Stride*

Andrei Bondarenko, Mathias Ooms, Stan Schepers, and Laurens Van Damme

University of Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium

Abstract. This paper analyzes how parameters and data interact with the Stride simulation software and analyzes the performance.

Keywords: Stride · Epidemiology · Simulations.

1 Introduction

Stride (**S**imulate **tr**ansmission of **i**nfectious **d**is**e**ases) is epidemiological simulation software that can be used to examine how epidemiological diseases spread over a population in a given time. This paper analyzes how parameters interact with the Stride simulation software, examines how the different populations influence the outcome of Stride and analyzes the performance.
The purpose of this paper is to give perspective to future changes to the software. With the results the developer can compare his/her changes to a stable version.

2 Simulations

This section contains the results of the simulation exercises using Stride and our interpretation of these results. Simulations were run using the STAN and pyStride controllers.

2.1 Stochastic variation

Random numbers are used in Stride. These numbers will cause a stochastic variation in the results of the simulations.
After running multiple simulations, using 10 and 100 seeds, it seems that the chance has a big impact on the results. While keeping other parameters as seeding rate, r_0 etc. constant. The graph in figure 1 shows the two possible outcomes:

- Outbreak: The amount of infected people starts small but quickly starts to grow. Around 30000 people will be infected at the end.
- Extinction: A few people get infected (with a threshold of around 35 taken from figure 2) and the amount stays constant throughout the remaining time. As seen on figure 2.

* Supported by organization COMP.

Different persons can have very different contact profiles, resulting that the choice of the first infected people (seeding rate) determines the outbreak. For example if the chosen people are four retired persons who live alone at home and thus don't meet other people, the disease won't spread. In comparison with two persons that are going to work in big companies and thus will be in contact with lots of people, which will cause the disease to spread very fast. No other situations where, for example, only 10000 people were infected are present. With this configuration the simulations using 100 seeds had more or less 50 outbreaks every time. Therefore it's possible to say that there's a ± 50% chance for an outbreak or extinction.

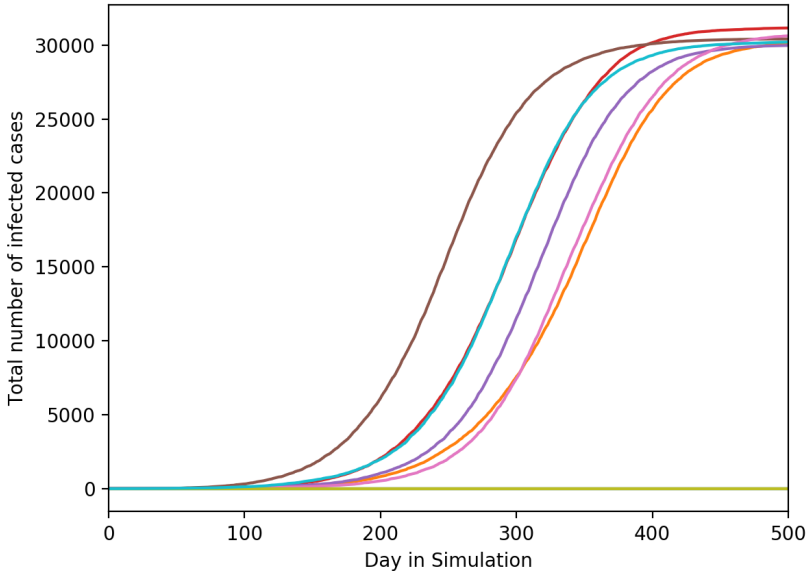

Fig. 1. Plot of cumulative cases per day of 10 simulations using 10 random seeds

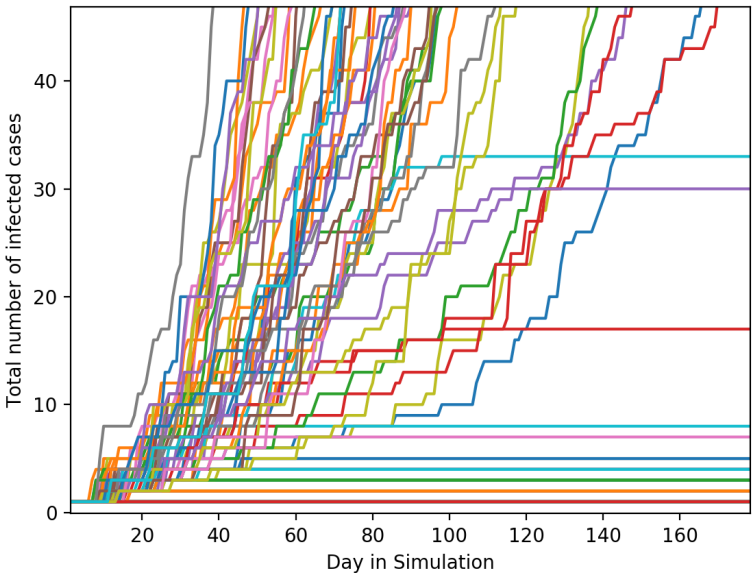

Fig. 2. Zoomed plot of cumulative cases per day of 100 simulations using 100 random seeds

