# Lab16

Ali Ghasemi - S289223

## INTRODUCTION

I. A Hawkes process is a type of point process that is commonly used to model the timing of events or activities. A point process is a mathematical model used to describe the occurrence of events in a certain period. In a Hawkes process, the occurrence of an event at time "t" not only increases the probability of an event happening in the future, but also has an influence on past events. This influence is modeled by a function called the kernel function, which describes how the intensity of the process changes as a function of time.

## ALGORITHM AND SIMULATION PROCESS

In the first part of the simulation, the Hawkes process is simulated with given parameters.

The simulation is done using two different h(t) functions, which one of them is based on uniform distribution and the other one is based on an exponential distribution.

This code simulates a Hawkes process, which is a type of point process that is commonly used to model the timing of events or activities. Several functions have been defined in the code, including one for the Hawkes simulation, and others for generating random values and distributions used within the simulation.

The" hawkes_simulation" function takes two arguments: decay and T, where decay is the decay rate (reproduction rate) and T is the upper bound for the time of the simulation which in the case of the first part is equal to 100.

The simulation starts by initializing some variables like s, dead_ppl(number of dead people), dead_ppl_list, event_times, infected_ppl (the number of infected people) and infected_ppl_list. Inside while loop, the function updates the variable s with delta_t, which is drawn from a uniform distribution. Then it calculates the intensity probability using the function sigma and function h_uniform/ h_exponential, which are defined earlier. Then it compares this probability with a random number and if this random number is less than the intensity probability it means that the event happens, thus increasing the number of infected people count and dead people count.

The simulation runs until s becomes greater than T then it returns the event times, number of dead people, list of dead people and list of infected people.

It is worth mentioning that the dead people count is a function of infected people that is The code defines: number of dead people = math.ceil(0.02 * number of infected people). You can see the comparison between the outputs in figures 1 and 2 for different h(t) functions.

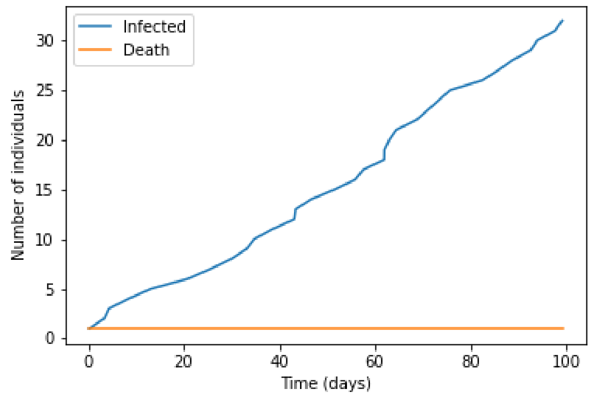


*Figure 1 - Uniform-based h(t)*

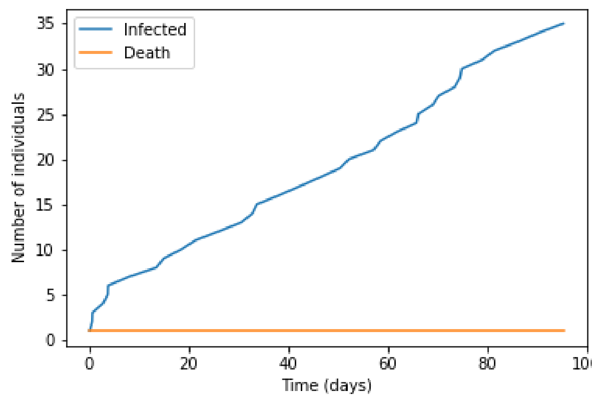


*Figure 2 Exponential-based h(t)*

There's also a plot for how the value of lambda changes over time, to see that plot, please run the code.

In the second part of the simulation a non-pharmaceutical intervention factor is introduced to the simulation after time 20, which tries to reduce the number of dead people.

minimize_scalar *function from SciPy has been used to simulate the non-pharmaceutical intervention. Here's a brief explanation of the code. Both uniform-based h(t) and the exponential-based h(t) have been used but the final code contains the exponential-based h(t). The code for the second part follows the steps as in the first part but two new functions have been introduced and some new modifications have been made. The "optimize_rho()" function is used to optimize the rho parameter that is used in the intensity of the outbreak, this function uses an optimization* *algorithm(*minimize_scalar*) to find the value of rho that minimizes the cost function.. rho is used to tune the intensity of the outbreak and the cost function is used to evaluate the performance of different values of rho. The "optimize_rho()" function takes action after the 20th day (as mentioned in the question).*

*You can see the output of the second part of the simulation in figures 3 and 4.*

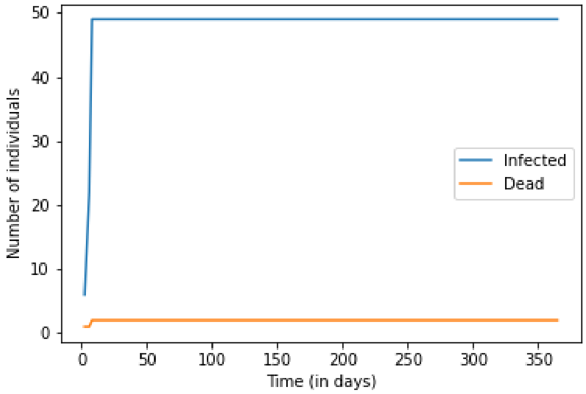


*Figure 3 - Dead and Infected*

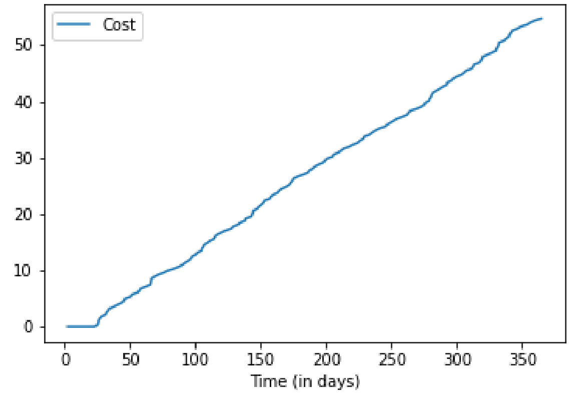


*Figure 4 - Cost change over time*