BAE-Simulation Assignments

Cedric De Schepper, Thanh Danh Le, Wannes Marynen
Stijn Vissers

February 2019

1 Simulation

1.1 Stochastic variation

1.1.1 What can you say about the distribution of cases per time-step?

We can clearly divide our simulation runs in two separate cases.
In the first case, no real outbreak occurs, and the disease stops spreading after infecting in between zero to thirty people.
In the second case, an outbreak does occur. This results in the disease spreading among the population, infecting around thirty thousand people.
Thus, we found that the distribution of cases per time-step is either almost equal to zero when no outbreak occurs, while the number of cases rises until approximately thirty thousand people are infected when an outbreak does occur. This is approximate 5% of the total population, while 10% of the population was not vaccinated, this means that 5% of the unvaccinated population is not infected.

1.1.2 Does chance seem to be an important factor in determining the outcome of a simulation?

In short: yes.
As you can observe in the graph (see figure 1), there exists a large difference between the number of cases observed in the simulated runs. This is probably due to the placement of our "patient zero".
Depending on the placement of our patient zero, an outbreak either occurs or it does not, thus influencing the number of cases observed (note that in the graph it seems that only one run gives no outbreak but the horizontal line actually consists of around 50% of all runs).
In real life, we can sometimes observe such phenomenon, in particular when a community decides to not 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.

1.1.3 Can you think of an explanation for the distribution of cases per day that you observe?

We will again discuss two different cases, one where no outbreak occurs, and one where an outbreak does happen.
When there is no outbreak, the number of cases per day is pretty much static. Occasionally, a single person may get infected, but this is not the standard case, as the average of cases per day is more or less equal to zero. This can be explained with the placement of our patient zero. It is likely that when patient zero is placed in an environment with a high vaccination rate, a low number of cases will be observed.
When an outbreak does occur, we observed that the number of cases per day rises exponentially, until a certain maximum. After this maximum is reached, the number of cases per day shrinks again, until no one is further infected. When we plot this data, this results in a Bell-like curve of cases per day. We think this is due to the fact that at a certain point, no one that is susceptible will be exposed to the disease anymore, thus shrinking the amount of cases per day.

1.2 Determining an extinction threshold

1.2.1 Create a histogram of the frequencies of the final number of infected cases over the different simulations. Describe what you see.

When analysing a histogram containing all the exact frequencies, it's clear to see that it doesn't contain enough interesting information (See figure 2). By changing the histogram to contain bins, we get a different situation (see figure 3). Now its clear to see that we have 2 major cases. Firstly, we have a bin containing 50+% of all simulation runs. The low number of infected cases indicates that this bin contains all extinction cases. Secondly, there are several bins that together contain all the outbreak cases.

1.2.2 Is it possible to determine a threshold distinguishing extinction cases from outbreak cases?

By changing the histogram a bit, it becomes possible to clearly see a maximum amount of infected cases to remain an extinction case. However, this amount is not necessarily the actual threshold. The stochasticity present in the simulation, makes it difficult to get an exact result.

1.2.3 What threshold would you choose in this case?

After sorting the results of 128 simulation runs by increasing amount of infected cases (see figure 4), you might think the threshold is 15 in this case. However, when running another simulation that actually is a lot smaller (see figure 5), we notice that the threshold seems to be 25. As mentioned before, this difference occurs due to the stochastic influence. However one might deduce the threshold by taking the highest value before an outbreak occurs (15 or 25 in our simulations) and multiplying this by a factor like 2 or 4. This doesn't guarantee a correct threshold but might be sufficient for certain needs.

1.3 Estimating the immunity level

1.3.1 Vary the parameter that determines the immunity level in the population, and try to estimate which was the case in the original population

Remember that:

- Stochasticity plays a role in Stride
- The data we have are from an outbreak of measles

To solve this, we started running simulations with different immunity levels. We used the median to reduce the effect of stochasticity in Stride. Doing all this, made it possible to reduce our range of possible immunity levels to 69-71 % by comparing the different plots. Analysing figure 6, an immunity level of 71% seems to behave the most like the plot given (figure 7).