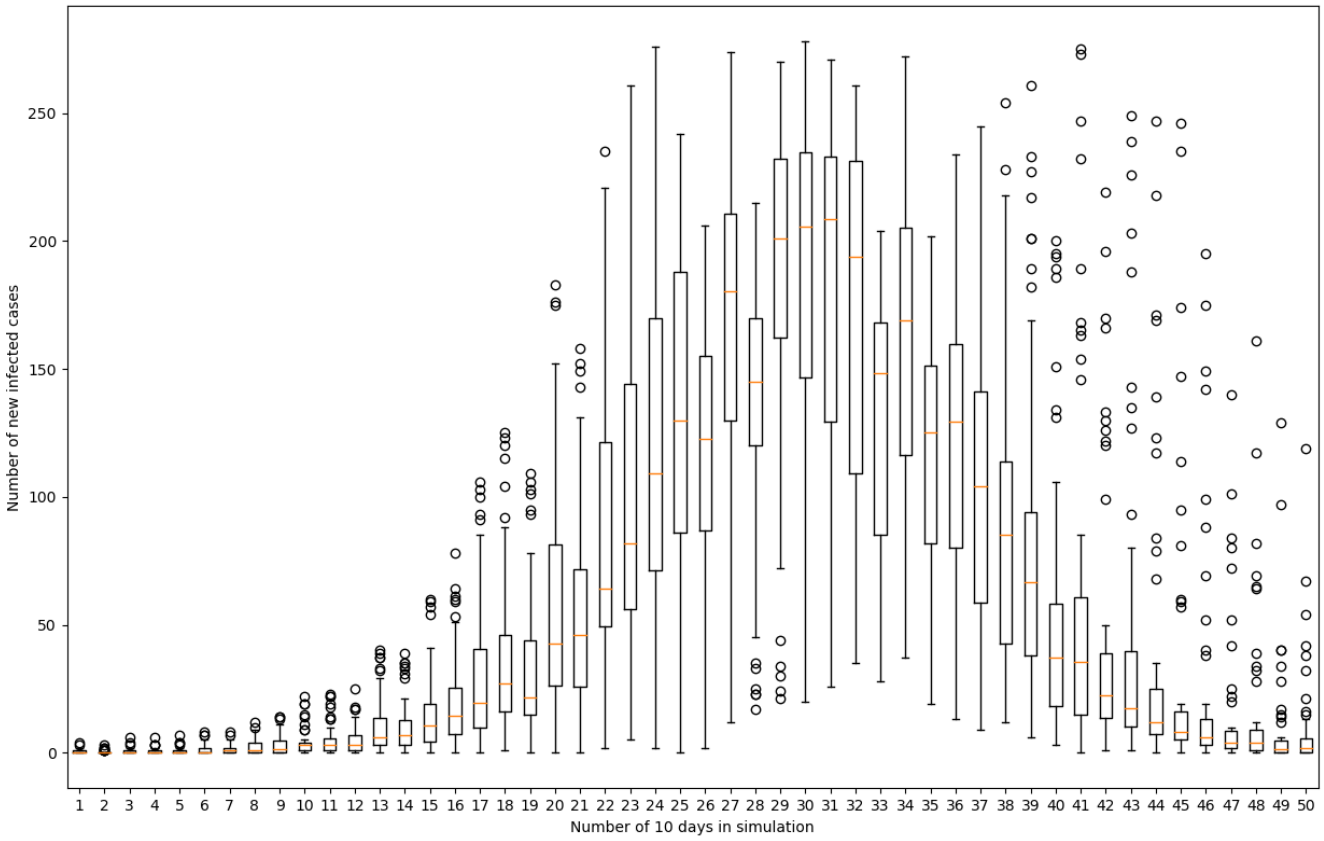

Fig. 3. Box plot sample of amount of new cases per day in 100 simulations using 100 random seeds, shown for every tenth day.

Plotting the number of new cases per day in a box plot graph of only the simulations where an outbreak is present gives too much information on a small graph which causes it to be chaotic. For this reason a sample is taken, as seen in figure 3, in which the measurements are only shown every tenth day.
The graph shows that the number of new cases per day follows a bell curve. It seems that the reason for this is the fact that at the beginning of an outbreak there are few infected people that can infect others. As time passes more people get infected and the number of new cases per day will thus increase as well, until

a point is reached where there are more people left with a lower chance of getting infected, thus resulting in a lower number of new cases per day. The outliers are data from simulations where the outbreak starts a bit earlier or later, this also causes the whiskers to be so big.

2.2 Determining an extinction threshold

Now that it's known that there are two possible outcomes of a simulation, extinction or outbreak, we can determine a threshold that determines which one of the two it is.
Plotting the final number of infected cases from the previous section in a histogram, results in figure 4. The two possible outcomes that could be seen in the previous section, can be seen here as well. Either the final frequency remains very low or it becomes very high. A very rough estimate of 10000 can be made for the extinction threshold based on this plot.

