

# Contagion Modelling and Hospital Placement Optimization

Gulde: Prof. Avinash Bhardwaj  
Arushi Srivastava

**Abstract**—This report presents my project's findings on modelling of a contagion on a population and effective hospital placement to reduce the effect of the outbreak. It is vital to model disease prevention strategies and find the most optimised combination of resources to mitigate risks in real-life scenarios. For example, during COVID-19, understanding how to efficiently allocate hospital beds, implement targeted lockdowns, and roll out vaccination programs became essential to managing the spread. Modelling these strategies allows us to simulate different scenarios and assess the impact of each intervention, helping to determine the best course of action. By anticipating needs and outcomes, we can create well-informed protocols that can save lives, reduce strain on healthcare systems, and minimise socioeconomic disruption in the event of an outbreak.

## I.BACKGROUND & MOTIVATION

I was inspired to take this up as my topic when I researched about a game I had played quite a lot when I was young, called *Plague Inc.* *Plague Inc.* is a game where you create and evolve a disease to try and infect everyone on Earth. The challenge is to spread it as widely as possible without getting cured. You have to use strategy to optimise your limited resources: deciding whether to focus on how it spreads (like through air or water) or on symptoms that make it more deadly. You also need to adapt to different environments, like hot or cold countries, to keep infecting people effectively. It's all about making smart choices to keep your disease spreading and avoid detection. I was always interested on how to best the game, and had searched many different techniques back then on how to best the game, and win based on strategy and not on luck.

I knew I wanted to create an optimization model with similar parameters to this nostalgic game, where a disease is spreading hard and fast in its attempt to wipe out humanity. But with a twist. Instead of designing a more powerful contagion, I wanted to develop a model where the focus is on saving as many lives as possible in a fast-evolving environment. In my version, while the disease advances with new traits, the population responds with dynamic strategies for containment, treatment, and recovery. The goal shifts to

building resilience and optimising resources to counteract the outbreak, modelling how society can work strategically to protect and heal against the spread.



Fig. 1. Plague Inc.

## II. CONTAGION MODELLING

In my research, I examined the mathematical principles behind virus spread by studying the paper "Agent-Based Computational Epidemiological Modelling," published by the National Library of Medicine. This paper provided a detailed understanding of how agent-based models simulate individual interactions to capture the complex dynamics of disease transmission. By thoroughly reviewing this work, I gained insight into the mathematical foundations and computational techniques that drive these models, helping me to better understand and apply these principles in my own approach to modelling contagion and population responses.

### A.THEORY BEHIND AGENT-BASED COMPUTATIONAL EPIDEMIOLOGICAL MODELLING

When building a pandemic model, I learned that a big part of capturing disease spread is understanding and representing when people are in the same place at the same time. For example, we can generate a contact network by drawing an edge between two people if they share a location at the same time; we call this "co-location." This means that any time two people overlap in a space, such as being at the same workplace or even a grocery store at the same time, there's a potential path for the disease to spread between them. Over time, as people move between places, this network can grow and change, which creates a dynamic picture of how a disease could spread through a community.

To make this even more accurate, the model can vary the probability of transmission depending on the time two people spend together. For example, if two people are in the same room for just a few minutes, there's a lower chance of transmission than if they're co-located for an hour. The idea of time-varying edges, where the length and frequency of contacts are recorded, helps create a more realistic contact network that reflects real-life movement patterns and interaction times.

Once we've set up the co-location framework, we calculate the likelihood that a disease will actually spread from an infected person to a susceptible one. This is where the probability formula comes in:

$$p_{i \rightarrow j} = 1 - (1 - r_i s_j \rho)^t$$

Here's what each term means:

- $p_{i \rightarrow j}$ : the probability of transmission from infected person i to susceptible person j.
- t: the duration of exposure, meaning the time two people are in the same place.
- $r_i s_j$ : the infectivity of person i and the susceptibility of person j, respectively. These values capture individual characteristics, like how infectious or vulnerable someone is.
- $\rho$ : the transmissibility of the disease, which is a constant for the disease itself

This formula means that the longer people are exposed to each other, the higher the probability of infection, and individual factors—such as a higher infectivity due to poor hygiene, or higher susceptibility—play a key role in adjusting the overall risk.