1.4 Estimating R_0

1.4.1 Re-examine the question of the previous assignment, but take different values of R_0 into account

- What do you notice?
- Is your conclusion dependent upon your estimation of R_0 ?

By comparing the results generated by changing the R_0 value (See figure 8-9-10), it's clear that our previous conclusion is not stable. The R_0 parameter has a significant impact on our conclusion. For example, the value of new cases per day at day 50 varies from 35 to 150, creating a very different situation.

2 Population generation

2.1 Investigating the influence of demography on epidemics

Generate populations for both regions, and investigate how the risk for an outbreak is different between the two regions

For each population, determine the percentage of runs that results in an outbreak. Which population has a higher chance of outbreaks occurring?

Figure 11 for region A and figure 12 for region B are both created using a low seeding rate (0.00000167) and a high number of days. You can see that there is a small difference between both regions. While in both regions the outbreak more or less behave the same, it's the amount of extinction cases that is interesting. For region A, the percentage of runs that results in an outbreak is around 80%. For region B, this percentage lowers to around 75% meaning region A has a higher chance of outbreaks occuring.

2.2 Vaccinating on campus

Generate a population where 60% of 18 to 26 year olds attend higher education. Use an age-dependent immunity profile, as supplied in lower student immunity.xml. In this scenario, 18 to 26 year olds are insufficiently vaccinated against measles.
Investigate what the effect would be if, a week after an individual infected with measles is introduced in the population, all students attending higher ed- ucation would be vaccinated. Compare this to a situation where no action is taken. To do this, you can write a call-back function and register it through the PyStride environment.

2.3 Is commuting to work important for disease spread?

Investigate the impact of the percentage of working persons that commutes to another city on the risk of outbreaks, and their size. Furthermore, examine whether the 'peak' of the epidemic - this is the day on which most newly infected cases are observed - is impacted by the number of commuting individuals in the population Generate different populations, varying the percentage of the population that commutes to another city for their work. Other parameters can be assumed to be similar to those in the run generate default.xml. Discuss the outbreak occurrence, sizes, and evolution over the different scenarios.

3 Performance profiling of sequential code

For this part of the assignment, we analyzed the code using GProf. 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, the GProf tool then uses this output to analyse the performance.
We analyzed the effect of the following parameters:

- Number of days
- Population size
- Immunity rate

- Seeding rate
- Contact log mode

In the following table, we can see the data of the runs with our different parameters versus the default scenario.

3.1 Default scenario

	population from file	generate population
population generation	13.33%	23.15%
simulation	14.17%	12.56%
total time	63.94s	58.78s

This result is the average of 10 runs, this will be the case for every parameter that is tested.

3.2 Number of days

10 days	population from file	generate population
population generation	34.61%	50.92%
simulation	8.59%	6.82%
total time	66.04s	73.29s

500 days	population from file	generate population
population generation	1.81%	3.33%
simulation	17.91%	19.47%
total time	78.74s	75.87s

As you can see, the number of days clearly plays a role in the distribution of the weight of the program. The mean of the running times of the simulator are surprisingly enough higher for every run, 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.

3.3 Population size

60K	population from file	generate population
population generation	N/A	28.38%
simulation	N/A	9.91%
total time	N/A	55.95s

6M	population from file	generate population
population generation	N/A	27.34%
simulation	N/A	11.48%
total time	N/A	56.52s

For this parameter, I 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. For this part, I only generated the population based on the configuration file, because it is made to have a varying parameter for the population.

3.4 Immunity rate

0.2 rate	population from file	generate population
population generation	12.86%	27.91%
simulation	13.08%	11.20%
total time	66.24s	56.07s

0.95 rate	population from file	generate population
population generation	11.80%	27.33%
simulation	13.17%	11.40%
total time	67.24s	56.53s

With the immunity rate, we have the same story as before, no big differences can be observed when we vary this parameter.

3.5 Seeding rate

0.02 rate	population from file	generate population
population generation	12.54%	20.71%
simulation	12.39%	10.94%
total time	67.07s	62.93s

0.0002 rate	population from file	generate population
population generation	14.96 %	27.23%
simulation	14.82%	13.94%
total time	62.23s	53.25s

The seeding rate also does not affect the running times or distribution of time used in the simulator.