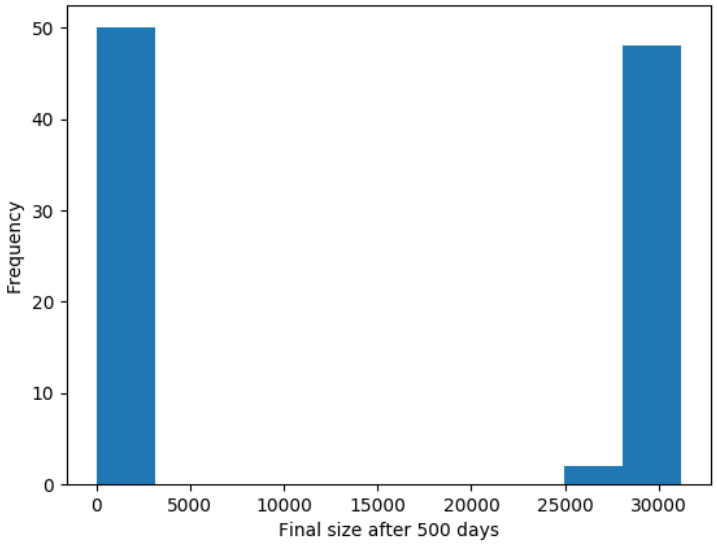

Fig. 4. Histogram of final number of infected cases after simulation of 500 days

2.3 Estimating the immunity level

To find the best estimate for the immunity level of the population the vaccination profile was set to *None* and the immunity profile to *Random*. Then for each immunity level, ranging from 60% to 80%, 100 simulations were run using different seeds, and the median was then plotted to compare with the given plot. An immunity level of 71%, bore the most resemblance to the given plot, as can be seen in figure 5.

The median was chosen over the mean because as can be seen in figure 6 the results produced by Stride can contain some distant outliers because of the stochastic character of its simulations. The presence of a very distant outlier can greatly affect the mean, while the median is less susceptible to this kind of influence by outliers.

So taking into account the stochasticity of the simulations it is quite safe to say that the immunity level of the given population is somewhere around 71%.

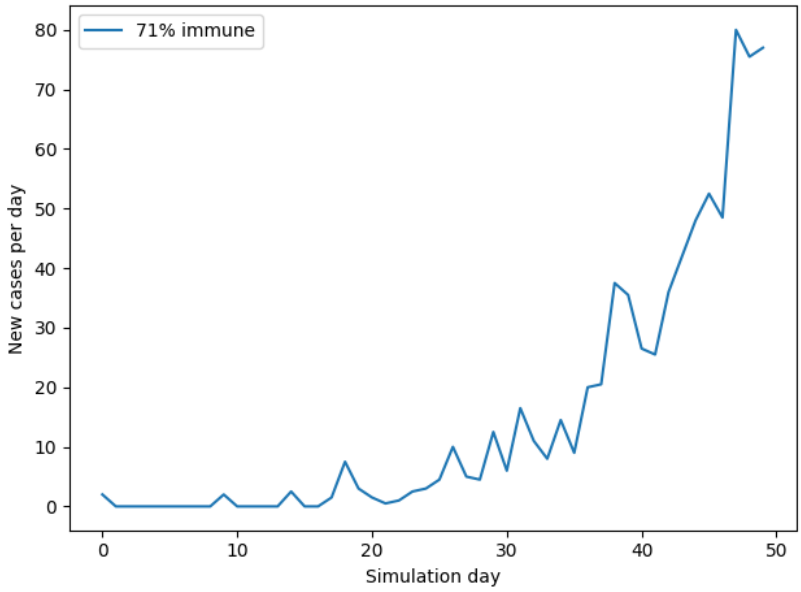

Fig. 5. New cases per day for immunity level of 71%, 100 seed median

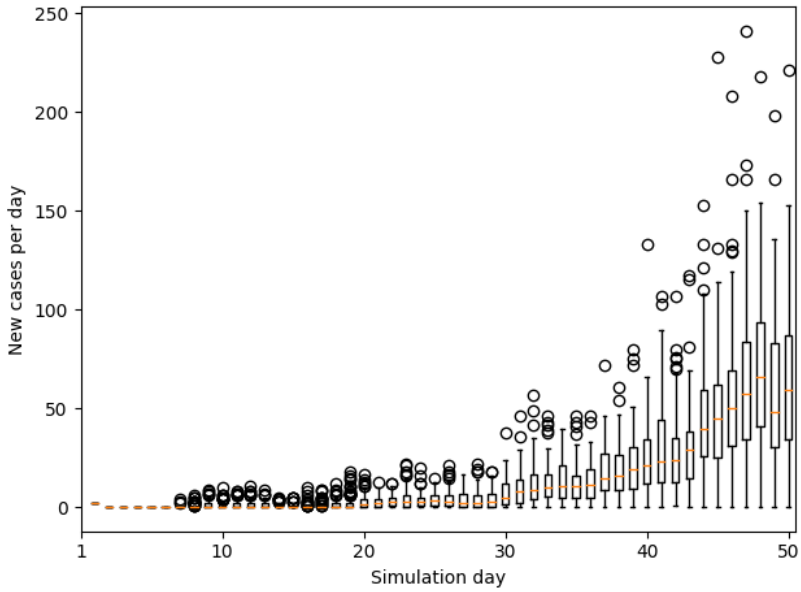

Fig. 6. New cases per day for immunity level of 71%, 100 seed boxplot per timestep

2.4 Estimating R_0

During the simulations in the previous section a fixed value of 15 was used for the basic reproduction number, R_0. To see whether or not the conclusion of that section is dependant upon this value, the immunity level was fixed at a good approximation, i.e. 71% (figure 5), and simulations were run for R_0 ranging from 12 to 18, a range used for the basic reproduction number of measles.

Plotting the results of those simulations in one graph resulted in figure 7, which clearly shows that the results of the simulations and thus the conclusion of the previous section are indeed dependant upon the value for R_0. There is a clear positive correlation between the value for R_0 and the speed at which the number of new cases per day increases.

