CS556: Project 1 Report

Rakesh Podder

1 Experimental Setup

The initial procedure involves importing necessary Python libraries for graph manipulation, visualization, and file operations, such as networkx, matplotlib, and csv. It then proceeds to generate network graphs using a custom function generate_and_save_graph, which creates and saves various types of networks (Barabási-Albert, Erdös-Rényi, Watts-Strogatz) with different node counts (1000, 5000, 10000) to CSV files. These networks are then used to simulate the spread of an infection without any interventions, using another function simulate_infection_without_defense. The simulation parameters include the network file, initial infected node, and infection probability. The simulations aim to analyze the spread by observing the number of iterations and the number of infected nodes per iteration, providing initial insights into the dynamics of disease spread across different network structures.

1.1 Program 1

The purpose of this program is to study the propagation of worm on the three different types of networks through simulation when no cure (that is, worm defense) is applied. The simulate_infection_without_defense function is designed to model the spread of an infection through a network represented as a graph. Here is a detailed step-by-step description of how the algorithm works, along with its inputs and outputs:

- **Inputs**:
  - `net`: A graph representing the network, where nodes represent entities (such as individuals or computers), and edges represent connections between them (such as social relationships or network cables).
  - `patient_zero`: The identifier of the initial node from which the infection starts spreading. This node is considered as the first infected entity in the network.
  - `infection_probability`: A floating-point value between 0 and 1 representing the probability that an infection will spread from an infected node to a susceptible neighboring node in each round.
- **Outputs**:
  - `round_count`: The total number of rounds (iterations) it takes for the infection process to conclude.
  - `infections_per_round`: A list where each element corresponds to the number of new infections that occurred in each round.

Algorithm Steps

1. **Initialization:**
   - A set called `infected` is created, initially containing only the `patient_zero`.
   - A counter `round_count` is initialized to zero to keep track of the simulation rounds.
   - An empty list `infections_per_round` is initialized to record the number of new infections for each round.
2. **Infection Spread Simulation:**

- The simulation enters a loop that continues until the number of infected nodes equals the total number of nodes in the network, indicating that all nodes have become infected.
- During each round of the simulation, a temporary set `newly_infected` is used to keep track of nodes that become infected in that round.

3. **Infecting Neighbors:**
   - For each node currently in the `infected` set, the algorithm iterates through its neighbors.
   - If a neighbor is not already in the `infected` set, a random number between 0 and 1 is generated.
   - If the random number is less than or equal to the `infection_probability`, the neighbor is considered infected and is added to the `newly_infected` set.
4. **Updating Infection Status:**
   - After processing all neighbors of currently infected nodes, the `infected` set is updated to include all newly infected nodes from the `newly_infected` set.
5. **Recording Infections:**
   - The number of nodes in the `newly_infected` set is appended to the `infections_per_round` list to keep a record of the spread of infection for that round.
6. **Incrementing Round Counter:**
   - The `round_count` is incremented by one to indicate the completion of a simulation round.
7. **Termination Check:**
   - If no new infections occurred during the round (i.e., `newly_infected` is empty), the loop terminates, as the infection can no longer spread.
8. **Return Results:**
   - The function returns the `round_count` and the `infections_per_round` list, providing a summary of the infection spread dynamics throughout the simulation.

This script describes a process for simulating the spread of an infection through a network of nodes. Initially, all nodes are considered available for infection, with one node designated as the initial infector. The infection spreads by assigning a random infection strength to each neighbor of an infected node. If this strength meets or exceeds a certain threshold, the neighbor is deemed infected and added to an infected list, otherwise, it's added to a safe list and considered immune. This process repeats, with a new infector chosen from the infected list each time, to avoid redundant iterations. The simulation continues until there are no more available nodes to infect, at which point it concludes by reporting the number of iterations and the tally of infected nodes at each step. Notably, the model assumes a linear infection path where each node can infect only one other node, and once a node is deemed safe or infected, it is not reconsidered for infection.

Program 1 delves into data visualization and analysis of the infection spread without cure. It involves plotting the number of infected nodes over time for each network type, using bell-shaped curves to illustrate the progression and containment of the infection, as well as S-shaped curves for cumulative infections over time. This visual analysis helps in understanding how the structure of the network influences the spread and containment of infections, offering a visual representation of the infection dynamics in scale-free, random, and small-world networks.

1.2 Program 2

The purpose of this program is to study the propagation of worm on the three different types of networks through simulation when no cure (that is, worm defense) is applied. To assess the effectiveness of different inoculation strategies in mitigating the spread of a worm through various network topologies. This program builds upon Program 1 by introducing a parallel inoculation process that competes against the infection spread.

Inputs

- `net`: A graph object representing the network with nodes and edges.
- `patient_zero`: The identifier for the initial infected node.
- `infection_probability`: The probability of infection spreading from an infected node to a susceptible node.
- `inoculator`: The identifier for the initial inoculator node.
- `inoculation_probability`: The probability of successfully inoculating a node.

Outputs

- `round_count`: The total number of simulation rounds.
- `infection_history`: A list with the number of new infections per round.
- `inoculation_history`: A list with the number of new inoculations per round.

Algorithm Steps

1. **Initialization:**
   - Create a set `infected` with `patient_zero`.
   - Create a set `inoculated` with `inoculator`, if provided.
   - Set `round_count` to zero.
   - Initialize `infection_history` and `inoculation_history` as empty lists.
2. **Simulation Loop:**
   - Continue until there are no new changes, or all nodes are infected or inoculated.
3. **Infecting Neighbors:**
   - Iterate over neighbors of each node in `infected`.
   - If a neighbor is not infected or inoculated and a random value is $\leq$ `infection_probability`, add it to `newly_infected`.
4. **Inoculating Neighbors:**
   - Iterate over neighbors of each node in `inoculated`.
   - If a neighbor is not inoculated and a random value is $\leq$ `inoculation_probability`, add it to `newly_inoculated`.
5. **Update Sets:**
   - Update `infected` and `inoculated` with newly affected nodes.
6. **Record History:**
   - Append counts to `infection_history` and `inoculation_history`.
7. **Increment Round Counter:**
   - Increment `round_count` by one.
8. **Termination Check:**
   - If no new infections or inoculations, stop simulation.
9. **Return Results:**

- Return `round_count`, `infection_history`, and `inoculation_history`.

