

Graph Data Mining

Project 5 - Virus Propagation

Vamshi Chidara
(schidar)

Akhil Kumar Mengani
(amengan)

Nivedita lodha
(nnlodha)

Option 1: Virus Propagation on Static Networks

1. Calculate the effective strength (s) and answer the following questions.

a. Will the infection spread across the network (i.e., result in an epidemic), or will it die quickly?

Ans :

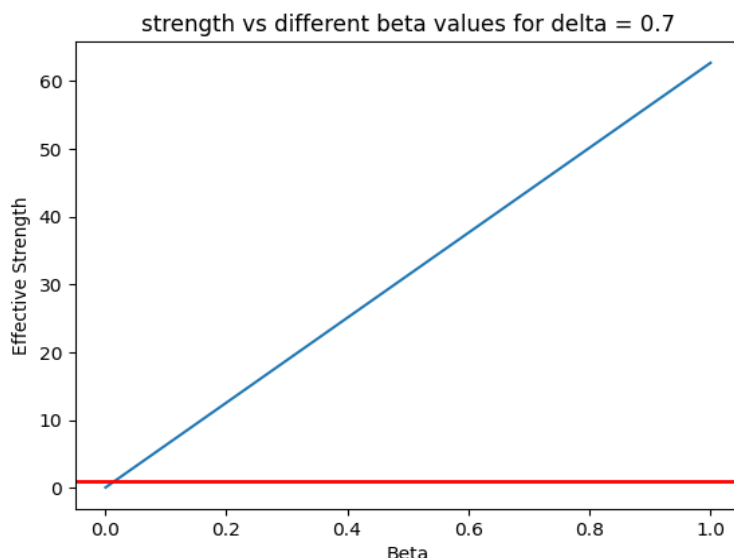
- 1) With $\beta = 0.20$ and $\delta = 0.70$, the observed effective strength is 12.529. As the observed effective strength is greater than 1, there will be an epidemic because of the infection.
- 2) With $\beta = 0.01$ and $\delta = 0.60$, the observed effective strength is 0.7309. As the observed effective strength is less than 1, there will not be an epidemic because of the infection and it will die quickly.

b. Keeping δ fixed, analyze how the value of β affects the effective strength of the virus (suggestion: plot your results). What is the minimum transmission probability (β) that results in a network-wide epidemic?

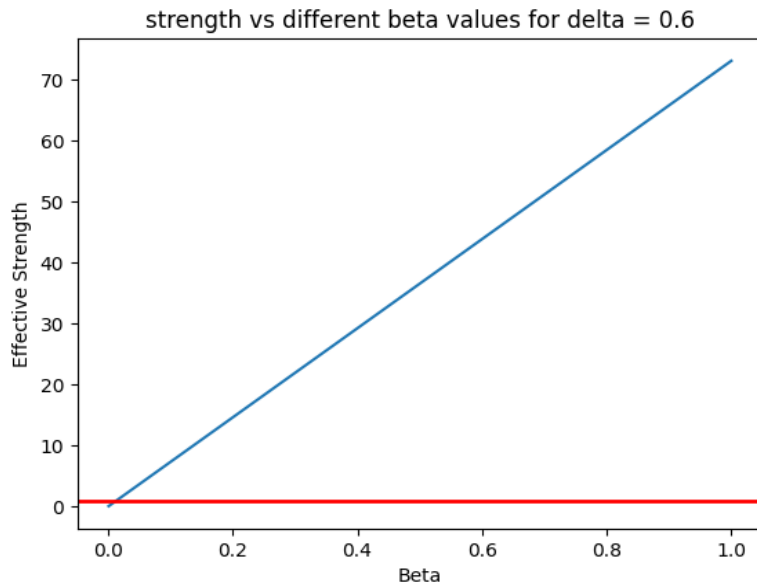
Ans:

To analyze how the value of β affects the effective strength, we varied the value of β from 0.001 to 1 with step being 0.001. Here is what we identified.