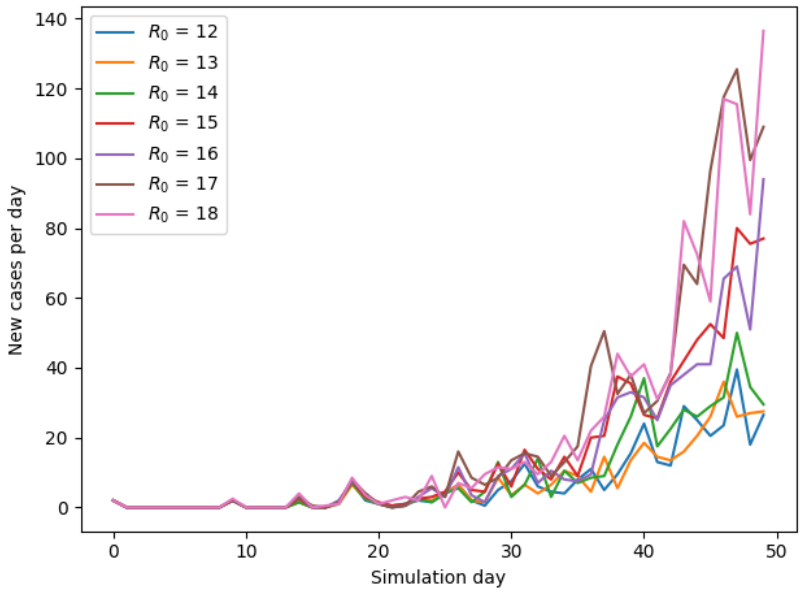

Fig. 7. New cases per day for 71% immunity level, 100 seed median for several R_0

3 Population generation

Simulation results can also depend on the population in which a simulation is run. The following sections describe how they do that.

3.1 The influence of demography on epidemics

Two populations were created, named Region A and Region B. Both are quite similar, but what is important to notice is the difference in age. The people in Region A are significantly younger than those in Region B as seen in figure 8. The results from running simulations on both populations show that in 95% of the cases, an outbreak (threshold = 5000) is present for Region A while Region B only shows outbreaks in 89% of the cases. So there is more chance of an outbreak in Region A with more younger people. Younger people are going out much more to several locations which causes them to meet much more people than older people who do not only commute less but are also more susceptible to diseases. This way the sickness can be spread more easily and that explains why the chance is higher in Region A.

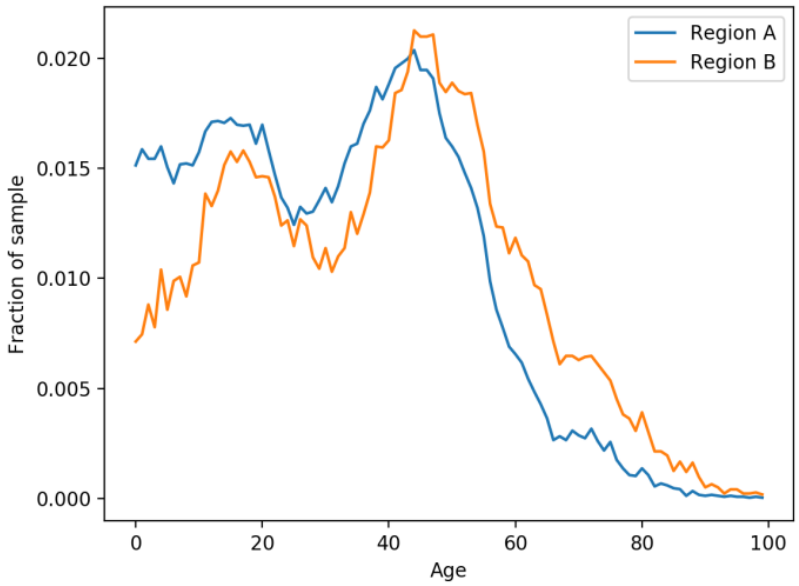

Fig. 8. Comparison of age distributions in household samples from region A and region B.

