

Project 2 breakdown

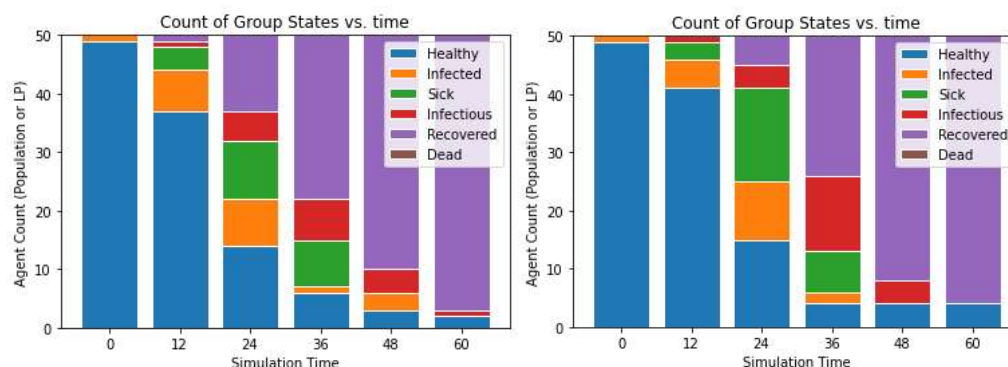
The `apl_sim_extra` program. This performs a simulation with several parameters. It is designed to imitate a disease with varying parameters. It then creates a graph of the varying states in the program at 6 evenly spaced times. The program can be altered with `r0` - the infection rate, `sick` - the percent infected that get sick, `infected` - the initial number infected, `length_time` - the length of the infection before recovered (can be cut short if the agent becomes sick, but `length_time` also affects the sickness length, and the length they are infectious after being sick before fully recovered). Once recovered, an agent is assumed immune. Also, once a person becomes sick, it is assumed they do not do anything and just stay home. The `r0` is only used for one `length_time`, so if someone has an infected and infectious period, they will likely overshoot the `r0` value.

Methods

The simulator can show a defined difference between several different diseases. Due to its simplicity, it will not be as accurate as many other simulators but may provide insight into comparing diseases. By modifying values such as `r0`, `sickness`, and allowing the simulator enough time to run to completion, attributes like 'herd immunity', or what consistent improvements to `r0` can do to slow the growth of a disease like CoVid. So, the main effects I would consider are `r0`, `sickness`, and time.

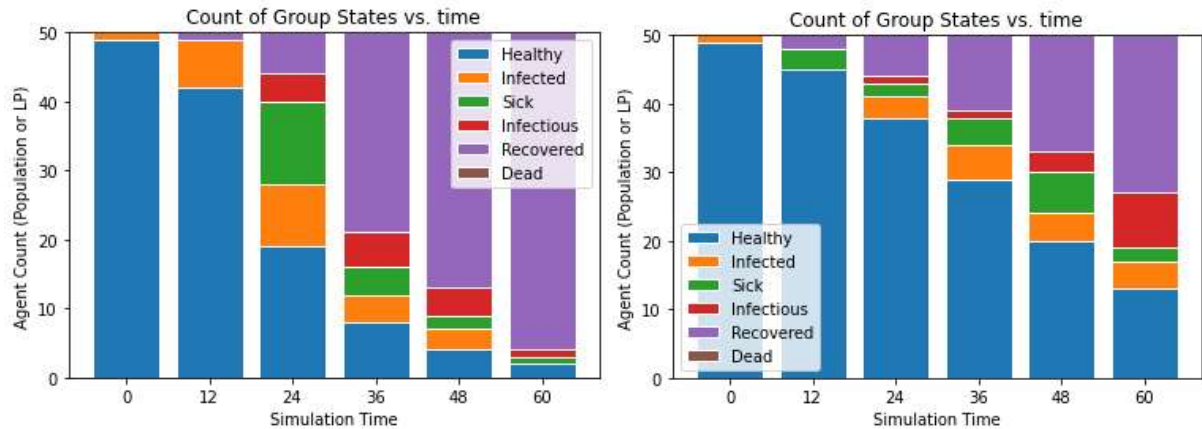
Starting with `r0` of 2.5¹, `sickness` of 60%, death rate of $0.025 \times 0.6 = 0.015$

`-r 42 -dr0 2.5 -dinfected 0.02 -dlength_time 7 -dsick 0.6 -p 50 -s 80` parameters. `-dr0 2` on right