- 1) For $\delta = 0.70$, identified minimum transmission probability = 0.016



2) For $\delta = 0.60$, identified minimum transmission probability = 0.014

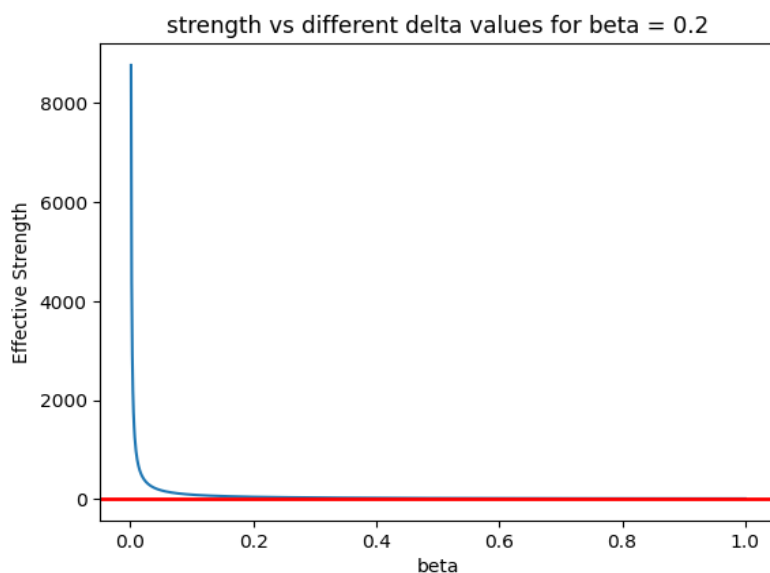


c. Keeping δ fixed, analyze how the value of β affects the effective strength of the virus (suggestion: plot your results). What is the maximum healing probability (δ) that results in a network-wide epidemic?

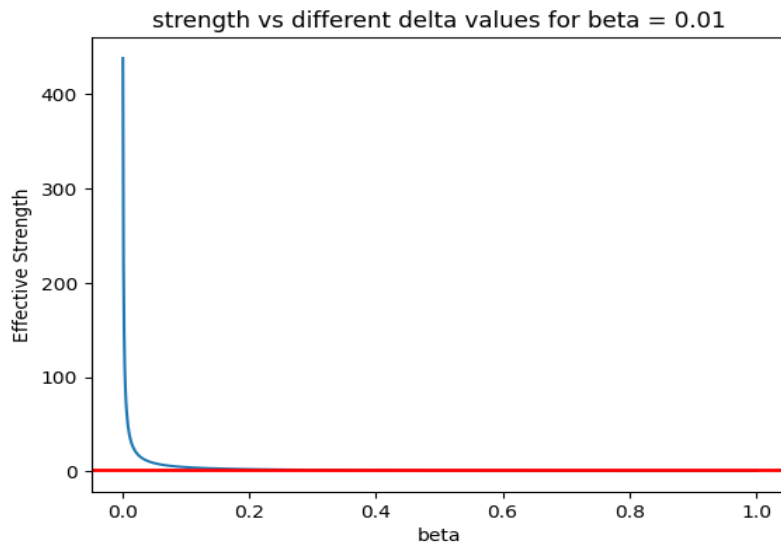
Ans:

To analyze how the value of β affects the effective strength, we varied the value of δ from 0.001 to 1 with step being 0.001. Here is what we identified.

1. For $\delta = 0.2$, the identified maximum healing probability = 1. This means that for $\delta = 0.2$, there will be a network-wide epidemic.



2. For $\beta = 0.01$, the identified maximum healing probability = 0.439. This means that for $\beta = 0.01$, there will not be a network-wide epidemic.