This model simulates the dynamics of infection and disinfection within a network of nodes. Initially, all nodes are available for infection or disinfection, with specific nodes designated as the initial infector and disinfector. The simulation unfolds in linear phases, starting with the infection phase where the infector targets all adjacent nodes, assigning them a random infection strength. Nodes with strength equal to or greater than a certain probability become infected and are removed from the available pool. Conversely, nodes with lower strength are deemed immune and also removed. Following the infection phase, the disinfection phase begins, applying a similar mechanism but with cure strength determining if an infected or available node gets disinfected. The process alternates between infecting and disinfecting, with new infectors and disinfector selected from their respective lists to avoid redundancy. The simulation ends when there are no nodes left to infect or disinfect, outputting the iteration counts and numbers of infected and disinfected nodes at each step. Nodes that fail in either process are not reconsidered, ensuring the model's progression towards termination.

Program 2 focuses on evaluating mitigation strategies through the simulation of various inoculation (vaccination) strategies, such as random selection of nodes, targeting high-degree nodes, and adaptive inoculation that adjusts based on the infection spread. This part of the procedure involves developing functions to simulate these strategies and comparing their effectiveness by plotting the simulation results. The goal is to assess the impact of different mitigation strategies on the rate of infection spread and the final number of infected nodes, thereby providing insights into effective interventions for controlling disease spread in networked populations. Through these simulations, Program 2 aims to answer specific questions related to the effectiveness of mitigation strategies and the implications of different inoculation approaches on controlling the spread of infections across various network types.

2 Questions

2.1 Answer a:

S-shaped and Bell-shaped Curves: Yes, number of nodes infected over times in a network shows an S-shaped curve if we take a cumulation of nodes infected or a bell shaped curve when we just plot the number of infected nodes at any given time instance if there is no cure involved.

- **S-shaped Curve**: The cumulative number of nodes infected over time typically follows an S-shaped curve due to the logistic growth model. Initially, the infection spreads slowly because there are few infected nodes to spread the worm. As the number of infected nodes increases, the rate of spread increases because there are more nodes to infect others. However, as the network becomes saturated with infected nodes, the rate of spread slows down again because there are fewer susceptible nodes available, leading to the upper bend of the S-shaped curve.
- **Bell-shaped Curve**: The rate of new infections at any given time usually forms a bell-shaped curve. This curve rises to a peak when the infection is spreading most rapidly and then falls as the rate of new infections decreases, following the logic described above for the S-shaped curve.

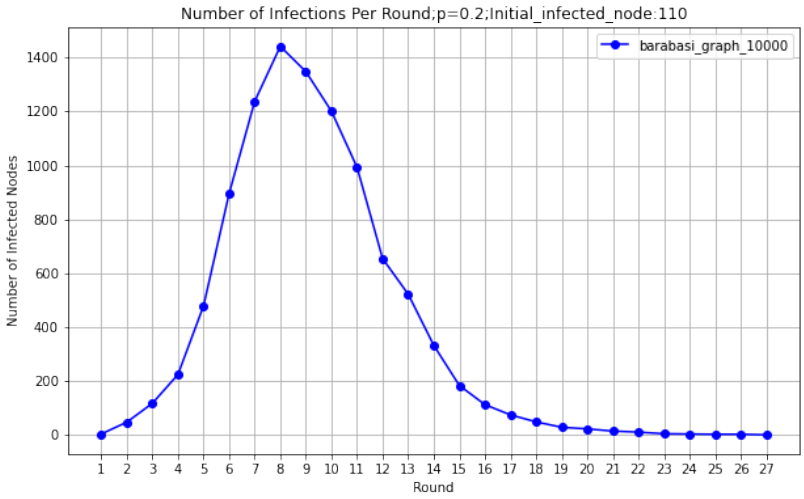

Figure 1: Barabási-Albert model with probability of infection 0.2, bell-shaped curve

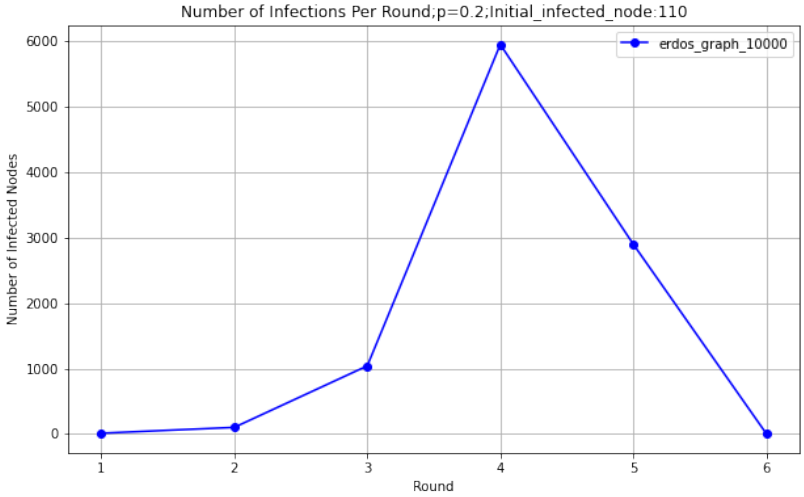

Figure 2: Erdős-Rényi model with probability of infection 0.2, bell-shaped curve

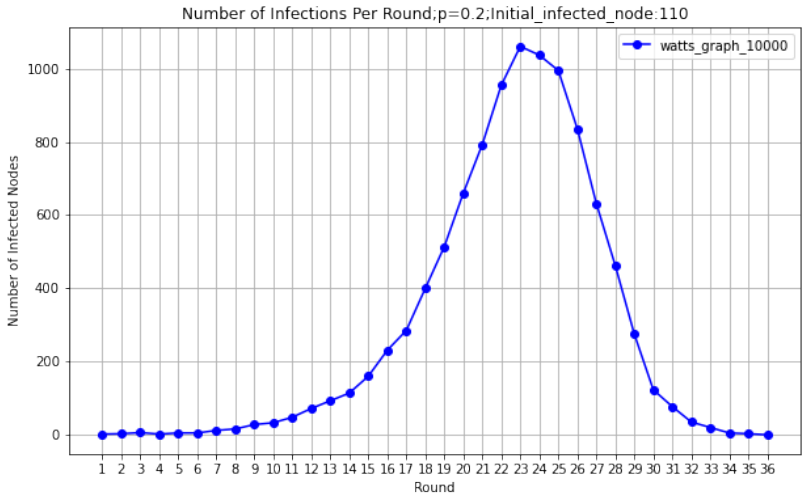

Figure 3: Watts-Strogatz model with probability of infection 0.2, bell-shaped curve