Additionally, constructing the model involves creating a “synthetic population,” which is like a data-based representation of a community. This population is a “digital twin” that captures demographics like age, occupation, and location patterns to accurately represent how people in a real population might interact. Each person has their activity pattern, which includes the places they visit and the times of these visits, which helps simulate daily life interactions.

This network and probability setup allow us to explore questions about how a disease spreads. For instance:

- We can investigate if a harmful disease state might reach certain individuals based on their connections.
- We can see if there are conditions under which the disease reaches a steady state—where infection rates stabilise, which could mean it's no longer spreading or has become endemic.

After reading this paper and understanding the model, I realised how much complexity goes into accurately simulating disease spread. These insights help predict not just how a pandemic could unfold but also guide interventions, like where to focus isolation efforts or how long exposure times should be minimised to prevent further spread.

## B.SIMULATION OF BASE POPULATION

Initially, I began the simulation for the population in Python, setting up nodes and charts to track and infect individuals in each iteration. While this model was mathematically sound, it lacked a strong visual element that could provide a more intuitive understanding of virus spread dynamics. After our first few meetings, Prof. Avinash recommended moving the project to AnyLogic, which offered richer visualisation capabilities and could significantly enhance our grasp of the virus's transmission patterns.

I started with familiarising myself with the AnyLogic software and going through the trial videos as well as the IE630 slides, which I borrowed from a student.



**Fig. 2.** First Model on AnyLogic

I created various locations: **Houses, Schools, Offices, and Parks**. Each of these locations became points where agents (mostly children and adults) could interact, which was important for simulating how the virus could spread realistically in a structured setup. I created these locations as agent types with parameters such as number of children and adults.

Then, I focused on setting up **Children and Adults** as distinct agent types, giving them specific places to be during the day to simulate how they interact in real life. Here's how I did it:

- I defined Children and Adults as separate population types to reflect different types of agents and their roles in the simulation. This distinction lets me see how the infection moves through the population, given each group's unique habits, exposure, and parameters.
- Each agent is assigned to a **House** to create a

## IE688 Project Report

family-like setting, with a random 2-3 Children and Adults per house to keep things varied and realistic.

- I then set up **Schools** and **Parks** specifically for Children, while Adults were assigned **Offices**. This setup allowed Children and Adults to have both shared and unique environments throughout the day, adding depth to the interactions and infection opportunities.

Children's Daily Routine:

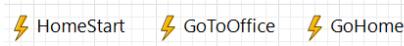
- Home*: 12 AM - 8 AM
- School*: 8 AM - 3 PM
- Park*: 3 PM - 5 PM
- Home*: 5 PM - 12 AM



**Fig. 3.** Children's Daily Routine

Adults' Daily Routine:

- Home*: 12 AM - 8 AM
- Office*: 8 AM - 6 PM
- Home*: 6 PM - 12 AM



**Fig. 4.** Adults' Daily Routine

I used timeout cyclic events to trigger the movement from one location to another, but a StateChart inside AnyLogic could also be used. However, that particular feature was not triggering the event I needed as I had not enabled this feature in creating the agents, hence I used events

After this, I created functions, parameters, and variables to count and update things such as the connection duration between two agents

```
double currentTime = time() % 24;
for (Child child1 : children) {
    for (Child child2 : children) {
        if (child1 != child2) {
            // Check if both children are at school at the same time
            if (currentTime >= 8 && currentTime <= 15) { // 8 AM to 3 PM in minutes
                if (child1.School == child2.School) {
                    child1.connectionDurations.put(child2, child1.connectionDurations.getOrDefault(child2, 0.0) + 1);
                    child2.connectionDurations.put(child1, child2.connectionDurations.getOrDefault(child1, 0.0) + 1);
                }
            }
        }
    }
}
```

### III. INTRODUCTION OF CONTAGION AND ITS SPREAD

To model the spread of infection, I started by setting up a variable to control the initial infection rate across the

population. For now, I fixed this rate at 5% of the total population, meaning that 5% of agents start out infected.