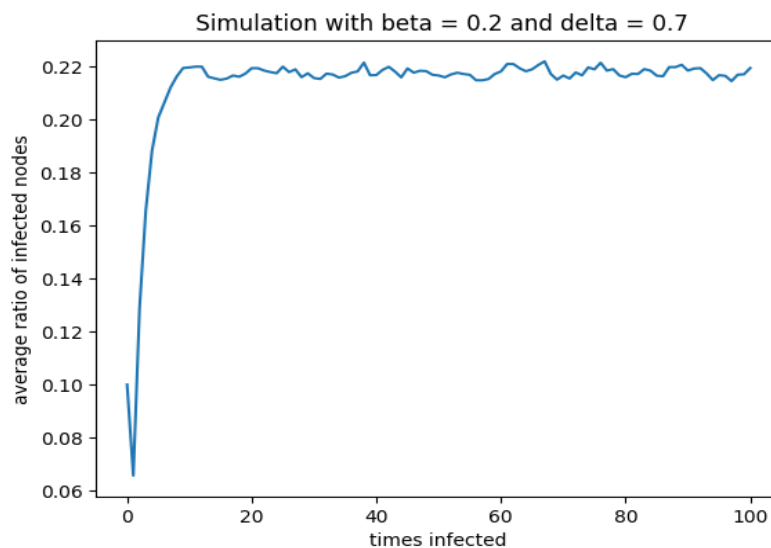


2. Simulate the propagation of virus with the SIS VPM and answer the following questions.

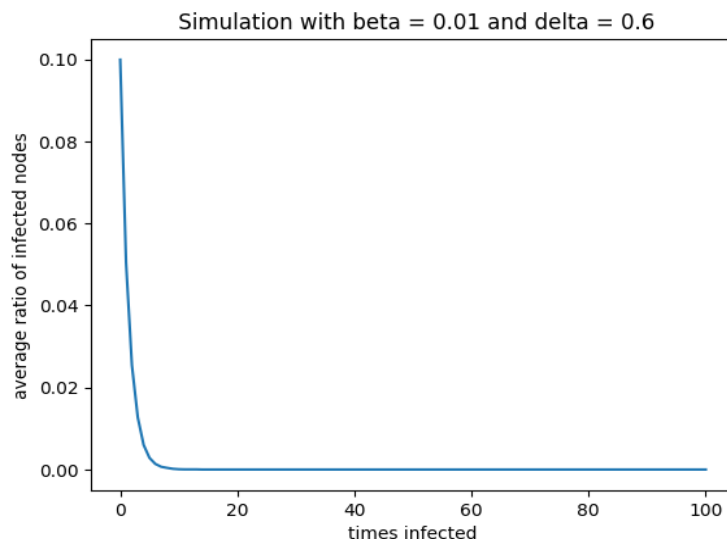
b. Plot the average (over the 10 simulations) fraction of infected nodes at each time step. Did the infection spread across the network, or did it die quickly? Do the results of the simulation agree with your conclusions in (1a)?

Ans:

- 1) When we simulate the propagation of virus in SIS VPM (10 times) for $\beta = 0.20$ and $\delta = 0.70$, we observed that there will be an epidemic. This result is along the same lines as observed in 1a.



- 2) When we simulate the propagation of virus in SIS VPM (10 times) for $\beta = 0.01$ and $\delta = 0.60$, we observed that the virus dies out quickly and will not result in an epidemic. This result is along the same lines as observed in 1a.



3. Implements an immunization policy to prevent the virus from spreading across the network. Answer the following question.

- a. What do you think would be the optimal immunization policy? What would be its time complexity? Would it be reasonable to implement this policy? Justify.

Ans:

The optimal immunization policy is to find k nodes which are the most significant in the network (k -node subset). Significance of a node can be measured in terms of node's eigen-value or pagerank or degree etc. The time complexity involved in identifying these k most significant nodes has to be polynomial so that the immunization policy is practically feasible. If not, the policy will be computationally expensive and cannot be applied.

For each heuristic immunization policy (A, B, C, and D) and for the static contact network provided (*static.network*), answer the following questions:

- b. What do you think is the intuition behind this heuristic?

- **Policy A:** In this immunization policy, we select k random nodes in the network and immunize them. We expect that immunizing these random nodes will stop or prevent the spread of virus in the network. This policy will not be the ideal one if we have a large network and the value of k is small. We can also not guarantee that this policy will prevent the spread of virus as we are choosing nodes randomly.
- **Policy B:** In this immunization policy, we select k nodes which have the highest degree in the network and immunize them. So, we are choosing k most significant nodes

where the significance of a node is measured as its degree in the network. This policy is more effective than policy A in which we select k random nodes for immunization.

- **Policy C:** In this immunization policy, instead of choosing k nodes with highest degree in the first stage itself, we choose the node with the highest degree at that point, immunize that node, remove that node from the network and repeat until all vaccines are administered. In other words, this process is repeated k times as we have k vaccines.
- **Policy D:** In this immunization policy, we find the eigen-vector corresponding to the largest eigen-value of the network. We find k largest absolute values in this eigen-vector. We immunize the nodes with positions corresponding to these k largest values. In other words, we use eigen-value as a measure and select k most significant nodes and then immunize them.

c. Write a pseudocode for this heuristic immunization policy. What is its time complexity?

Ans:

Policy A:

Input : k : Number of vaccines to be administered
adj_matrix : Adjacency matrix of the graph

Output: adj_matrix : Updated adjacency matrix after immunization.

Pseudo code:

```
def policyA(adj_matrix, k):
    nodes_to_be_immunized = [ ]
    while(nodes_to_be_immunized < k):
        r = rand(0, len(adj_matrix[0]))
        if(r not in nodes_to_be_immunized):
            nodes_to_be_immunized.append(r)

    for node in nodes_to_be_immunized:
        for i in range(len(adj_matrix[node])):
            adj_matrix[node][i] = 0
            adj_matrix[i][node] = 0

    return adj_matrix
```

Time complexity :

Complexity involved in selecting k nodes : $O(k)$

Complexity involved in immunizing chosen k nodes : $O(kN)$

Total time complexity : $O(k) + O(kN)$

Policy B:

Input : k : Number of vaccines to be administered.
adj_matrix : Adjacency matrix of the graph.
graph : Graph representation of the network.

Output: adj_matrix : Updated adjacency matrix after immunization.