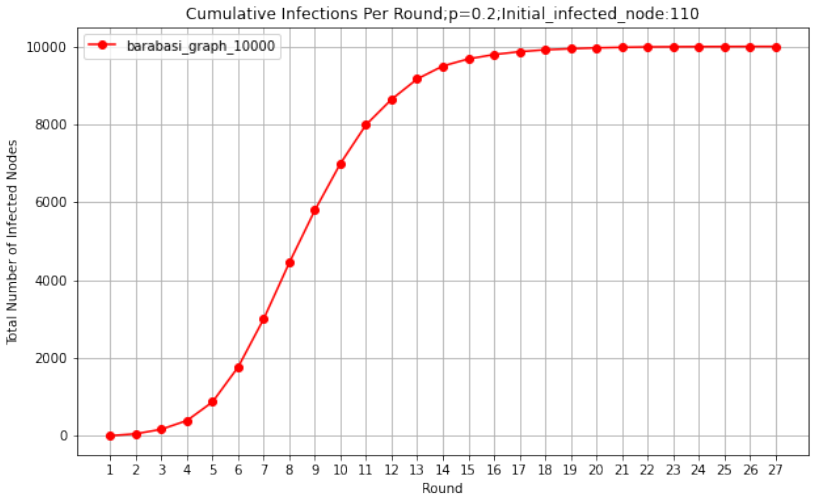

Figure 4: Barabási-Albert model with probability of infection 0.2, S-shaped curve

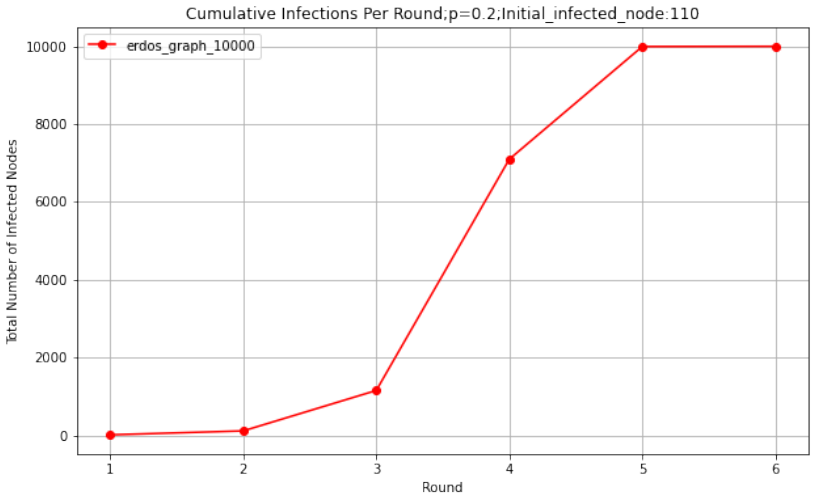

Figure 5: Erdős-Rényi model with probability of infection 0.2, S-shaped curve

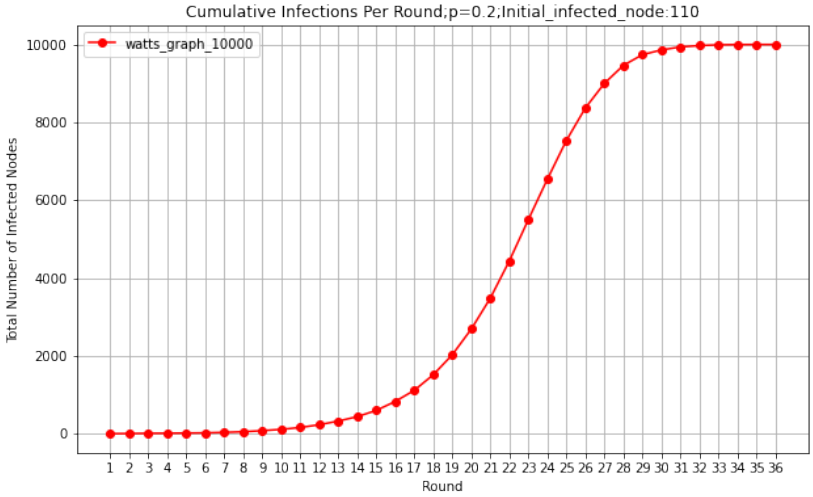

Figure 6: Watts-Strogatz model with probability of infection 0.2, S-shaped curve

The rate of spread does depend on the type of network chosen due to differences in network topology. While the basic shapes of the curves (S-shaped and bell-shaped) are consistent across different types of networks because they follow the fundamental dynamics of infectious spread, the steepness of the curves and the time to reach saturation can vary. These differences can be accounted for by the distinct topological features of each network type:

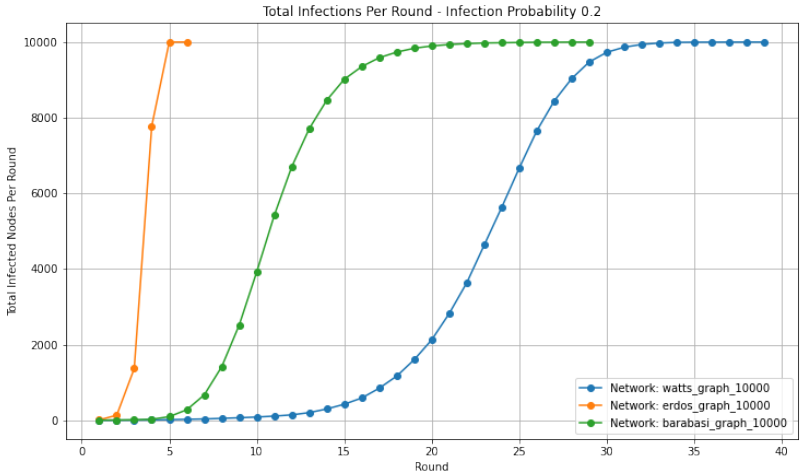

Figure 7: Models with 10000 nodes and probability of infection 0.2, S-shaped curve