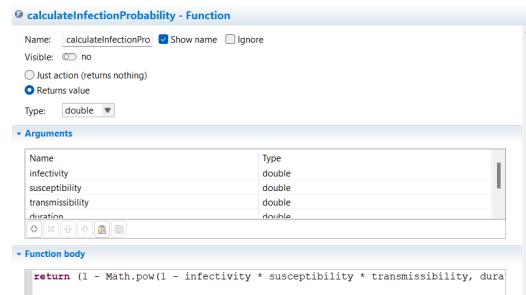
Next, I created a function to calculate the probability of infection spreading from one agent to another. Using this function, each agent's state could change from "susceptible" to "infected" based on specific criteria. When an agent was in contact with an infected individual, I set the probability threshold for infection to occur at 0.5 (though this can be adjusted). If the infection probability exceeded 0.5 during the interaction, the agent would become infected.

```
for (Child child1 : children) {
    if (child1.currentState.equals("Susceptible")) { // Susceptible child
        for (Map.EntryAgent, Double> entry :
            child1.connectionDurations.entrySet()) {
            Agent otherAgent = entry.getKey();
            double connectionDuration = entry.getValue();
            // Check if the other agent is infected (either Child or Adult)
            if (otherAgent instanceof Child &&
                ((Child)otherAgent).currentState.equals("Infected") || 
                otherAgent instanceof Adult &&
                ((Adult)otherAgent).currentState.equals("Infected")) {

                // Fetch infectivity and susceptibility correctly
                double infectivity = ((Child)otherAgent).infectivity;
                double susceptibility = child1.susceptibility;

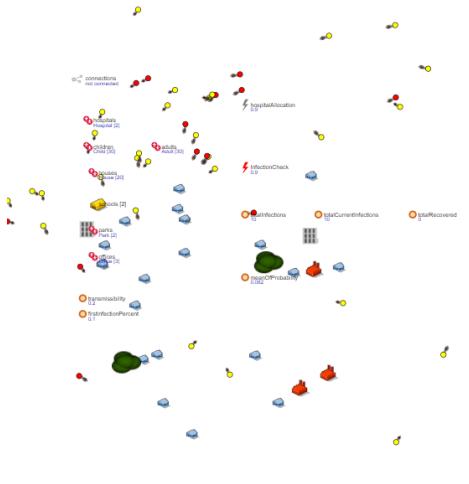
                // Calculate infection probability
                double infectionProbability =
                    calculateInfectionProbability(infectivity, susceptibility,
                        transmissibility, connectionDuration);
                infectionProbabilities.put(child1,
                    infectionProbability);
                if (infectionProbability >= 0.1) {
                    child1.currentState = "Infected";
                    child1.Marker.setFillColor(Color.RED);
                    totalInfections++;
                    break;
                }
            }
        }
    }
}
```

Similarly, this check was done between adult-adult connections as well as child-adult connections



**Fig. 5.** Infection Probability Calculating Function

I then added variables that would be calculating the number of infected and recovered (will come later) agents, counters for each individual and how long they have been infected, etc to help in further calculations in the model. This would later help in creating conditions for hospital allocation, policy building, etc.



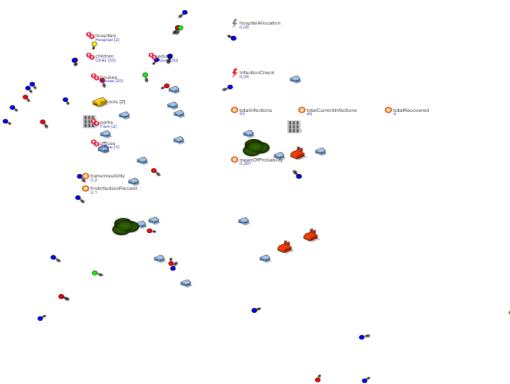
**Fig. 6.** Spread of Disease

For the states, I chose the following colour code:

Yellow- Susceptible  
Red- Infected  
Blue - Recovering  
Green - Recovered

#### IV. INTRODUCTION OF RECOVERY METHODS (HOSPITALS, POLICIES, RESTRICTIONS)

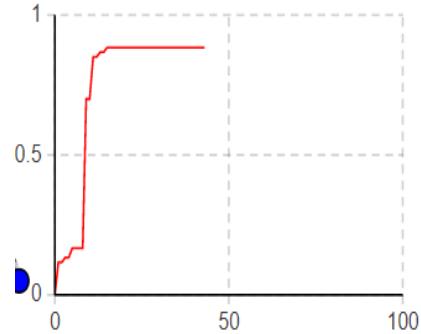
Next, I introduced hospitals into the model by creating a dedicated agent type, *Hospital*, with parameters like *capacity* and *currentPatients*. This setup allowed agents to be admitted based on specific conditions, such as meeting a minimum infection duration threshold. I added two hospitals for this model, with patients directed to one or the other depending on factors such as distance, available capacity, and the infection duration of the agent. Each patient's recovery time was randomised between 4 and 10 days to simulate variability in response to treatment. This structure enabled a dynamic healthcare simulation, using tailored variables for each agent and hospital to realistically model hospital admissions and recoveries.



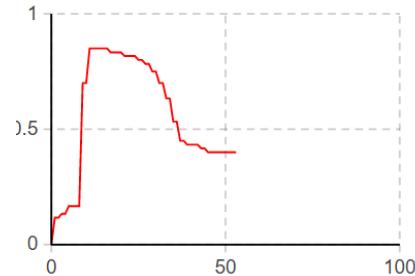
**Fig. 7.** Patients Recovering through Hospitals

Next, I used charts to analyse the impact of introducing

hospitals on the population. By adjusting various parameters, I observed how these changes affected the graphs representing the total number of infected individuals (over time) and the number of currently infected people (excluding those who have recovered). This allowed me to visualise the effects of hospital interventions on disease spread and recovery dynamics.



**Fig. 8 .** Overall Infected Agents/ No. Of Agents over Time



**Fig. 9 .** Infected People at an Instant of Time/ No. of Agents

As hospitals are placed within the model, and with the given constraints in place, infected individuals begin to be admitted for treatment. Over time, the total number of infected agents in the system reaches a point of stabilisation. As a result, the number of newly infected individuals decreases, and the overall infection count gradually declines.

Current settings:

1. Houses have a capacity of maximum 3 children and 3 adults
2. Children are distributed over two schools and two parks
3. Adults are distributed over two offices
4. The susceptibility and infectivity of every agent is a random number 0 and 1
5. The transmissibility of the virus is 0.3

#### V. NEXT STEPS

To improve hospital placement in my model, I'll be using the **Warehouse Location Problem (WLP)**. This method helps find the most cost-effective locations for facilities while meeting the needs of the population. In my case, the

## IE688 Project Report

“facilities” are hospitals, and the “needs” come from the infected individuals in various locations like houses, schools, and offices.

How I'll implement it:

1. I'll use binary variables to decide whether or not to place a hospital at a given location. These will be set to 1 if a hospital is placed there, and 0 if not.
2. The demand will depend on the number of infected individuals in each area. As the disease spreads, I'll update the demand dynamically.
3. The goal is to minimise the total cost of building and operating hospitals, which includes:
  - o Fixed costs for setting up the hospitals.
  - o Variable costs based on the number of patients treated.
4. I'll ensure that hospitals are placed close enough to infected individuals to minimise treatment time. I'll use simple distance calculations to make sure people can easily reach the hospitals.
5. Several constraints will guide the optimization:
  - o Every infected person needs access to treatment.
  - o Hospital capacities must not be exceeded.
  - o I'll set a budget limit for hospital construction and operation.

For the implementation of this optimization problem, I'll use AnyLogic's Optimisation Block. I will familiarise myself with this particular module over the next semester, and will also look into optimisation of policies including lockdowns, how it affects the spread of the virus, etc.

### REFERENCES

- [1] [Agent-Based Computational Epidemiological Modeling - PMCL](#)
- [2] [Tutorials | AnyLogic Help](#)
- [3] IE630 Slides PDF
- [4] [Optimal Warehouse Layout and Routing to Solve the Storage Location Assignment Problem – AnyLogic Simulation Software](#)