3.6 Contact log mode

trace	population from file	generate population
population generation	9.76%	26.76%
simulation	11.11%	11.01%
total time	71.75s	58.06s

debug	population from file	generate population
population generation	11.1%	27.73%
simulation	13.6%	10.81%
total time	67.65s	57.75s

info	population from file	generate population
population generation	13.98%	28.97%
simulation	12.49%	10.48%
total time	64.95s	55.77s

warn	population from file	generate population
population generation	18.66%	27.25%
simulation	11.51%	10.7%
total time	61.76s	57.42s

error	population from file	generate population
population generation	15.2%	28.75%
simulation	13.89%	10.42%
total time	62.18s	55.86s

critical	population from file	generate population
population generation	15.22%	28.33%
simulation	12.86%	10.6%
total time	64.6s	56.5s

off	population from file	generate population
population generation	14.58%	27.62%
simulation	12.35%	11.17%
total time	65.91s	57.51s

Out of this data, we can see that very marginal speed gains are made when using little or no logging, but we do observe that the simulator can be slowed down by having a heavy log mode such as *trace*.

A Appendix A

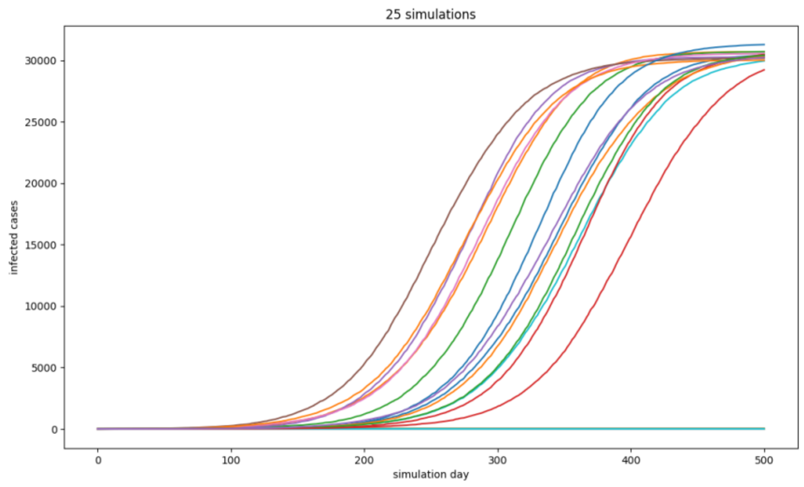