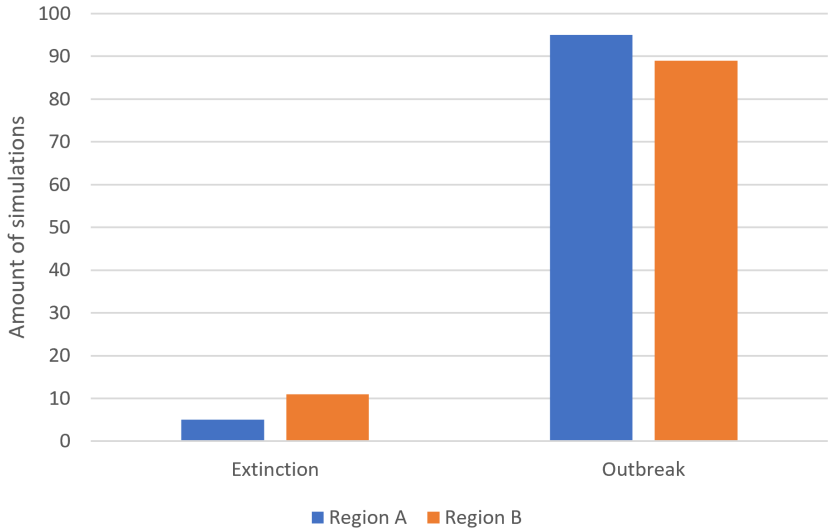

Fig. 9. Amount of outbreaks for both regions

3.2 Vaccinating on campus

One of the solutions to fight diseases is vaccination. So in this scenario we're looking to vaccinate people in time in order to prevent or minimize outbreaks. Students get vaccinated in the simulation after 7 days in comparison to a normal outbreak. It's interesting to create an average cumulative cases per day (of ten seeds) of these two situations, because by plotting the cumulative version we get a much better overview of the total number of infected as seen in figure 10. It's safe to use the average because while plotting the cumulative cases per day there won't be outliers, and if there are they won't differ much, which can influence the average.
The average number of cumulative cases per day when students are vaccinated is significantly lower then the average outbreak, causing the disease to spread slower and will result in a smaller outbreak. So now we can confirm firmly that vaccinating students after 7 days has a positive effect (less infected per day) when an outbreak is suspected.

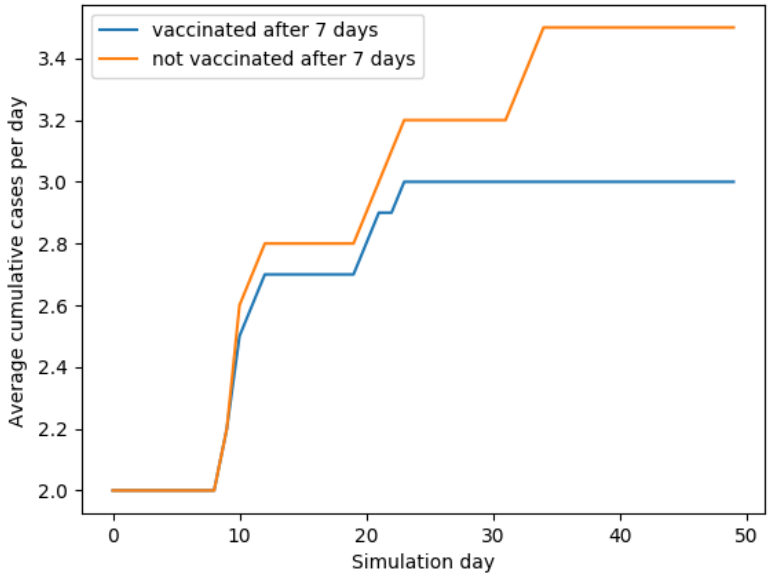

Fig. 10. The (10-seed) average cumulative cases per day plot of 10 simulations each, where students are vaccinated after 7 days

3.3 Commuting to work

When individuals commute to another city every day to attend their workplaces, this might have an impact on how diseases can spread. Here we will examine this impact of the commuting percentage.
We first tried to plot an average of the amount of new cases per day, but here it's quite difficult to notice a significant difference. A better result can be acquired by taking the average cumulative number of new cases per day (figure 11), this way we avoid the oscillations from the different random seeds.
It's noticeable there's a difference between the commuting levels, with the most significant one being between the 75% and the 100%. To put this in perspective we also calculated the same simulations over 500 days (figures 12 and 13). Here we see that a population that commutes for 100% has a major but quick outbreak, where a commuting level of 0% has a more slowly increasing outbreak spread over a larger period.
The intuitive interpretation of these results can be derived as follows: If nobody has to go to their work space, so assuming they stay in the same city, the new cases will rise slow. If you don't meet a lot of new people, then passing the disease to someone new is much more difficult. The higher the amount of people that have to commute to work, the faster pikes of new cases for a day will appear.
The peak-values differ for different commuting levels, which is visible in figure 12. However in the end, the total amount of people that get infected in each case stays the same (figure 13). Concluding us that the commuting level doesn't have an impact on the total number of infected people, but it has impact on the distribution of the number of new infected people per day.

The outbreak percentages for the five commuting levels (using a threshold of 100 cases): 0.00 = 98% 0.25 = 99% 0.50 = 100% 0.75 = 99% 1.00 = 99%

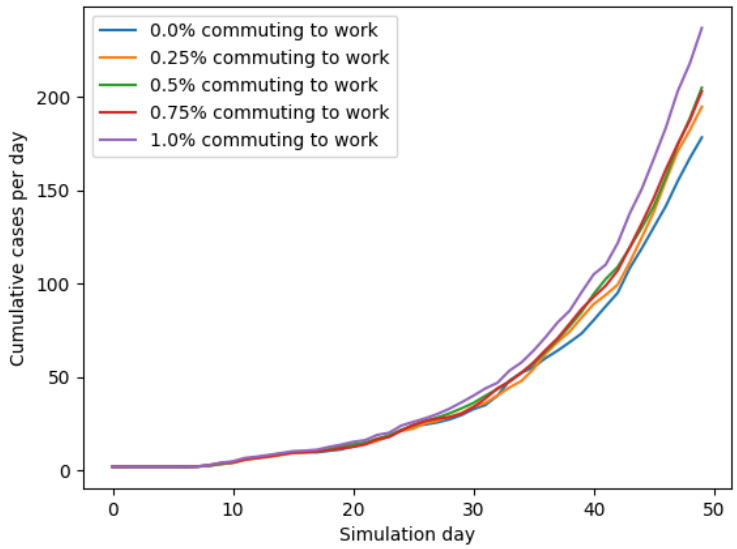

Fig. 11. Plots of the (10-seed) cumulative cases per day of 5 simulations with different commuting percentages over 50 days