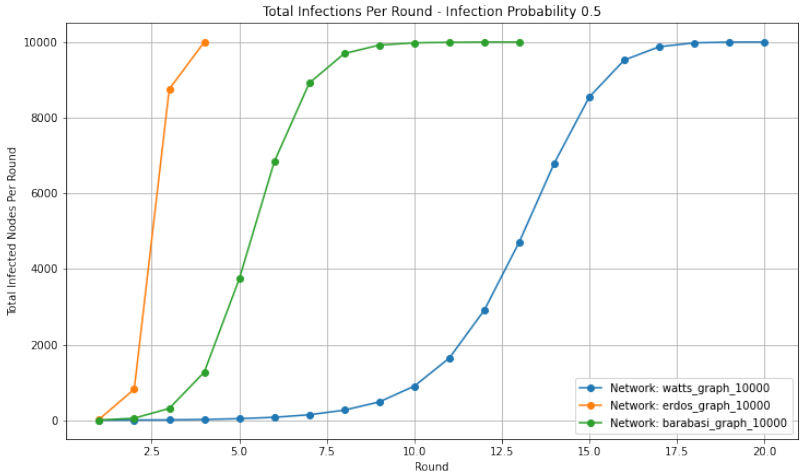

Figure 8: Models with 10000 nodes and probability of infection 0.5, S-shaped curve

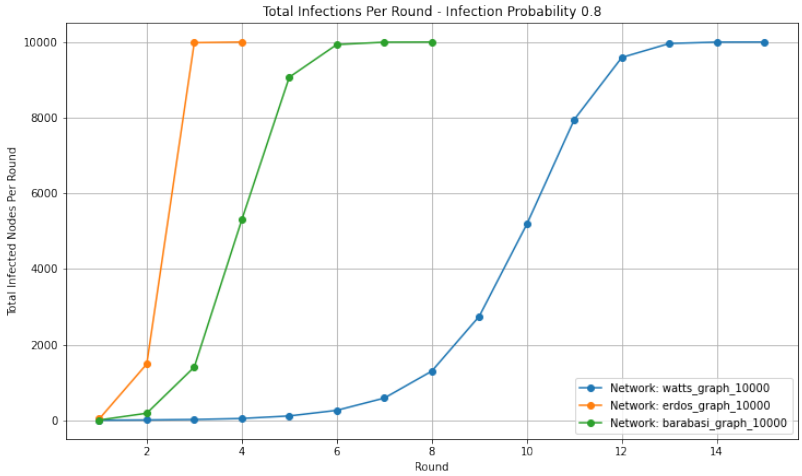

Figure 9: Models with 10000 nodes and probability of infection 0.2, S-shaped curve

As shown in figures 1 and 4, in a Barabási-Albert network with 10,000 nodes and an infection probability of 0.2, the spread of infection takes 27 iterations to reach full saturation, contrasting with faster dynamics in other models. This scale-free network initiates with a high rate of infection spread due to its hubs—nodes with significantly more connections—which efficiently disseminate the infection. However, as the infection progresses, the spread rate decreases, reflecting the diminishing number of susceptible nodes and the network's reliance on less connected nodes for continued propagation. This pattern highlights the influence of the network's scale-free topology on the temporal dynamics of infection spread.

From figures 2 and 5, an Erdős-Rényi network with 10,000 nodes and an infection probability of 0.2, the infection spreads exceptionally fast, reaching full network saturation within just six iterations. This rapid spread, characterized by a sharp peak in the infection rate, is due to the network's uniform connectivity, where each node pair has an equal chance of being linked. This high interconnectedness minimizes barriers to infection spread, allowing for swift network-wide contagion, thereby illustrating the significant impact of network topology on infection dynamics.

As illustrated in figures 3 and 6, in a Watts-Strogatz network with 10,000 nodes and an infection probability of 0.2, the infection spread exhibits a distinct pattern, requiring 36 iterations to achieve saturation. Unlike other models, the initial spread rate is low, gradually increasing as the network approaches saturation. This is followed by a swift move to full saturation after reaching the peak rate. This behavior is influenced by the Watts-Strogatz network's small-world properties, combining regular lattice and random connections. Initially, the spread is slower due to the local clustering, but as the infection reaches critical nodes that bridge clusters, the rate accelerates, demonstrating the network's efficient yet delayed propagation characteristic.

2.2 Answer b:

The spread of infections in network models shares several analogies with the spread of COVID-19, offering insights into the dynamics of real-world pandemics.

Erdős-Rényi (Random Network) and Early Spread: The rapid spread of COVID-19 in densely populated areas can be likened to the Erdős-Rényi model, where the high degree of interconnectedness among individuals facilitated swift transmission. Similar to the model's early saturation due to uniform connectivity, cities with dense social networks saw rapid increases in COVID-19 cases, underscoring the challenge of controlling spread in highly connected populations.

Barabási-Albert (Scale-free Network) and Super-Spreaders: The Barabási-Albert model, with its hubs or highly connected nodes, parallels the phenomenon of super-spreader events in the COVID-19 pandemic. Just as infecting a hub in the network leads to widespread transmission, super-spreader events—where one individual infects a disproportionately large number of others—significantly accel-

erated the pandemic's spread. This highlights the critical role of network hubs in the dynamics of infectious diseases.

Watts-Strogatz (Small-world Network) and Community Spread: The gradual increase and then rapid spread in the Watts-Strogatz model reflect the community spread of COVID-19. Initially, the virus spread slowly within localized clusters. However, as it reached connected individuals who acted as bridges between communities, the rate of spread increased, leading to widespread transmission. This model illustrates the importance of considering both local and global connections in pandemic response strategies.