Pseudo code:

```
def policyB(adj_matrix, k):
    nodes_to_be_immunized = [ ]
    node_degrees = sorted(list(graph.degree()), key = lambda ele:ele[1], reverse = True)
    for i in range(k):
        nodes_to_be_immunized.append(degree[i][0])

    for node in nodes_to_be_immunized:
        for i in range(len(adj_matrix[node])):
            adj_matrix[node][i] = 0
            adj_matrix[i][node] = 0

    return adj_matrix
```

Time complexity :

Complexity involved in sorting : $O(N\log N)$
Complexity involved in selecting k nodes : $O(k)$
Complexity involved in immunizing chosen k nodes : $O(kN)$
Total time complexity : $O(N\log N) + O(k) + O(kN)$

Policy C:

Input : k : Number of vaccines to be administered
adj_matrix : Adjacency matrix of the graph
graph : Graph representation of the network

Output: adj_matrix : Updated adjacency matrix after immunization.

Pseudo code:

```
def policyC(adj_matrix, k, graph):
    nodes_to_be_immunized = [ ]
    N = len(adj_matrix[0][0])
    while(nodes_to_be_immunized < k):
        degrees = sorted(list(graph.degree()), key = lambda ele:ele[1], reverse = True)
```

```

nodes_to_be_immunized.append(degrees[0][0])
graph.remove_node(degree[0][0])

adj_matrix = [[0 for _ in range(N)] for _ in range(N)]
for i in nx.edges(graph):
    adj_matrix[i[0]][i[1]] = 1
    adj_matrix[i[1]][i[0]] = 1

return adj_matrix

```

Time complexity :

Complexity involved in sorting : $O(kN\log N)$

Complexity involved in removing k nodes : $O(kN^2)$

Complexity involved in generating updated adjacency matrix : $O(N^2)$

Total time complexity : $O(kN\log N) + O(kN^2) + O(N^2)$

Policy D:

Input : k : Number of vaccines to be administered

adj_matrix : Adjacency matrix of the graph

graph : Graph representation of the network

```

def PolicyD(adj_matrix, k, graph):
    N = len(adj_matrix[0][0])
    eigen_val, eigen_vec = np.linalg.eig(adj_matrix)
    eigen_set = [(eigen_val[i], eigen_vec[i]) for i in range(len(eigen_val))]
    eigen_set = sorted(eigen_set, key=lambda ele: ele[0], reverse=True)
    largest_eigen_vector = eigen_set[0][1]
    largest_eigen_vector = np.absolute(largest_eigen_vector)
    nodes_to_be_immunized = [ele[0] for ele in sorted(enumerate(largest_eigen_vector),
                                                    reverse=True, key=itemgetter(1))[:k]]

    for node in nodes_to_be_immunized:
        graph.remove_node(node)

    adj_matrix = [[0 for _ in range(N)] for _ in range(N)]
    for i in nx.edges(graph):
        adj_matrix[i[0]][i[1]] = 1
        adj_matrix[i[1]][i[0]] = 1
    return adj_matrix

```

Time complexity :

Complexity involved in computing eigen-values : $O(N^3)$

Complexity involved in sorting eigen-values: $O(N\log N)$

Complexity involved in removing k nodes : $O(kN^2)$

Complexity involved in generating updated adjacency matrix : $O(N^2)$

Total time complexity : $O(N^3) + O(N\log N) + O(kN^2) + O(N^2)$

d. Given $k = k_1$, $\beta = \beta_1$, and $\delta = \delta_1$, calculate the effective strength (s) of the virus on the immunized contact network (i.e., contact network without immunized nodes). Did the immunization policy prevent a network wide epidemic?

Solution:

Results of Effective Strength of the Virus:

Immunization using Policy A, $k=200$

For $\beta = 0.2$ and $\delta = 0.7$, effective strength = 12.308842563118215

For $\beta = 0.01$ and $\delta = 0.6$, effective strength = 0.7180158161818957

Simulation for Policy A

Immunization using Policy B, $k=200$

For $\beta = 0.2$ and $\delta = 0.7$, effective strength = 1.0802754802352497

For $\beta = 0.01$ and $\delta = 0.6$, effective strength = 0.06301606968038956

Simulation for Policy B

Immunization using Policy C, $k=200$

For $\beta = 0.2$ and $\delta = 0.7$, effective strength = 1.0845205395194362

For $\beta = 0.01$ and $\delta = 0.6$, effective strength = 0.06326369813863378

Simulation for Policy C

Immunization using Policy D, $k=200$

For $\beta = 0.2$ and $\delta = 0.7$, effective strength = 3.5392879901659318

For $\beta = 0.01$ and $\delta = 0.6$, effective strength = 0.20645846609301266

From the above results, it's clear that Policy B, Policy C have performed better than Policy A and Policy D with the same transmission probability of 0.2 and healing probability of 0.7. With the transmission probability of $\beta = 0.2$ and healing probability of 0.7, the effective strength of virus for Policy A and Policy D is 12.3 and 3.5 respectively. Whereas, for the β and δ , for Policy B and Policy C, the effective strength of the virus is 1.08 and 1.08 respectively.