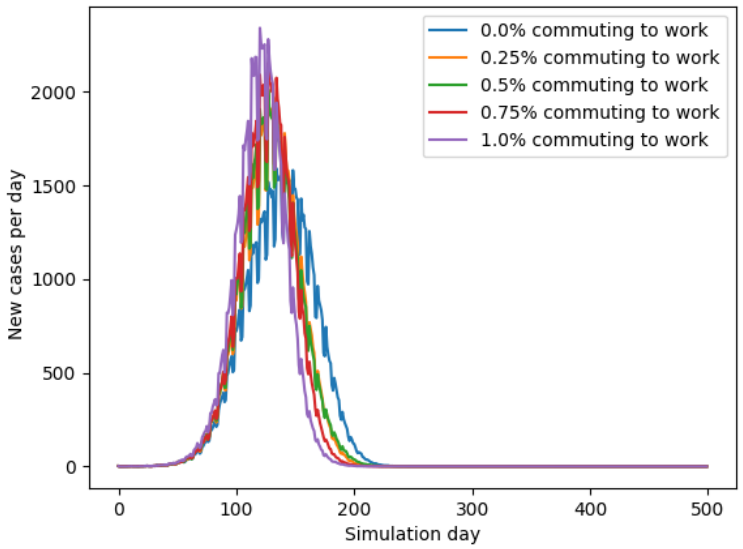

Fig. 12. Plots of the (10-seed) average new cases per day of 5 simulations with different commuting percentages over 500 days

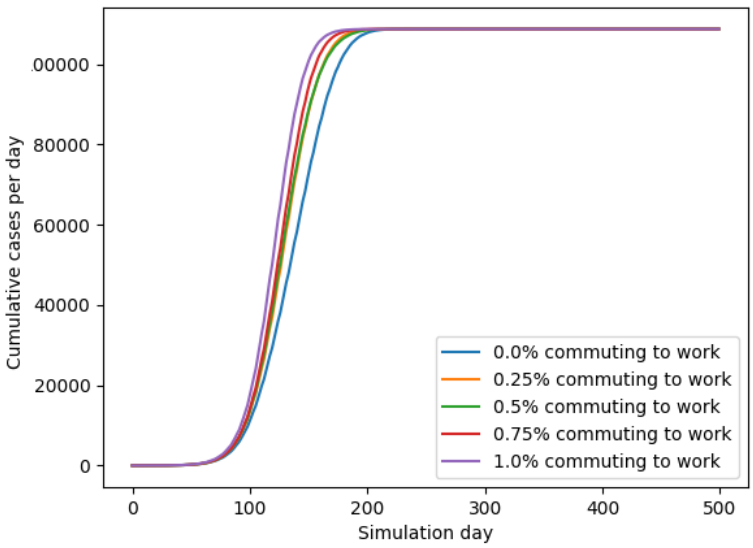

Fig. 13. Plots of the (10-seed) cumulative cases per day of 5 simulations with different commuting percentages over 500 days

4 Performance profiling of sequential code

In this section we will discuss the results from the performance profiling with different parameters. For the profiling we disabled OpenMP to get the performance of the sequential code. We run the default configuration and change only the parameter and measure the wall clock time. We don't evaluate absolute measurements. The results shown are averages from 3 tests so that they are not interfered with other processes on the computer.

4.1 Simulated Days

In figure 14 we see that the amount of days is linearly proportional to execution time. Simulating a day takes a constant time in Stride. This results in a linearly proportional correlation.

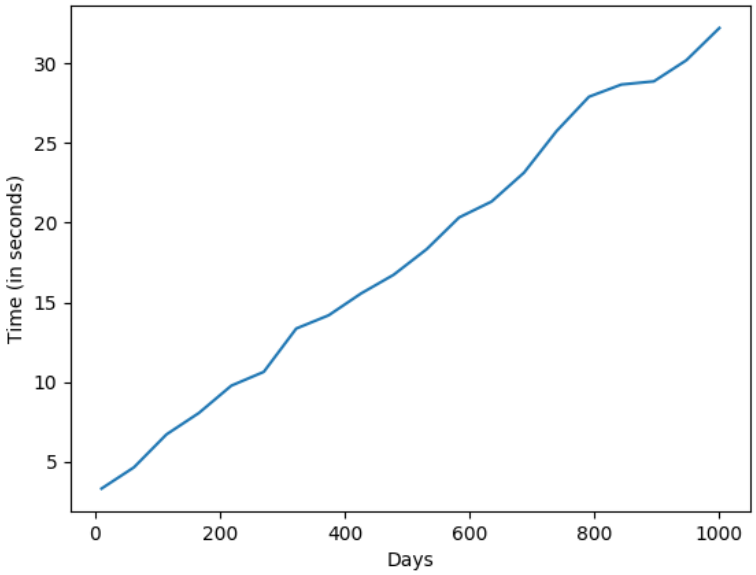

Fig. 14. Profiling plot days versus time

4.2 Population size

In figure 15 it also appears to be a linearly proportionality between the population size and time. Stride will loop over every (infected) person and his or hers contact pool. This will result in a linearly proportional correlation as seen in the figure 15.