2.3 Answer c:

If a defense mechanism is implemented, the number of infected nodes will be reduced compared to the scenario where no countermeasures are taken. This defense strategy aims to clean both the nodes that have been infected and those that have not yet been affected. In cases where the node's defense capabilities are insufficient to counteract the spread of the worm effectively, nodes undergoing cure interventions should be considered. The impact of such interventions across three network models is discussed below. We created three different scenarios: (i) where the infection probability matches the cure probability, (ii) where the infection probability exceeds the cure probability, and (iii) where the infection probability is lower than the cure probability.

Scenario 1 - infection probability (p) = cure probability (q): In this scenario, the networks of three different models each consist of 10,000 nodes, with an initial infected node 110 and cured node 120. Figure 18 depicts the simulation of the infected network, both with and without the implementation of a cure, at infection and cure probabilities (p and q, respectively) of 0.2. Figure 17 presents the simulation under the same conditions but with p and q increased to 0.5, and figure 12 does so with p and q at 0.8. It can be observed that, despite the infection probability (p) and cure probability (q) being equal, an increase in these probabilities results in fewer iterations needed to spread the infection and apply the cure. However, for each graph, the number of infected nodes rises, even when a cure is administered.

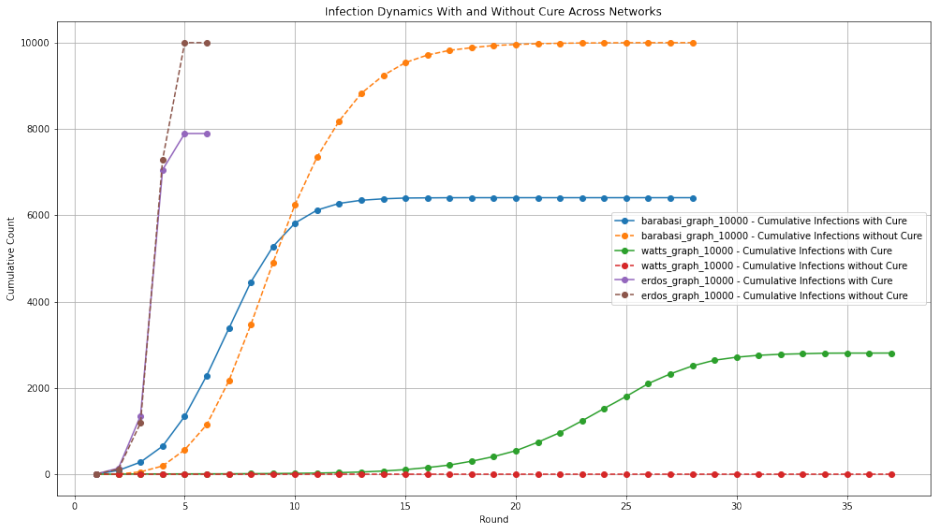

Figure 10: Models with 10000 nodes and $p = q = 0.2$

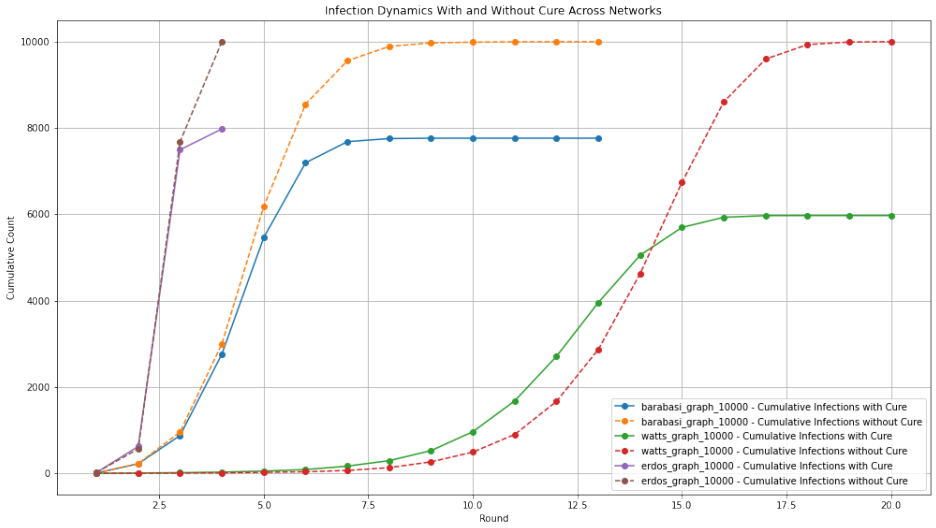

Figure 11: Models with 10000 nodes and $p = q = 0.5$

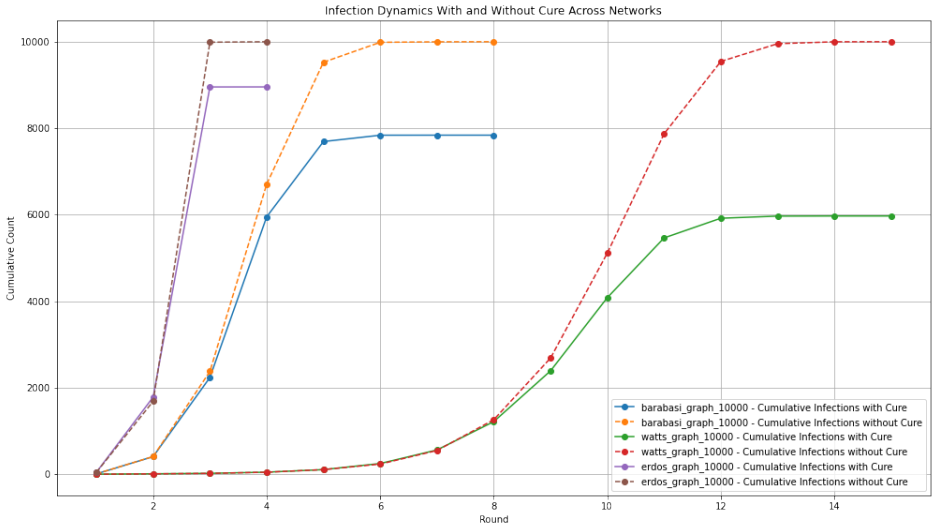

Figure 12: Models with 10000 nodes and $p = q = 0.8$

Scenario 2 - infection probability (p) > cure probability (q): With the same network configuration, we are accessing the situation where infection probability is greater than cure probability ($p = 0.5 > q = 0.2$). As shown in fig. 13, for 3 different networks the number of infected node with and with out cure is close to equal ans the number of iteration (unit time) is equal for infected networks.