Thus, Policy B and Policy C will be able to mitigate the propagation of the virus and can be used to prevent a network wide epidemic.

On the other hand, Policy A and Policy D cannot prevent a network wide epidemic because of the ineffectiveness in mitigating virus propagation.

e. Keeping β and δ fixed, analyze how the value of k affects the effective strength of the virus on the immunized contact network (suggestion: plot your results). Estimate the minimum number of vaccines necessary to prevent a networkwide epidemic?

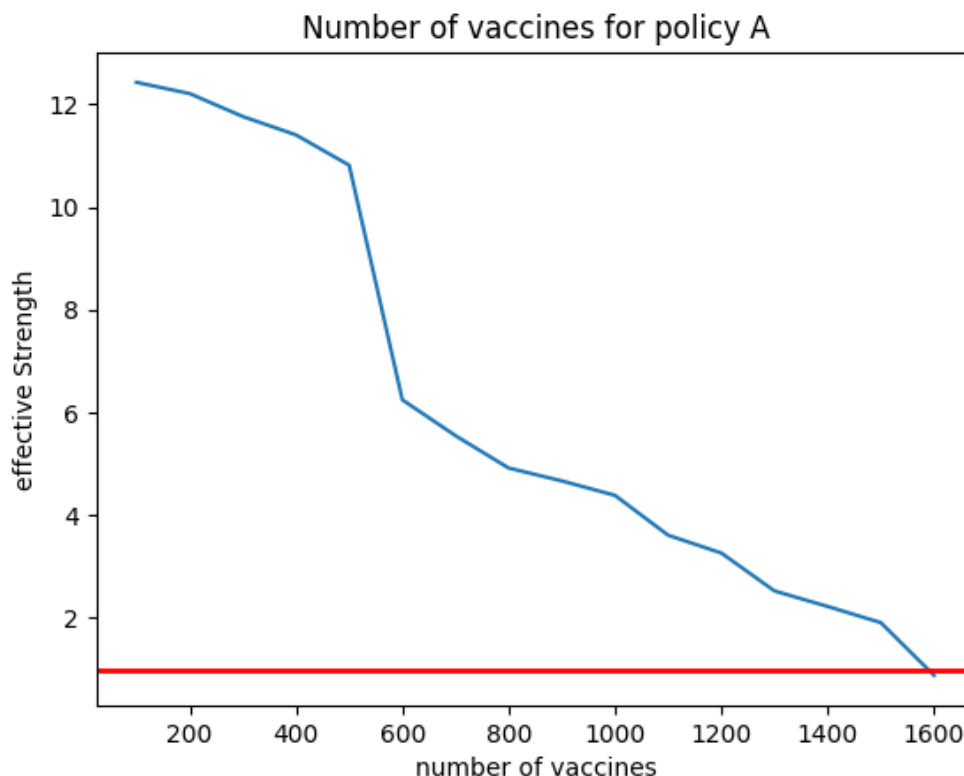
Solution:

Keeping β and δ fixed, k (Number of vaccines) is incrementally increased by 100. 'k' is increased until the effective strength of the virus dips down to less than 1. This is done to figure out the number of vaccines needed to prevent a network wide epidemic.

POLICY A:

Below graph shows the numbers of vaccines against effective strength of the virus needed to prevent the network wide epidemic. As shown in the graph, the intersection for number of vaccines is between 1500 and 1600 when effective strength = 1.0.

