Simulating Epidemic Dynamics Using Network Diffusion Models

Ben Schmitt

I have always been interested in diseases caused by bacteria and viruses. Maybe because I find it fascinating that something so small can inflict so much damage on populations of much larger organisms. Or maybe because I'm worried about an epidemic decimating the human population and throwing civilization into chaos. The ongoing COVID 19 pandemic has piqued my interest in how human behavior can influence epidemic dynamics.

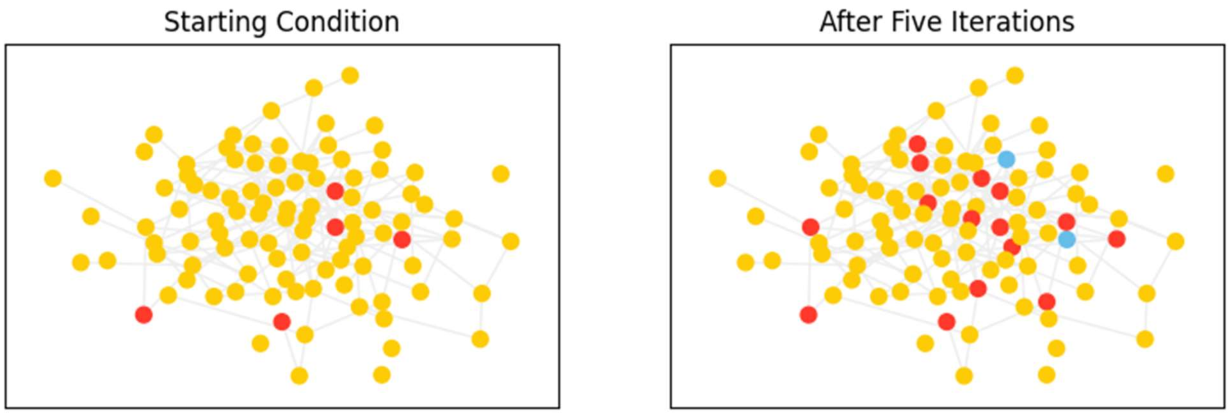
I decided to explore the impact of contact networks and intervention strategies on the dynamics of a simulated epidemic using the Python libraries NetworkX and NDlib. NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. NDlib is a Python package for simulating diffusion processes on complex networks.

Diffusion models are mathematical models used to understand the spread and propagation of epidemics within a population. These models help inform public health responses, guide decision-making, and contribute to effective control and prevention strategies during epidemics. Diffusion models are particularly useful for studying epidemics that exhibit spatial or temporal patterns and can provide insights into the dynamics of disease transmission.

Diffusion models can be used to evaluate the effectiveness of different intervention strategies in controlling the spread of epidemics. By simulating the impact of interventions such as vaccination campaigns, social distancing measures, or masking protocols, these models can help assess the potential outcomes of various control measures. This information guides decision-making by policymakers and public health officials in determining the most effective interventions to mitigate the epidemic.

Diffusion models often incorporate contact networks or social networks to analyze the dynamics of disease transmission. These models represent the social connections between individuals and the patterns of interactions that facilitate the spread of the infection. By considering the structure of the contact network, diffusion models can simulate the progression of the epidemic through the network and identify key individuals or groups that play a significant role in disease propagation. This knowledge is useful for implementing targeted interventions, such as contact tracing or isolating individuals with a high number of connections.

First let's set up a baseline scenario. There are 100 nodes (aka people). There are 300 edges (aka connections or relationships) between these nodes which are assigned randomly. We will assume the organism responsible for the epidemic infects nodes with a probability of 0.1 if they are exposed. We will also assume that once infected a person/node can die or recover from the infection and not be infected again for the duration of the simulation. The probability of an infected node being removed will be 0.05. Effectively, a node is only allowed to change its status from Susceptible (S) to Infected (I), then to Removed (R). This is known as an SIR model. Let's run this model for 5 iterations (we could think of an iteration as a unit of time like a day or week). We will start with 5 infected nodes assigned randomly.