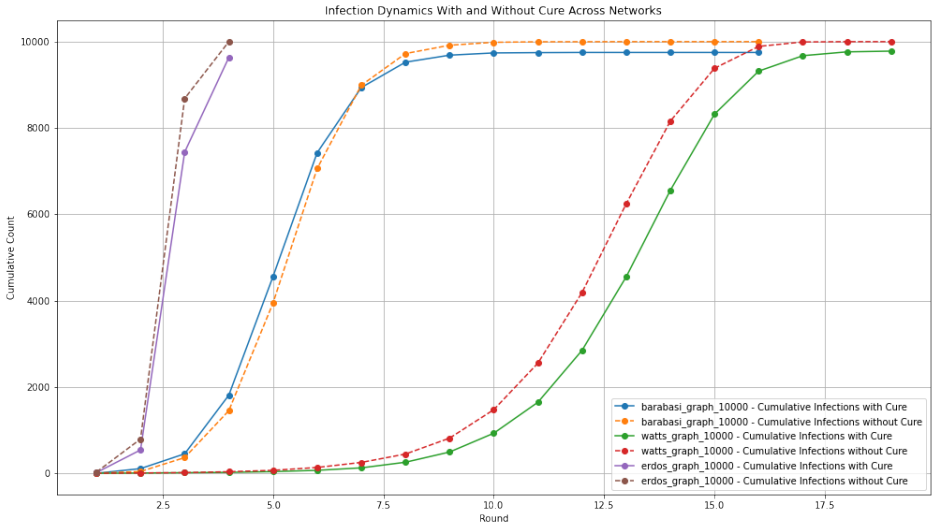

Figure 13: Models with 10000 nodes and $p = 0.5 \& q = 0.2$

Scenario 3 - infection probability (p) < cure probability (q): In this case infection probability is less than cure probability ($p = 0.5 < q = 0.8$). As ilustrated in fig. 14, for 3 different networks the number of iteration (unit time) is equal but there are a significant drop in the number infected nodes for an infected network with cure.

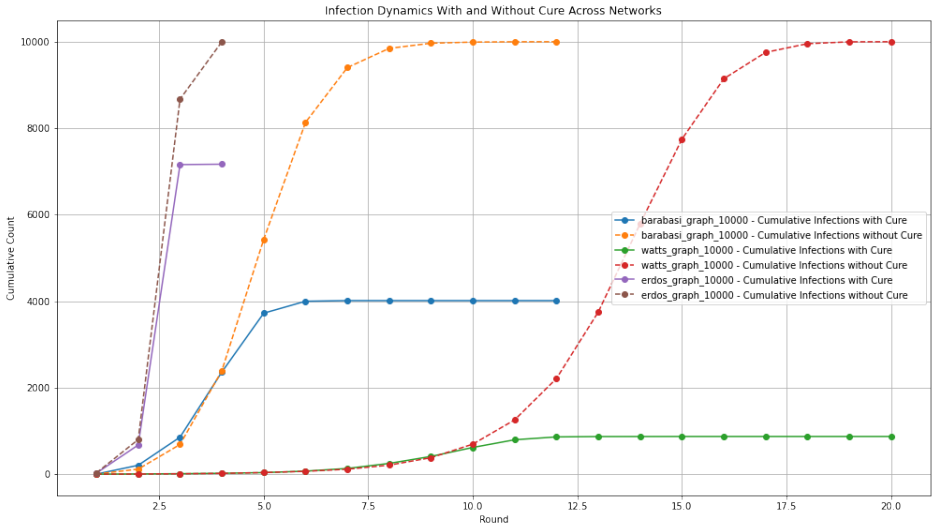

Figure 14: Models with 10000 nodes and $p = 0.5 \& q = 0.8$

2.4 Answer c:

The rate of infection differs across the three models upon the implementation of a cure. Figures 18 to 14 illustrate the infection rates for different models with different probabilities. The infection rate for the binomial network model, comparing scenarios with and without a cure exhibits significantly fewer iterations required for infection compared to the other two networks.The infection rate for the scale-free network is less than the infection rate for the small-world network model but higher than binomial network model. All network models consist of 10,000 nodes. The infection rate is influenced by the probability of infection, where a lower probability leads to a higher rate of infection, and conversely, a higher probability results in a lower infection rate.