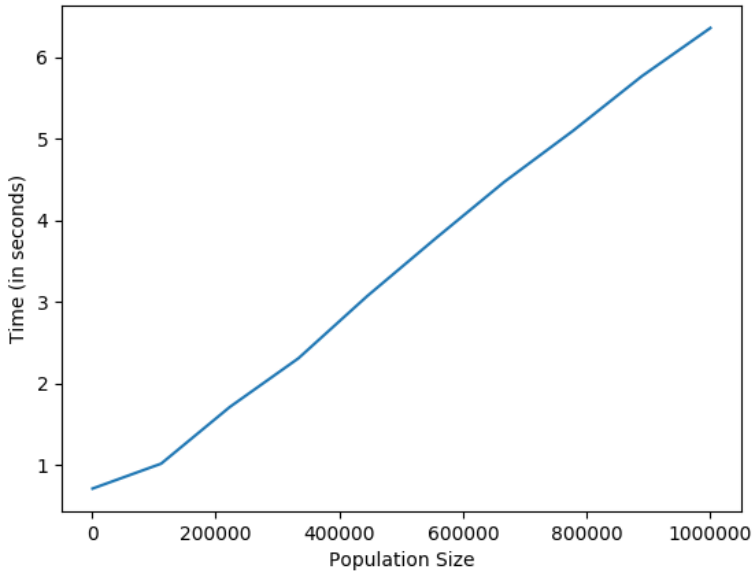

Fig. 15. Profiling plot population size versus time

4.3 Immunity rate

In figure 16 we see that the differences with all immunity rates doesn't have a trend and aren't large. Thus, the immunity rate doesn't affect the wall clock time of the simulation. The immunity rate doesn't affect the amount of loops were made and the amount of persons or days.

4.4 Seeding rate

In figure 17 we see small differences between the execution times so we can conclude that the seeding rate solely has no direct impact on the execution time.

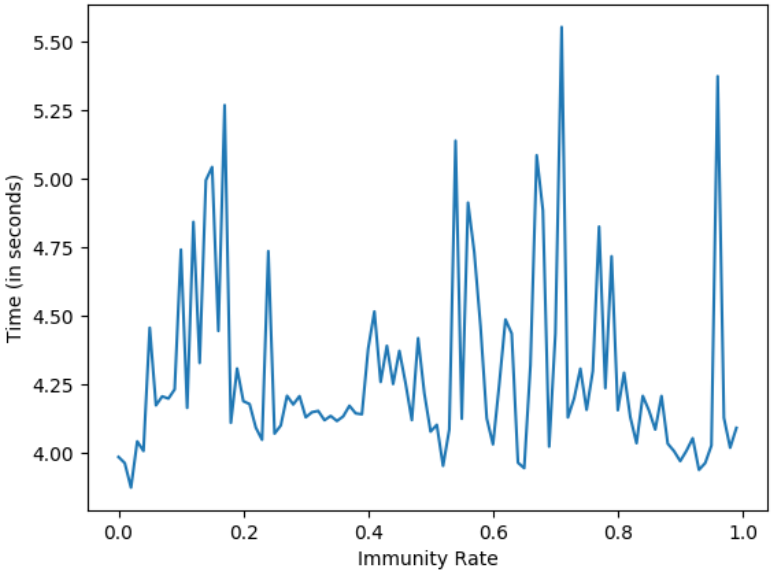

Fig. 16. Profiling plot immunity rate versus time

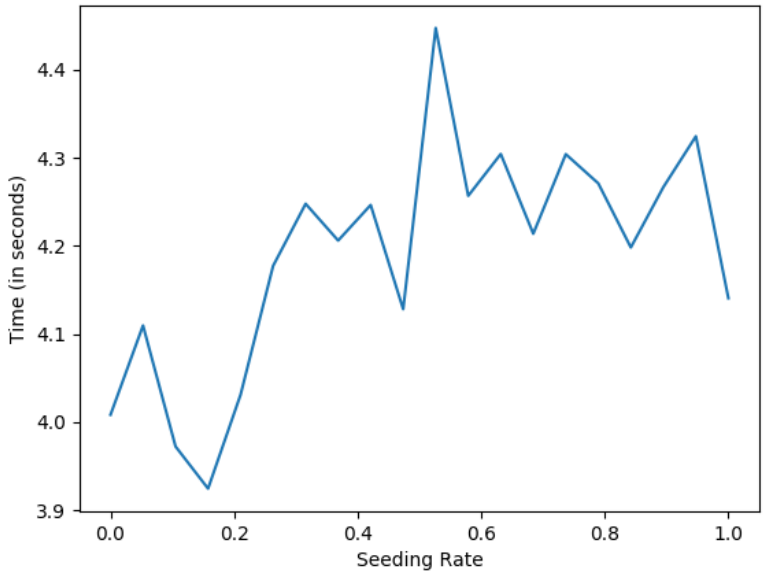

Fig. 17. Profiling plot seeding rate versus time

4.5 Contact log mode

Here we are going to monitor the time that is needed for different levels of contact log. In figure 18 we see that 'All' and 'Susceptibles' require more time for simulations. This is because the contact log modes 'All' and 'Susceptibles' use a algorithm that simulates all possible combinations instead of the more optimal algorithm that only simulates the interesting cases, cases that come in attention for get infected, used by 'None' and 'Transmissions'. An explantion can be found in the fact that 'All' and 'Susceptibles' will loop over all possible contacts and 'None' and only loop over 'Transmissions' will only the interesting ones.