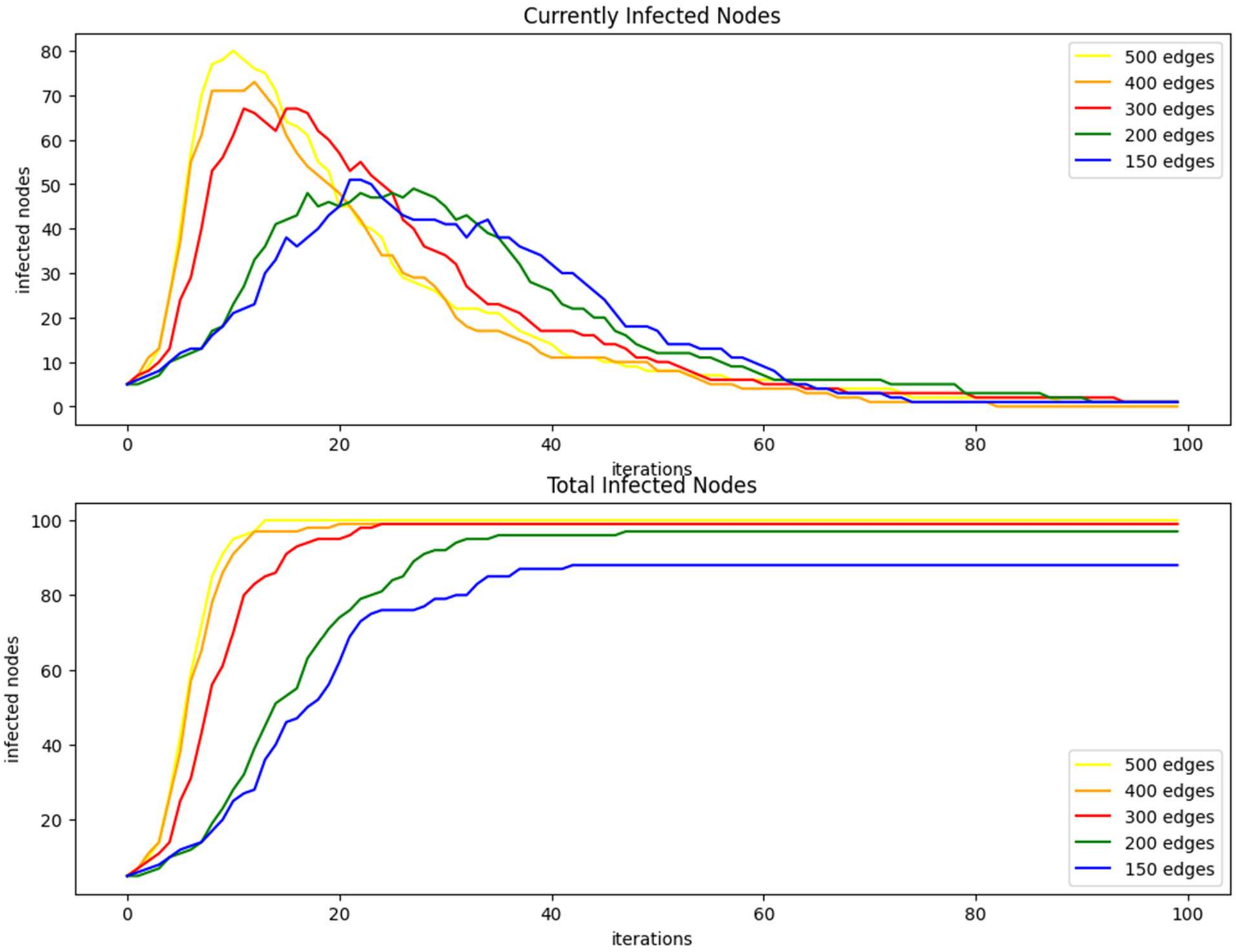


Starting Condition

After Five Iterations

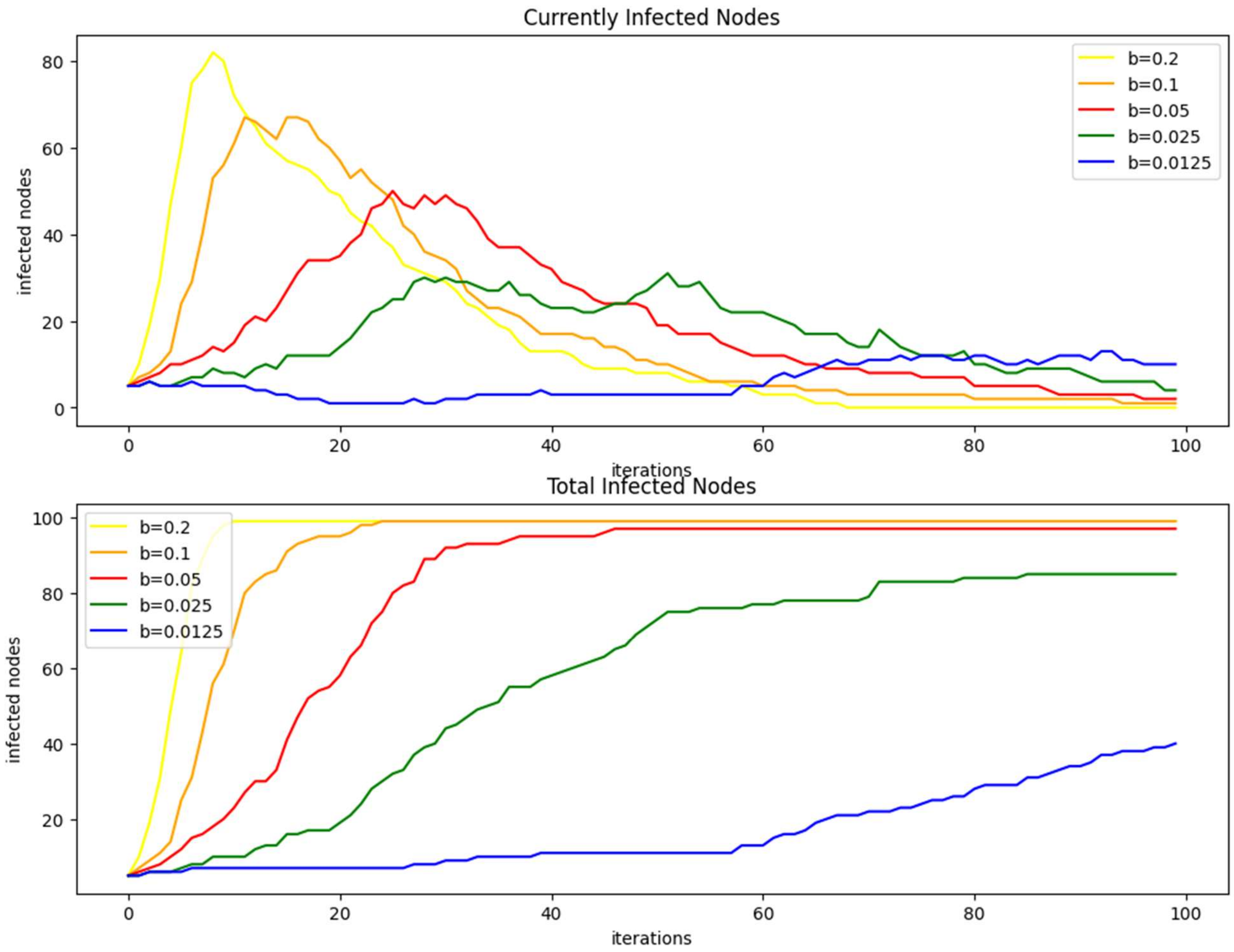
In the image above nodes are represented by colored dots and edges are represented by gray lines. We can see that after five iterations more nodes are infected (red color) and a few nodes have been removed (blue color).

Now let’s see what happens if we increase or decrease the number of edges between nodes. Decreasing the number of edges could represent the impact of social distancing policies. Increasing the number of edges could represent everyone partying because civilization as we know it is about to end. We’ll increase the number of iterations to 100 but keep the other parameters the same.