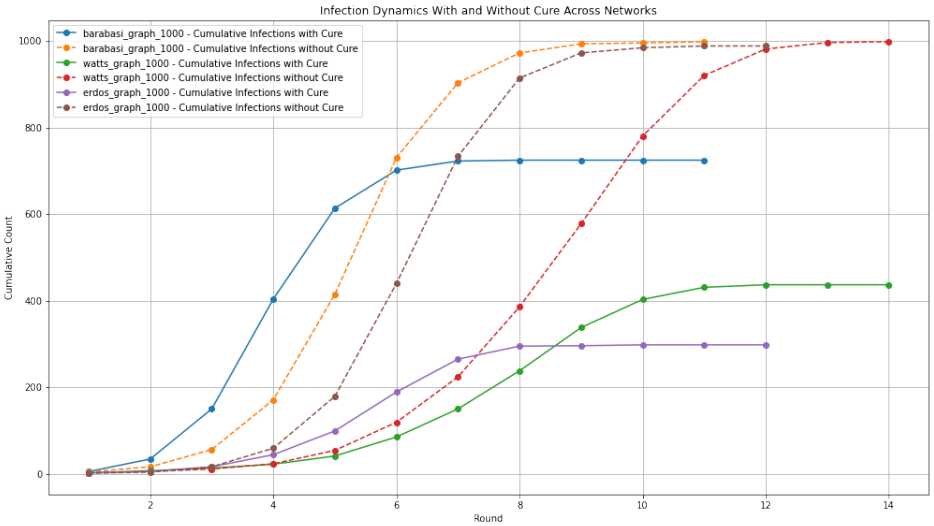

Figure 15: Models with 1000 nodes and $p = 0.5\&q = 0.5$

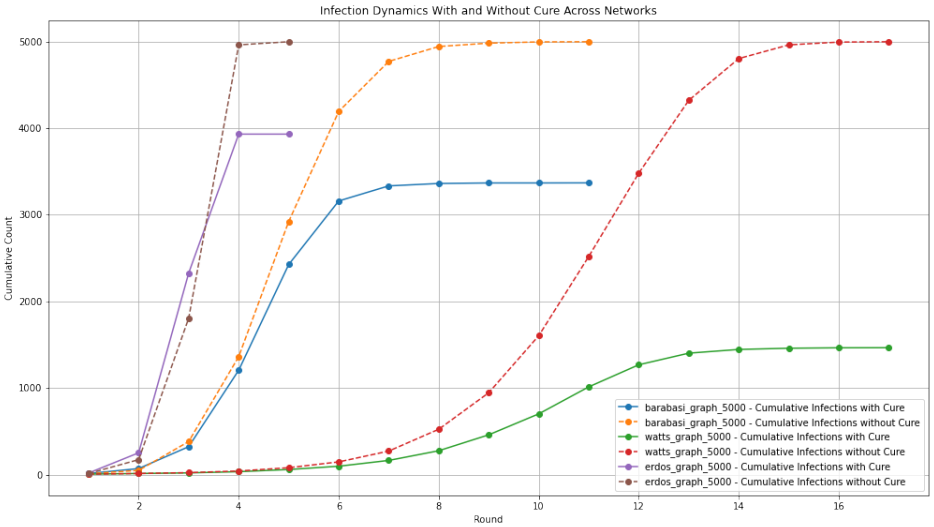

Figure 16: Models with 5000 nodes and $p = 0.5\&q = 0.5$

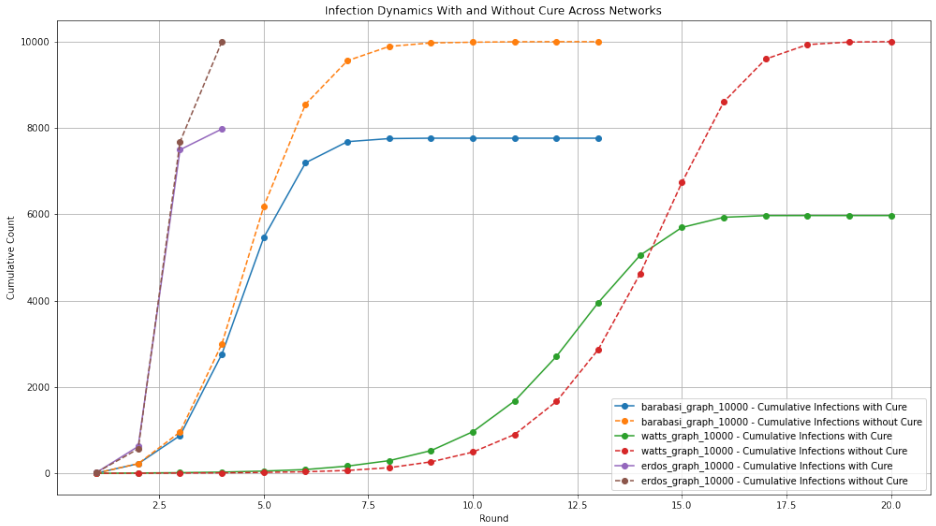

Figure 17: Models with 10000 nodes and $p = 0.5 \& q = 0.5$

As illustrated in figures 15 and 16 the infection rate and number of infected nodes with and without cure varies for for different network models. By comparing figures 18, 15 and 16, it is evident that the number of iterations gradually increases with increasing size of the networks. If we calculate the percentage of infected nodes with and without cure, there is an decrease in percentage of infected nodes as the size increases with increasing size of the networks.

2.5 Answer (e)

The analysis shows that the type of network significantly affects the spread of the worm, and simply increasing the rate of cure application is not a viable option, there are several strategic approaches to making the network more resilient against the spread. These strategies often involve identifying and targeting specific nodes based on their properties or roles within the network.

Target High-degree Nodes for Inoculation: In different network types, a small number of nodes have a disproportionately high number of connections (high-degree nodes). These nodes can act as super-spreaders if infected. A preemptive strategy could involve identifying and inoculating these high-degree nodes first, as their inoculation can significantly disrupt the worm's ability to spread. In the simulation we identified high degree nodes:

- High-degree nodes for Watts-Strogatz: ['5154']
- High-degree nodes for Barabási-Albert: ['0']
- High-degree nodes for Erdős-Rényi: ['1259']

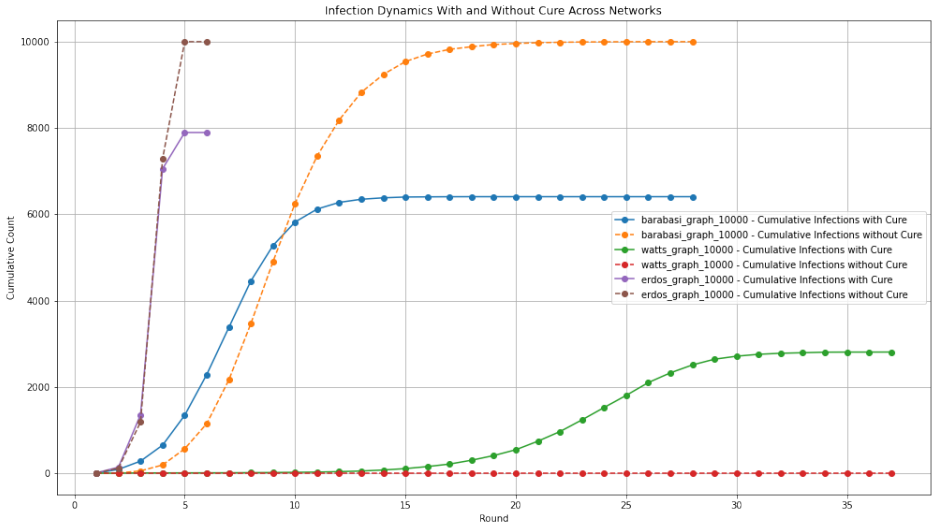

Figure 18: Models with 10000 nodes, initial inoculator node: ['120'] and $p = q = 0.2$