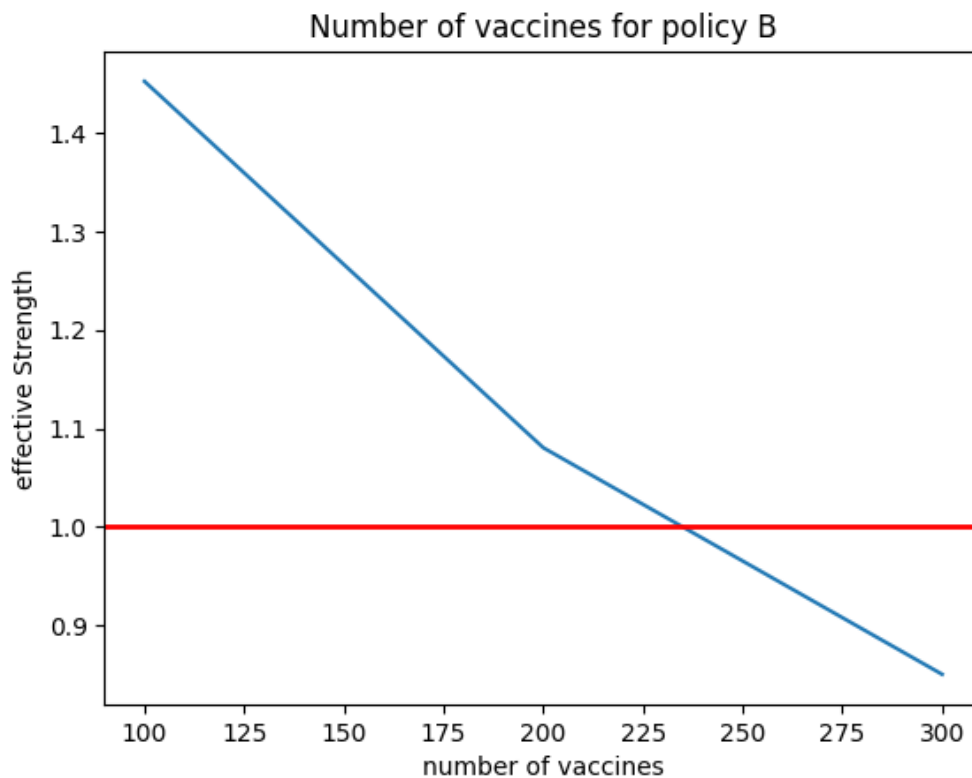
Therefore,
minimum number of vaccines necessary to prevent a networkwide epidemic
>1500 and <1600



POLICY B:

Below graph shows the numbers of vaccines against effective strength of the virus needed to prevent the network wide epidemic. It is clear from the graph that the number of vaccines to mitigate the effective strength of the virus to ≤ 1 is **between 225 and 250**.

Therefore,
minimum number of vaccines necessary to prevent a network wide epidemic
>225 and ≤ 250

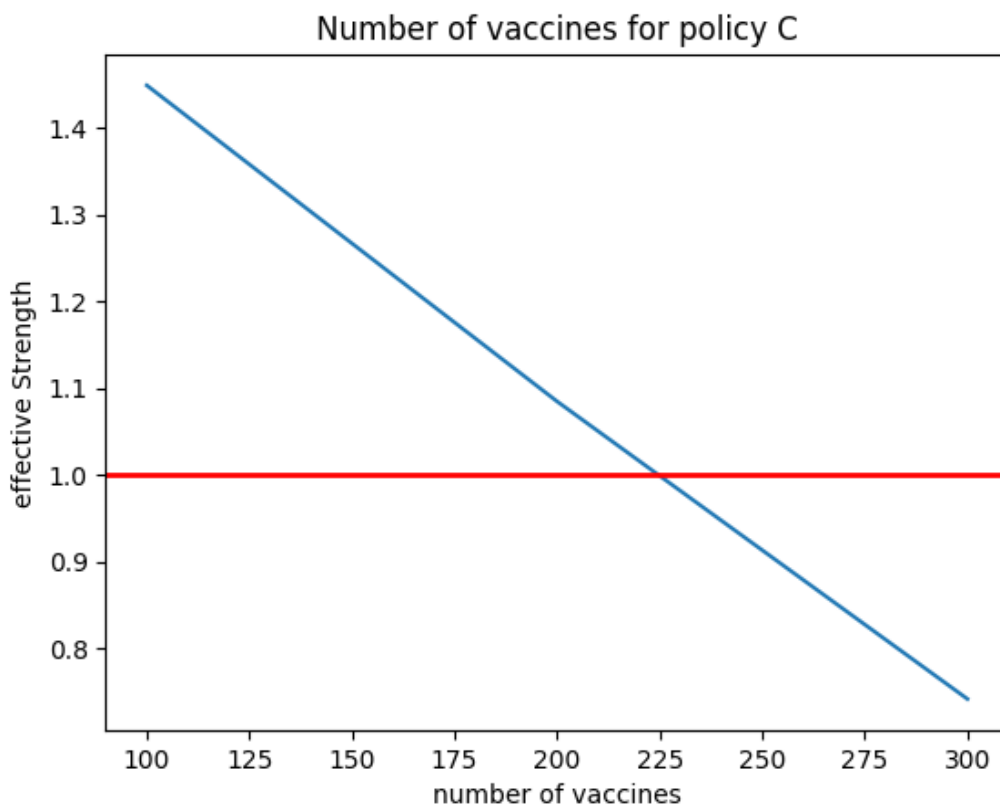


POLICY C:

Below graph shows the numbers of vaccines against effective strength of the virus needed to prevent the network wide epidemic. It is clear from the graph that the number of vaccines to mitigate the effective strength of the virus to ≤ 1 is **approximately 225**.