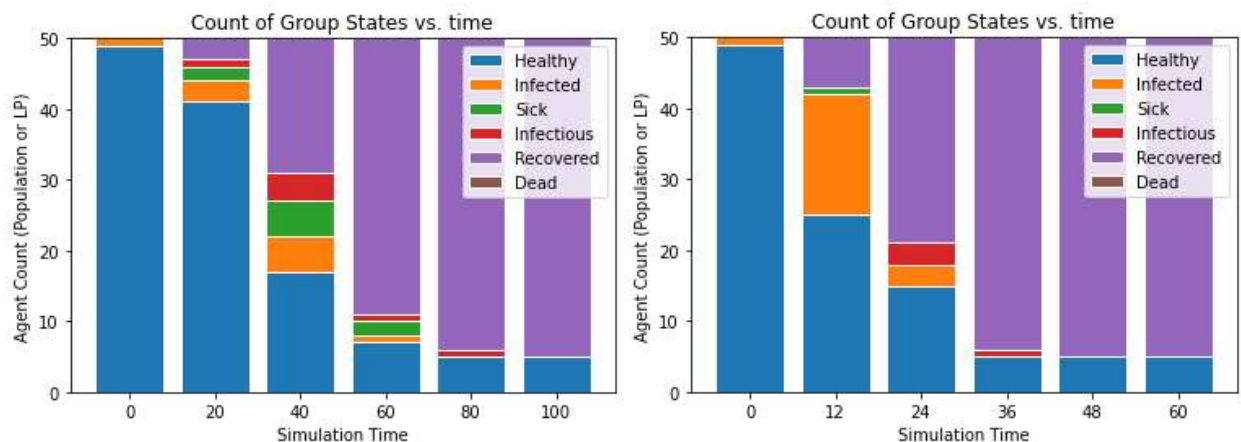


A difference in 2.5 to 2 does not produce a drastic change, but it does continually allow more time for people to respond and change to the sickness. Given many other factors like importance of hospitalization care in many cases this could be a drastic change. Given even other considerations like deaths are happening, and maybe the simulation of 'herd immunity' is not ideal, lowering the `r0` also shows an increased number of healthy people at all points in the graph. Surprisingly, this is nice for an instance where 'herd immunity' would make a difference. With a lower `r0`, it seems like a lower portion of the population need to be immune. This is more likely an interesting feature for vaccinations than important newsworthy viruses, but it is interesting.

¹ <https://www.cdc.gov/coronavirus/2019-ncov/hcp/planning-scenarios.html>

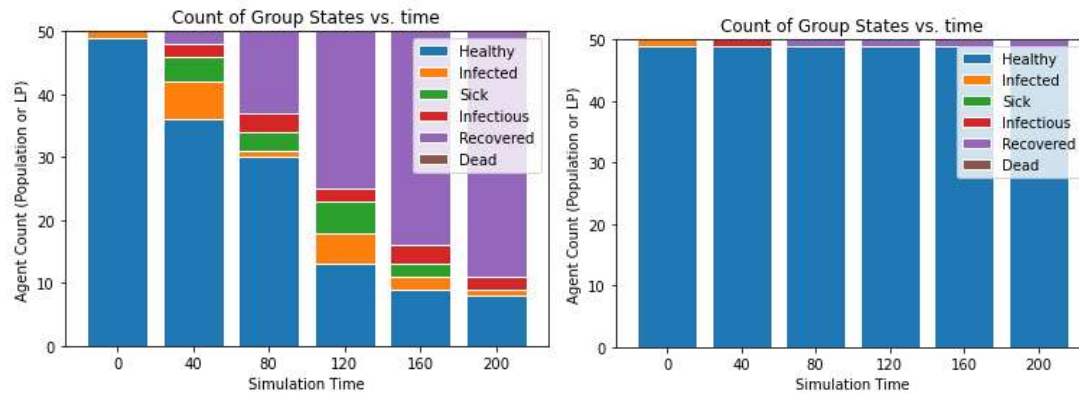


Reducing the r_0 value to 1.5 on this graph produces the graph above and to the left (-r 42). The graph on the right illustrates the huge random deviation spectrum (-r 72). So these insights may be indicative of a general trend, but there can be a large fluctuation between each simulation run. In general, it seems the lower the r_0 , the longer the simulation needs and the larger a portion of health population, but there can be a wide range from the random seed. A middle ground between the above to random factors at -r 60 and an increased simulation time of -s 100 allowed completion of the simulation and a slightly higher health population remaining afterwards.



Above to the right has a modified -dsick value (-r 60 -dr0 1.5 -dinfected 0.02 -dlength_time 5 -dsick 0.2 -p 50 -s 60) and it seems to vary widely, but there is much less change than expected. It has a much faster rate towards finish, but the end healthy result is very close to the non-modified dsick. Given the reduction in sickness rates and their ability to double r_0 , this was surprising.

Modifying parameters like time length also makes a large difference. With the parameter change to -dlength_time 20, when changing the simulation time to -s 200 to allow the simulation to get very close to finishing, you get a range of outputs. The infected size was very small, but at such a low chance for transmission, there were a few random values that achieved no pandemic (-r 10). But the below graph was much more of an average case (-r 60). Given the wide spread, it would be hard to indicate a direct trend, but from the programming (and r_0 change due to length), of course the time scale is lengthened, but there may be a slight increase in ending healthy population.



Caveats - this performs 'optimistically' well too far into the future and should probably have been styled as a stricter time-based program. Basically, further programming would be to either rollback if a state (infected) were assigned to sick or dead LP and send the infected message to a different agent. That said, this could be a start to looking into how high a population would need to be for 'herd immunity' to happen. Also, apparently without looking it up, and just using the simulation instead. This program could be further modified in a few ways: different disease vectors such as lyme disease where the disease is spread by an agent that never gets sick. Also, another version could use the same basis and create a polymer from a certain concentration of monomer molecules (simplistic version with likely no 'slowing' due to an increase in viscosity).

There is a form that includes hash values and dicts that allows for the death parameter to be used effectively, but that form also introduces a bug in the random generator, so values are not repeated. The file was included in the submission but was not included in this report.

Discussion

R_0 and length_time are both very interesting variables while keeping an understanding of the random seed and the simulation time. They can produce some insights into the simulation but can effectively change outcomes. R_0 has the largest effect and changing the R_0 through effects like masks, social distancing, even with a simple simulation is noticeable.