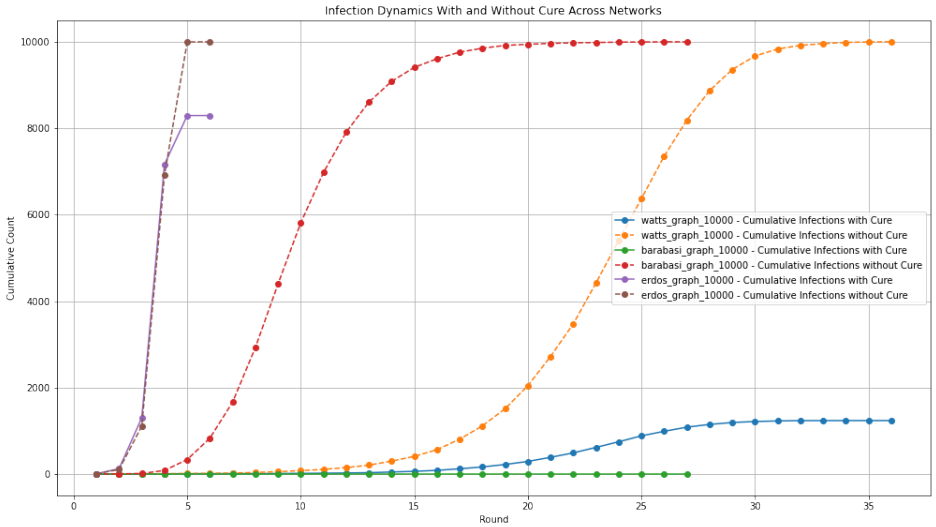

Figure 19: Models with 10000 nodes, Watts-Strogatz initial inoculated node : ['5154'], Barabási-Albert initial inoculated node : ['0'], Erdős-Rényi initial inoculated node: ['1259'] and $p = 0.2 \& q = 0.2$

By comparing figures 18 and 19, it is clear that by assigning this high degree nodes as initial inoculated node the number of infected nodes is less rather than randomly assigning.

Adaptive Inoculation Strategies: Implement adaptive strategies where the choice of nodes to inoculate evolves based on the current state of the network and the spread of the worm. This could involve real-time analysis of network traffic or infection patterns to predict the worm's next targets and inoculate them in advance.

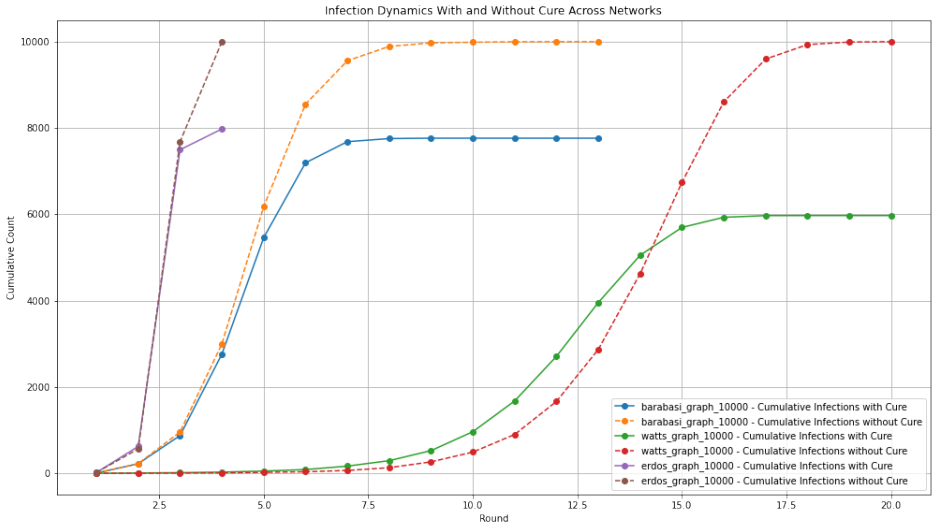

Figure 20: Models with 10000 nodes and $p = q = 0.5$, without Adaptive Inoculation Strategies

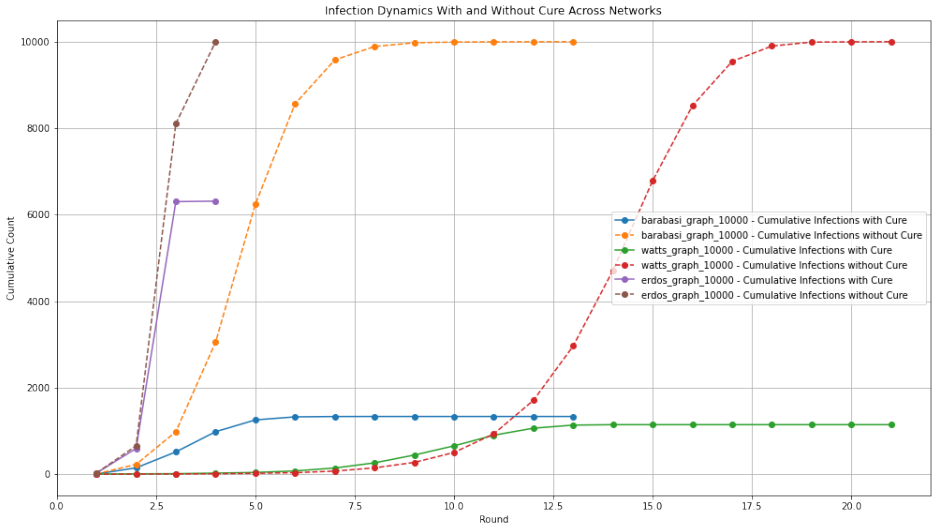

Figure 21: Models with 10000 nodes and $p = q = 0.5$, with Adaptive Inoculation Strategies

As shown in figures 20 and 21, by utilizing Adaptive Inoculation Strategies, the number of infected nudes with cure reduced significantly.