Therefore,

minimum number of vaccines necessary to prevent a network wide epidemic = 225

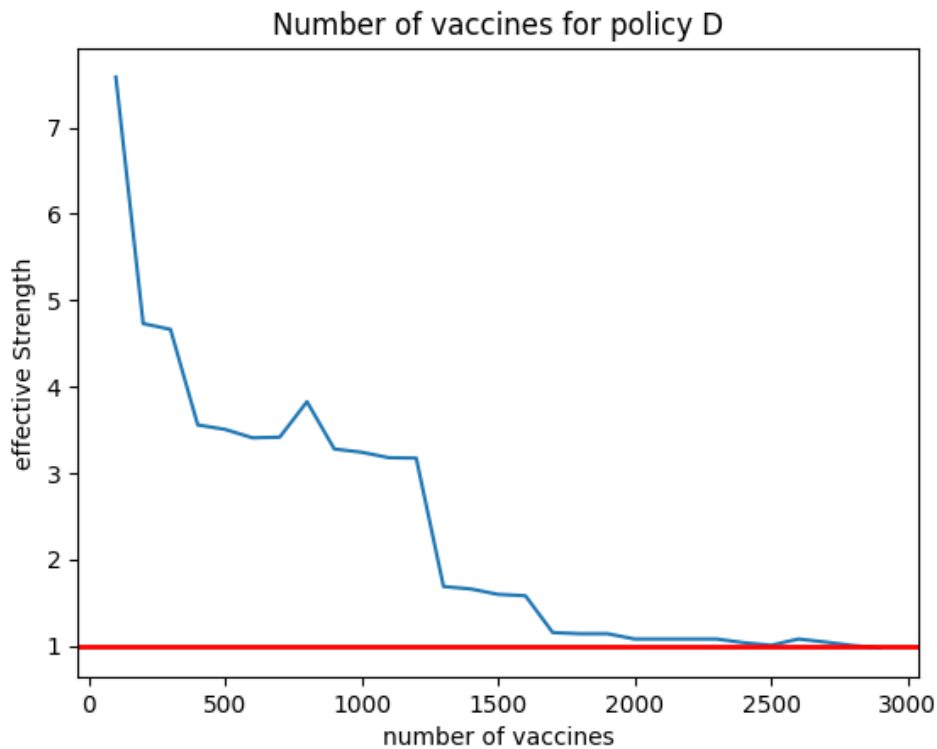


POLICY D:

Below graph shows the numbers of vaccines against effective strength of the virus needed to prevent the network wide epidemic. It is clear from the graph that number of vaccines to mitigate the effective strength of the virus to ≤ 1 is **between 2800 and 3000**.

Therefore,

minimum number of vaccines necessary to prevent a network wide epidemic = **2800 to 3000**

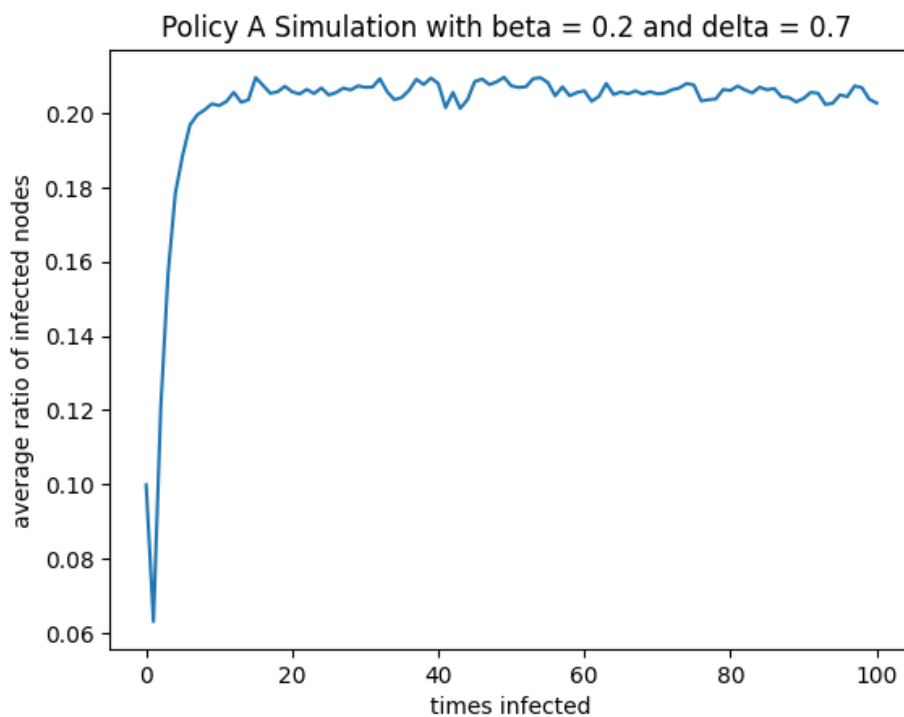


f. Given $k = k_1$, $\beta = \beta_1$, $\delta = \delta_1$, $c = n/10$, and $t = 100$, run the simulation from problem (2) for the immunized contact network 10 times. Plot the average fraction of infected nodes at each time step. Do the results of the simulation agree with your conclusions in (3d)?

