Gerardo Rivera

MATH 3020

12/02/24

MATH 3020 Project Paper

1. Introduction

The objective of the project assignment is to simulate the spread of a virus within a network of N-connected computers using a Monte Carlo method. Additionally, the following three estimates must be found during the completed simulation due to using random sampling with Monte Carlo methods to solve any problem involving some element of randomness or uncertainty.

2. Problem Description

Twenty computers are connected in a network. One computer becomes infected with a virus. Every day, this virus spreads from any infected computer to any uninfected computer with probability 0.1. Also, every day, a computer technician takes 5 infected computers at random (or all infected computers, if their number is less than 5) and removes the virus from them. Estimate:

(a) the expected time it takes to remove the virus from the whole network;

(b) the probability that each computer gets infected at least once;

(c) the expected number of computers that get infected.

3. The Program's Implementation and Methodology

(i) First, the program, which was written in Python, implements three functions with several variables that are crucial in creating one or more simulations of a virus being spread among a computer network over time.

(ii) A function called simulate_virus_spread takes three parameters, which are computers, virus_probability, and daily_repairs to create and perform a simulation of a computer virus being spread as described in the problem with the incorporation of a Monte Carlo method later in the program as this function does take important considerations such as probability and expectations to get accurate estimates for the problem described above. The function works to run a single simulation of the virus spread as the beginning of the simulation infects one computer, which is the first computer on the network, and a list is created to track which computers on the user's network have been infected at least once. Then, a while loop is used as the simulation runs to represent the computer virus spreading in the morning and the technician arriving in the afternoon to clean a certain amount of infected computers; furthermore, the while loop keeps running until no computers in the network are infected by the virus. Thus, the function's returned values are the number of days and the number of infected computers from a single simulation.

(iii) Next, another function called monte_carlo_simulation takes four parameters which are the values entered by the user to set the element of randomness as the program is running and the importance of this function is allowing the simulation, created in the simulate_virus_spread function, to run a large number of times so that the user is obtaining numbers of different realizations as those values are the long-run frequencies and averages used to calculate the probabilities and expectations in the problem. Then, the function initializes three variables to be averaged later in the program as each variable's purpose is to either track the total number of days, all computers that have been infected at least once, or the number of computers that have been infected at least once across all simulations. Furthermore, a for loop is created and works until a certain amount of simulations entered by the user, such as 10000, have been completed to aggregate the results, divide each result by the user's number of simulations, and the averages and probabilities, which are the three estimates, are calculated at the end of the function. Thus, the function's returned values are the estimates that were asked by the problem about simulating a computer virus across a network of computers.

(iv) Finally, the main function is to introduce the user to the simulation program and prompts the user to enter the following values:

- The number of computers on the network
- The probability that the virus spreads (ranging from 0 – 1)
- The number of computers repaired daily
- The number of simulations to run

Afterward, the simulation of the virus spreading among a computer network runs a large number of times so that the program returns the following three estimates to the user:

- The expected number of days for the virus to be cleared of the network
- The average probability that all computers have been infected at least once
- The expected number of infected computers

4. Numerical Results and Discussion

After running the program, I obtained the following results from my 10,000 simulations based on the given values in the problem:

(a) 56 days*

(b) 0.0005

(c) 2.95 computers

The first value can be interpreted as it takes 56 days on average to remove the virus from the whole network due to two factors like the low transmission probability, which is p = 0.1, for the virus to spread from any infected computer to any uninfected computer and the number of computers being repaired by the technician, which is 5, that causes the virus to spread slowly. Next, the second value means there's only a 0.05% chance that every computer in the network gets infected at least once during the simulations as the probability is very low because the virus is very unlikely to infect the entire network under the given conditions from the problem like the low infection probability and the technician's daily repairs to make things difficult for the virus to spread across the entire network. Lastly, the third value means that approximately 3 computers on average get infected at least once during the simulations because the virus' overall impact is highly contained in the situation provided by the problem as the virus spreads minimally and the technician's repair strategy is overall efficient based on the given values in the problem.

5. Program Link

(a) My Python Program File

(b) https://ato.pxeger.com/about

Note: the second website can be used to copy the Python code to run the program and the inputs are only the four values that the user must enter for the program to run. Afterward, the program outputs the three estimates as pointed out by the problem.