Figure 1: Simulation runs

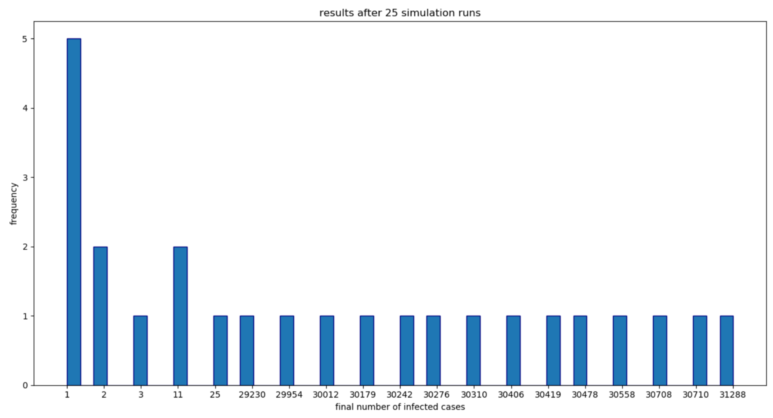

Figure 2: infected cases histogram

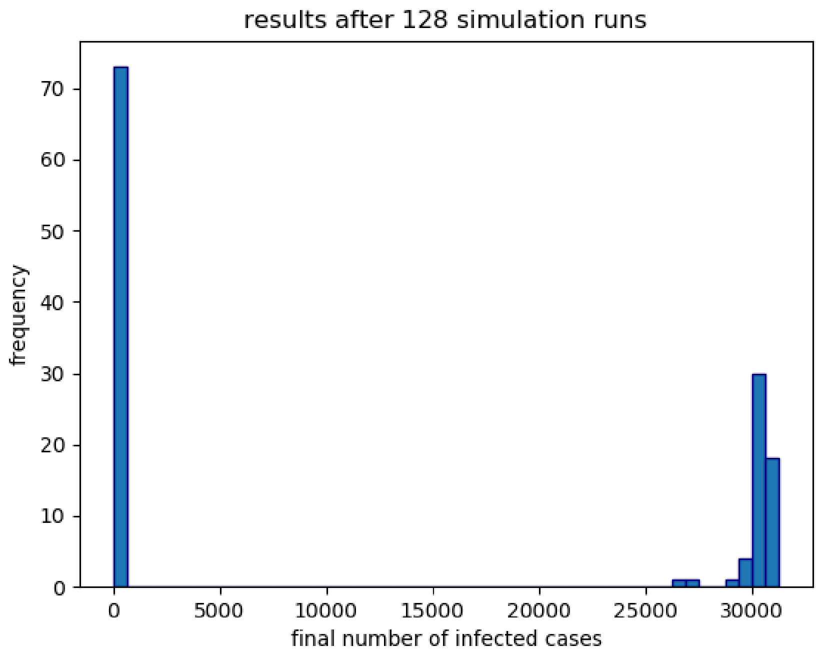

Figure 3: Infected cases histogram withs bins

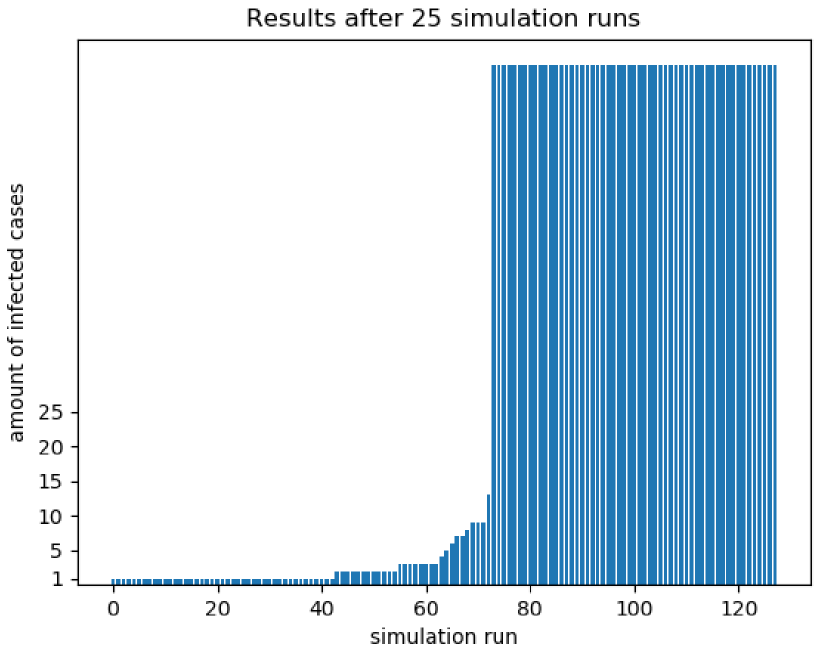

Figure 4: 128 Infected cases sorted

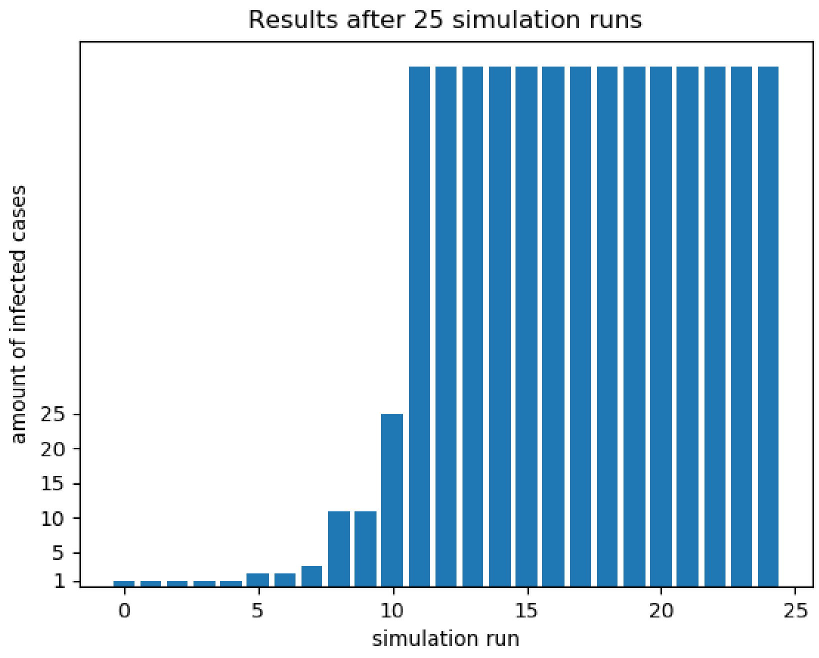

Figure 5: 25 Infected cases sorted

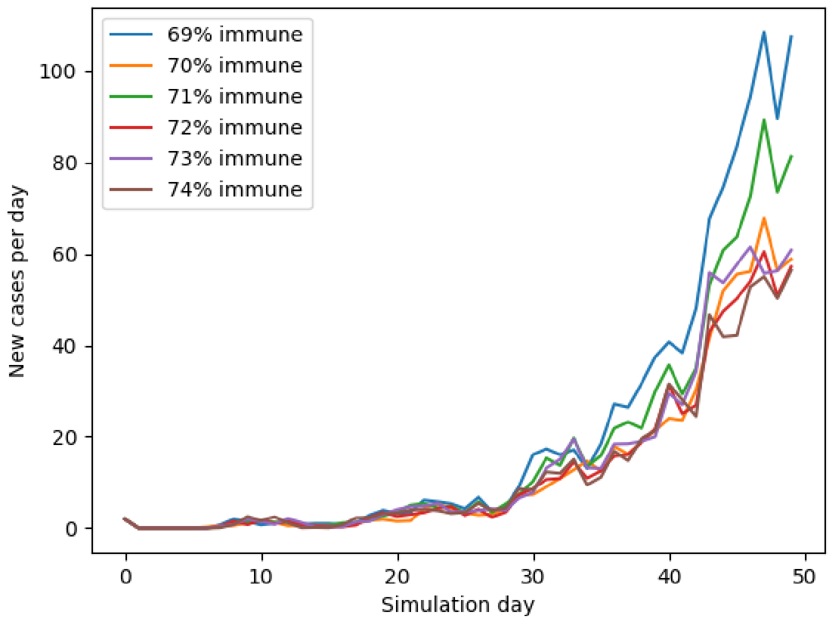