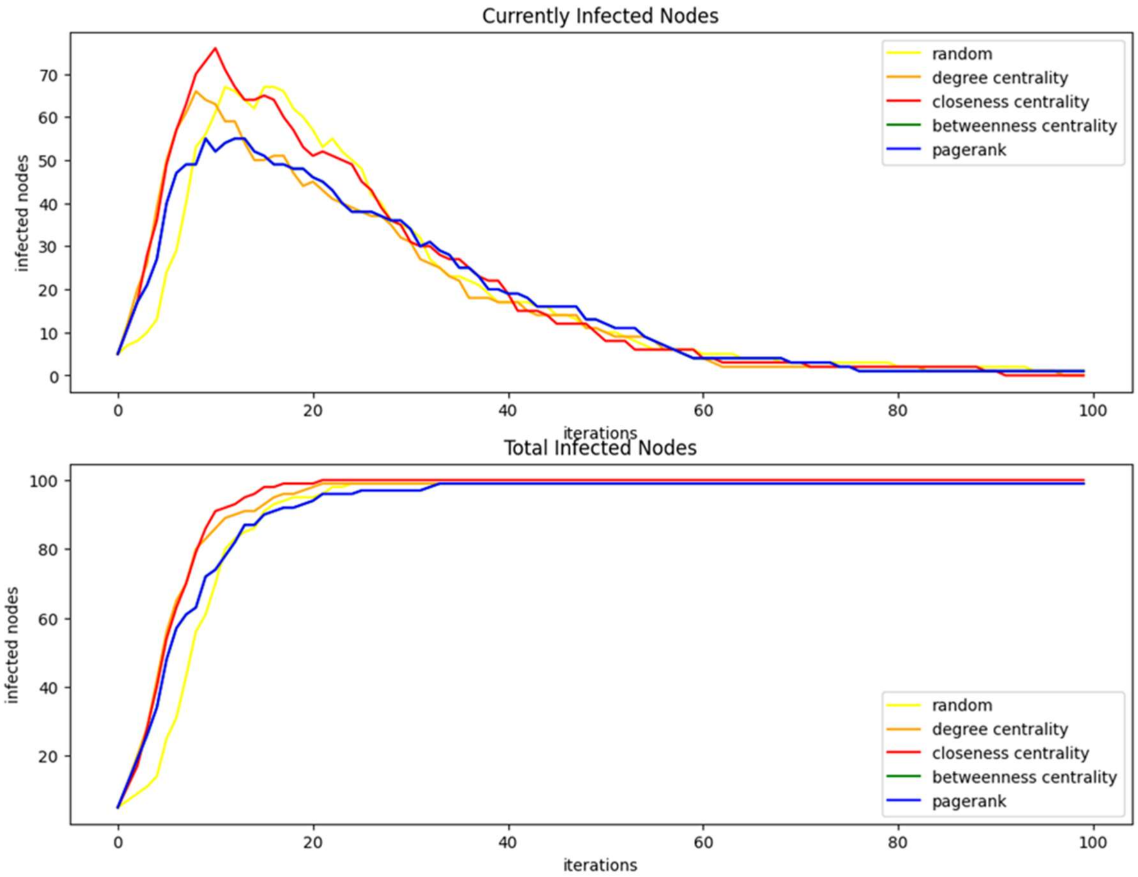
Here we can see that more edges resulted in a higher number of infected nodes earlier in the simulation. The 200 and 150 edge scenarios exhibited a much lower number of peak infected nodes and the peak occurred later in the simulation. Most nodes in the simulation still end up infected except in the very lowest edge scenario. Remember we have 100 nodes in the simulation and edges are assigned randomly. This means that in the lower edge scenarios there are likely nodes with no connections to other nodes.

Now let’s see what happens when we change the beta parameter. Remember that beta represents the probability a susceptible node will become infected if it encounters an infected node. This could represent everyone wearing masks. If everyone wore masks then beta would go down, assuming our simulated disease spread in the same was as COVID 19.



As you can see, changing the beta parameter has a large impact on peak infected nodes and total infections over the course of the simulation. Reducing the probability of infection seems like a good area to focus on if the goal is to slow the spread of an epidemic and reduce the total number infected.

Does it matter who starts out infected? Remember that so far, the starting infected nodes were assigned randomly. What if the starting infected nodes had more edge connections than most of the other nodes (high degree centrality)? What if the starting infected nodes were closer to more nodes in the network (high closeness centrality)? Let's look at what happens if we assign the initial infected nodes based on different measures of network importance instead of randomly.



It looks like by setting nodes with high measures of centrality as the initial infected nodes achieved peak infections earlier in the simulation as compared to random. However, each infected node assignment scenario resulted in almost all nodes being infected by iteration 20 to 30.