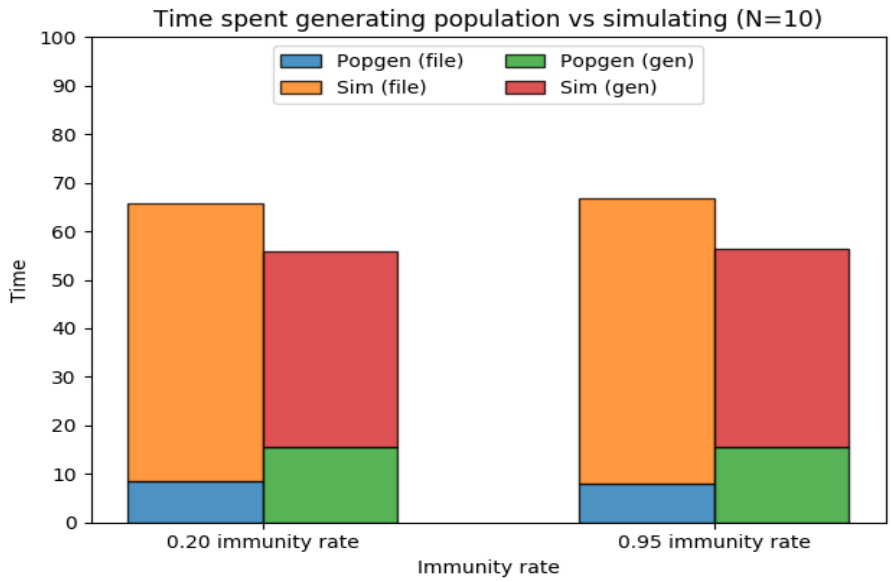

Fig. 22. Analysis of the influence of the immunity rate.

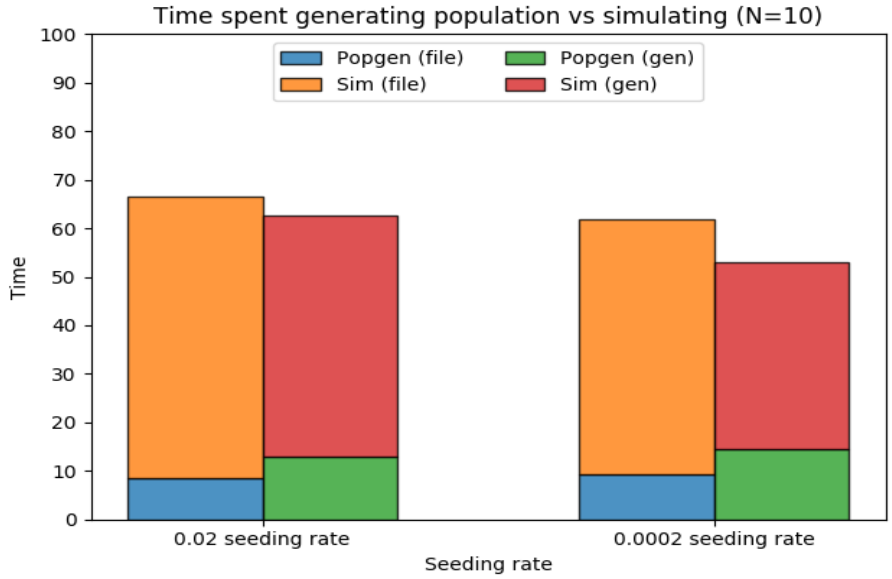

Fig. 23. Analysis of the influence of the seeding rate.

4.6 Contact log mode

Out of the data from Figure 24, we can see that no speed gains are made when using little or no logging versus a logging mode such as e.g. trace.

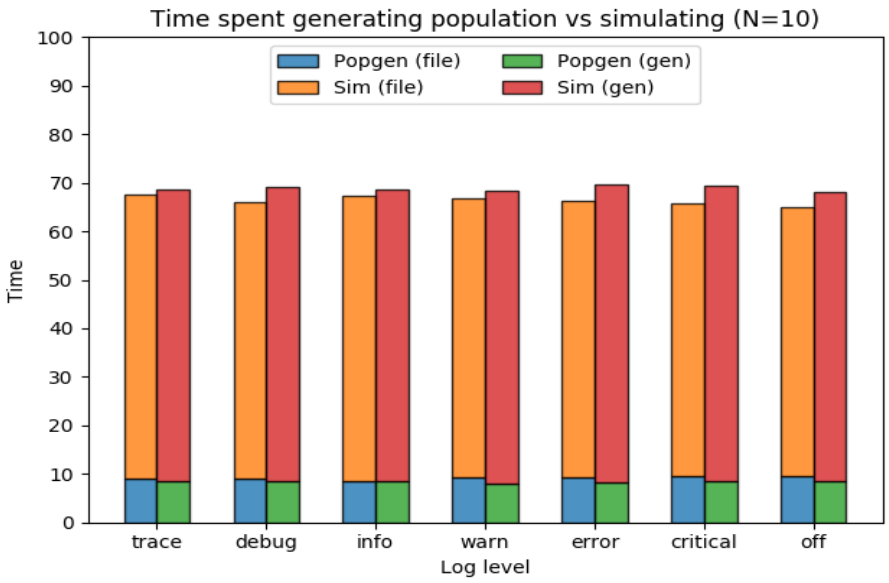

Fig. 24. Analysis of the influence of the contact log mode.

5 Discussion

Solving these assignments has learned us a lot about stride. Apart from the fact that we now can properly use the different components in the stride project, we also learned about the inner workings of stride.

The main thing to remember for the future is to check the small details. It has become clear that, for most cases, small changes or errors can greatly impact the results of the simulator. That is why it is important to verify our source data, to ensure that stride will be correctly simulating the outside world.

On the other hand, we also observed that our results are not always in compliance with our hypotheses. E.g. when checking if the work commuters contribute to the spread of diseases, we found results that contradict our intuitive thoughts. This does not always mean that we have introduced an error into our source data, but it can just be the result of a normal simulation. It is important that we examine both possibilities before rejecting a hypothesis.

References

1. Kuylen, E.: Social Contact Patterns in an Individual-based Simulator for the Transmission Infectious Diseases. ScienceDirect (2017)
2. BA Project Simulation Assignments (2019)