Solution:

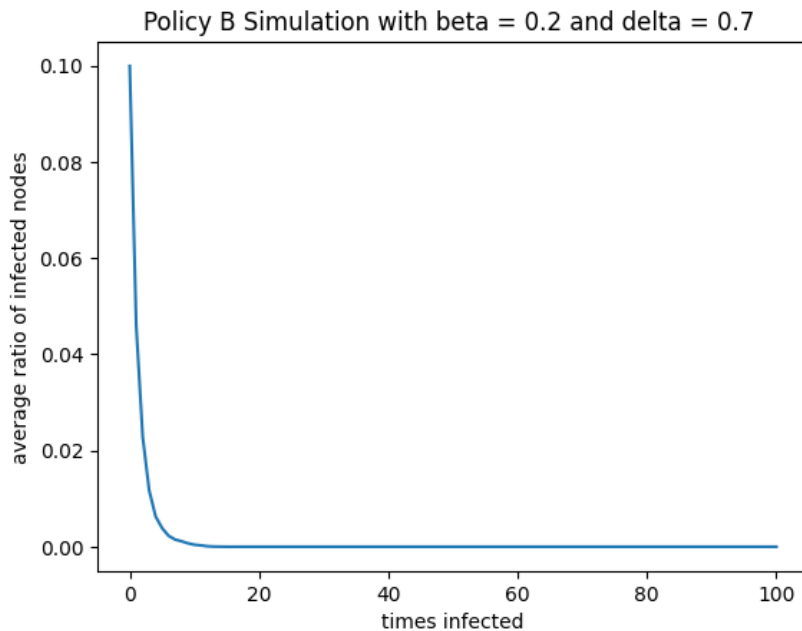
POLICY A:

From the below graph, Policy A still **will result in a network wide epidemic**. The results of the simulation agree with the conclusions made in (3d).



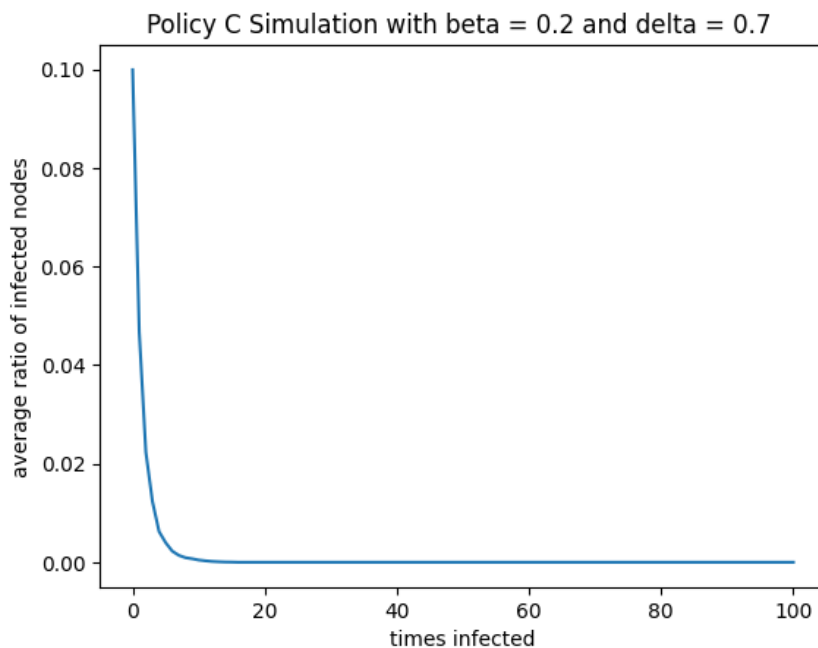
POLICY B:

From the below graph, Policy B still will **not result in a network wide epidemic**. The results of the simulation agree with the conclusions made in (3d).



POLICY C:

From the below graph, Policy C still will **not result in a network wide epidemic**. The results of the simulation agree with the conclusions made in (3d).



POLICY D:

From the below graph, Policy D will may result in a network wide epidemic sometimes or it may not result in a network wide epidemic.

The results of the simulation agree with the conclusions made in (3d).