Figure 6: Different immunity levels

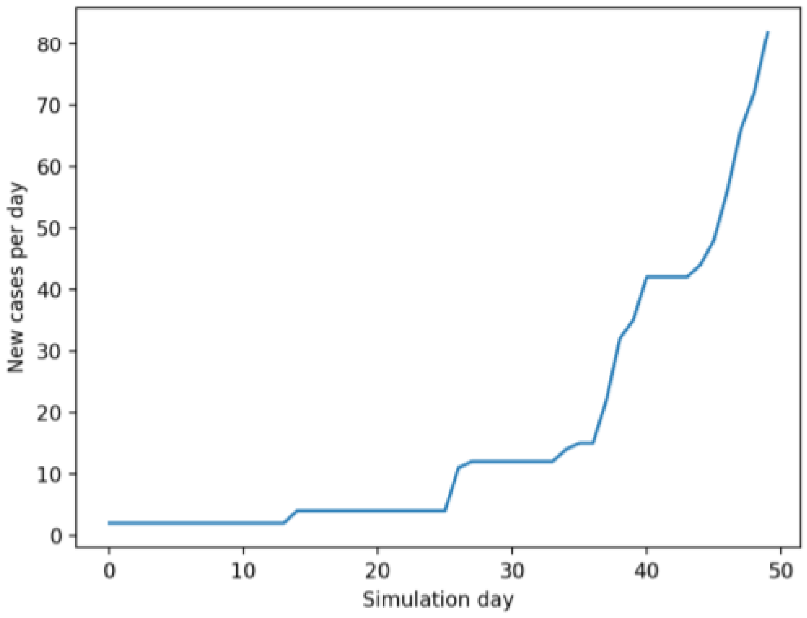

Figure 7: Plot given

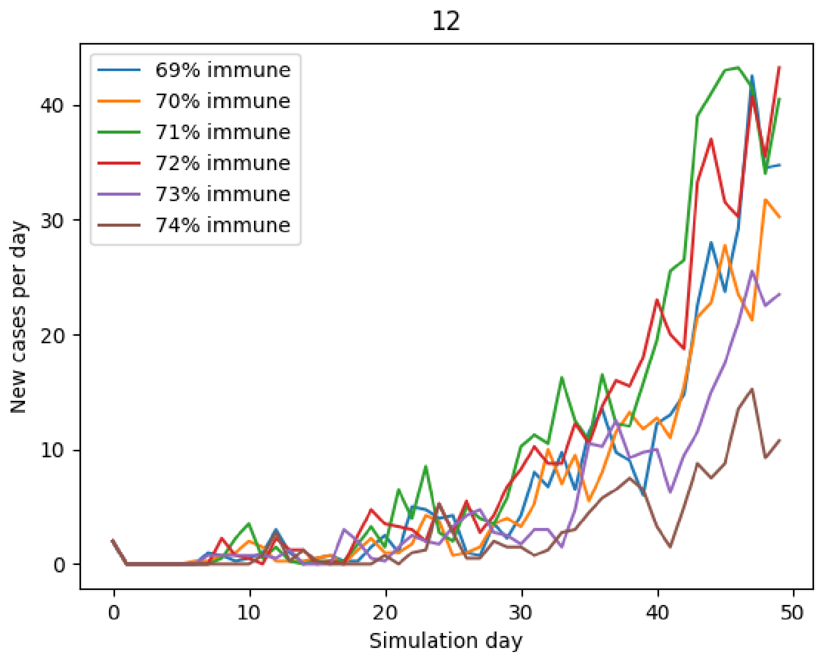

Figure 8: $R_0 = 12$

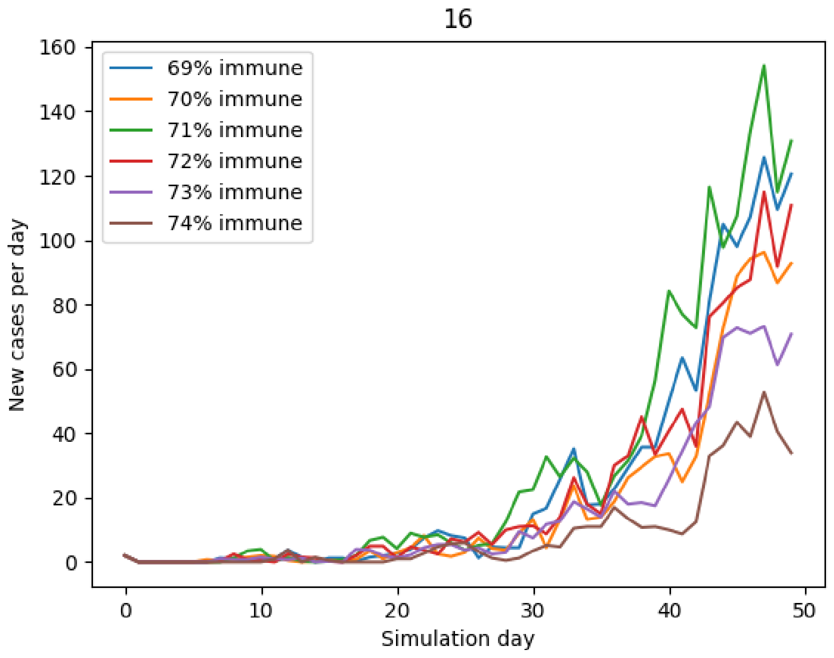

Figure 9: $R_0 = 16$

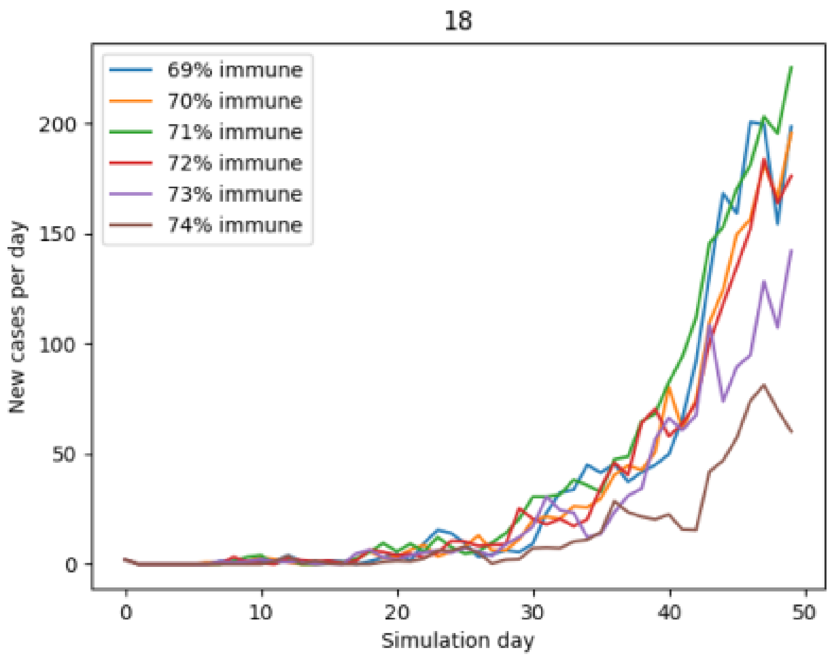

Figure 10: $R_0 = 18$

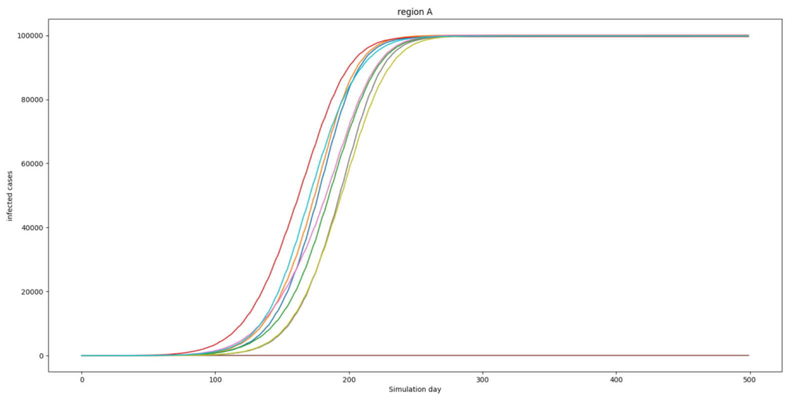

Figure 11: Region A

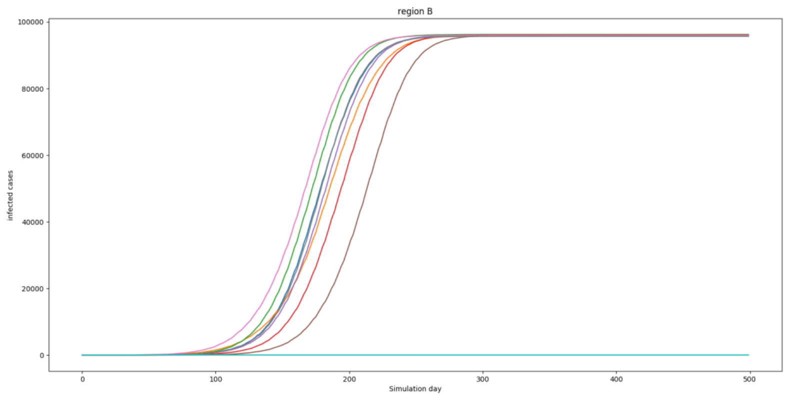

Figure 12: Region B