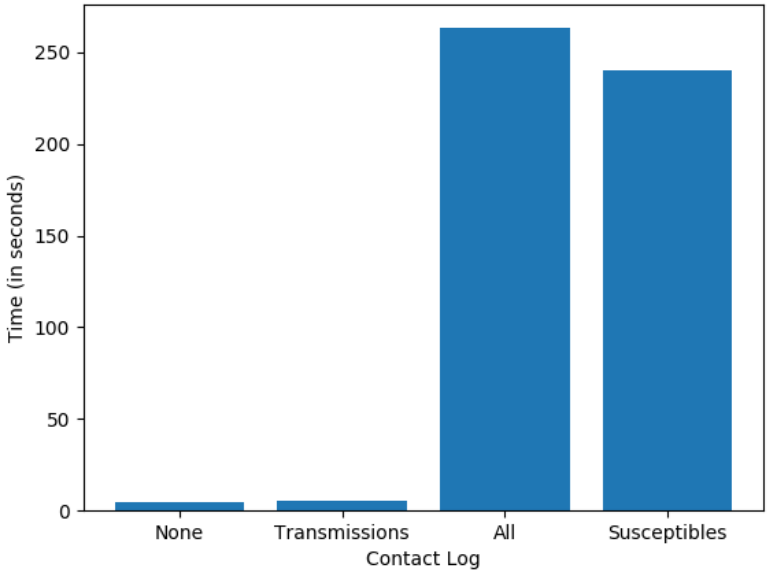

Fig. 18. Profiling plot time per type of contact log level

4.6 Call graph

In this section we try to see a overview of function timings so we can compare them later. We run Stride with the default settings found in $run\_default.xml$ configuration. For this tests parallelization is turned off. We give a top 20 table of functions which relatively take the longest. These 20 functions take about 7.43 seconds of the 14.88 seconds the simulation ran. They occupy almost 50% of the simulation time so they are a good indication of the whole simulation. The results were acquired with $gprof$, a profiling tool from GNU.

Name			
time (%)	time (s)	calls	ns/call
stride::Health::IsImmune() const			
13,68	2,04	23085278	88,3680066577496
stride::ContactPool::SortMembers()			
4,50	0,67	10360620	64,6679445824671
std::bitset<6ul>::reference::reference(std::bitset<6ul>&,unsigned long)			
3,97	0,59	150000000	3,93333333333333
std::_Base_bitset<1ul>::_S_whichbit(unsigned long)			
3,46	0,52	307366575	1,69179098280286
stride::ContactType::IdSubscriptArray<bool>::operator[] (stride::ContactType::Id)			
2,82	0,42	150000000	2,8
std::bitset<6ul>::reference::operator=(bool)			
2,62	0,39	150000000	2,6
std::_Base_bitset<1ul>::_S_maskbit(unsigned long)			
2,49	0,37	157366575	2,35119814992479
stride::Person::Update(bool,bool,bool)			
2,45	0,37	30000000	12,3333333333333
std::bitset<6ul>::operator[](unsigned long)			
1,95	0,29	150000000	1,93333333333333
stride::util::SVIterator<stride::Person,512ul,true,stride::Person const*,stride::Person const&,true>::operator++()			
1,48	0,22	61200000	3,59477124183007
stride::util::SVIterator<stride::Person,512ul,true,stride::Person const*,stride::Person const&,true>::operator*()			
1,48	0,22	61200000	3,59477124183007
stride::Population::GetInfectedCount() const			
1,48	0,22	102	2156862,74509804
stride::Health::IsInfected() const			
1,41	0,21	91200010	2,30263132646586
std::_Base_bitset<1ul>::_M_getword(unsigned long)			
1,31	0,20	150000000	1,33333333333333
boost::algorithm::detail::is_any_ofF<char>::is_any_ofF (boost::algorithm::detail::is_any_ofF<char>const&)			
1,01	0,15	30000040	4,99999333334222
stride::ContactType::ToSizeT(stride::ContactType::Id)			
0,97	0,15	198968934	0,753886533864628
unsigned int& std::forward<unsigned int&>(std::remove_reference<unsigned int&>::type&)			
0,67	0,10	56005212	1,78554810220163
__gnu_cxx::__atomic_add(int volatile*,int)			
0,67	0,10	10361106	9,65147929188255
stride::Infector<(stride::ContactLogMode::Id)1,false,true>::Exec (stride::ContactPool&,stride::AgeContactProfile const&,stride::TransmissionProfile const&,stride::ContactHandler&,unsigned short,std::shared_ptr<spdlog::logger>)			
0,67	0,1	10360620	9,65193202723389
stride::Health::IsSusceptible() const			
0,64	0,10	21404785	4,67185257875751
(total) 49,73	(total) 7,43		

5 Conclusion

Stride is capable of doing epidemiological research, in such a way that you can change certain parameters or arguments to run simulations and achieve statistical results. When running simulations, it's important to keep in mind that they run with random numbers. They cause a certain stochastic variation, which is also present and seen in the results generated by Stride. It is also possible to use Stride in the opposite direction. Determining the used parameters from given results.
While running simulations with different kinds of data, some conclusions about epidemic diseases could have been made. People that are younger in age, meet much more people in a day and are often more susceptible diseases. This causes diseases to spread faster but as a solution to this problem, we see that vaccination will slow down the spreading of the disease and it also causes less people to be infected at the end.
Looking at the performance of simulations, some parameters have a bigger influence than others. The number of days and persons have an effect on the time used to run the simulation. Also the contact log mode has also an impact on the performance. Parameters like immunity rate don't really change the performance.