

# ON THE IMMUNIZATION OF SMALL COMPUTER NETWORKS

BRENNEN FAGAN<sup>†</sup>  
CALIFORNIA STATE UNIVERSITY, FRESNO  
ADVISOR: DOREEN DE LEON\*  
CALIFORNIA STATE UNIVERSITY, FRESNO

ABSTRACT. As complex computer networks have become integral to our life, it has become important to create effective and efficient defenses against computer viruses. To model these viruses and networks, a computer network can be treated as a graph of nodes and edges. For small graphs, we can use centrality measures to identify which nodes to immunize first. We examine an existing immunization scheme based on the computation of a bound for the decay of a virus and compare it to some proposed schemes on random graphs. We observe that an immunization scheme based on the organization of a graph can outperform a greedy scheme. We also observe that the computation of the virus decay bound does not correlate in the expected way with the ability of the virus to infect an entire network.

## 1. INTRODUCTION

Many techniques exist to model the spread of infectious diseases in the populace, each customizable to the disease in question, as seen in [9]. Almost all such techniques approximate the unknowable and random nature of the contact patterns between individuals. In contrast, this almost impossible problem of determining effective contact between individuals does not exist in the computing world. Hence, it becomes possible to regard computers as discrete objects with well known connections to each other, enabling modeling using graph theory, as seen in [10]. In infectious disease modeling, it is understood that some organisms or communities are more important than others in the spread of disease [9]. The graph theory analog of this is centrality.

Literature, such as [2], has proposed the usage of PageRank and other centrality measures to stop the spread of viruses by identifying important nodes. In contrast, Preciado et al., in [7], suggest that no simple correspondence between graph organization and optimal immunization schemes exists. Instead, the use of a viral decay parameter  $\varepsilon$  is proposed. In [10], it is shown that the spread of a virus is similar to a random walk, suggesting that graph organization does influence the effectiveness of a virus. Our work seeks to determine the effectiveness of centrality and graph organization as predictors of optimal immunization schemes, as well as the efficacy of  $\varepsilon$  in predicting the success or failure of a virus. As far as the authors know, current literature does not address the correlations between  $\varepsilon$  and the characteristics of a graph or virus, instead addressing the usage of it as a way to produce an effective immunization scheme. This usage bypasses the organization of a graph usually, so herein we seek to relate  $\varepsilon$  to its context within the graph as a whole. We hope that this work will demonstrate the successes and failures of  $\varepsilon$  and its related immunization schemes.

---

*Date:* May 22, 2016.

<sup>†</sup>Electronic Address: rydrion@mail.fresnostate.edu.

\*Electronic Address: doreendl@csufresno.edu.

This paper focuses on how to measure a graph’s organization and how to convert this measurement into useful information, such as  $\varepsilon$ , in order to accurately predict how a virus will spread in a graph. First we discuss infectious disease modeling and graph theory in Section 2, with emphasis on properties of small graphs. In Section 3, we discuss  $\varepsilon$  and immunization schemes for small graphs. In Section 4 we discuss the implications of, and conclude, our work before finally discussing possible extensions in Section 5.

## 2. THEORY

**2.1. Infectious Modeling.** A number of models exist for infectious disease modeling; here, we use a simple **Susceptible-Infectious-Susceptible** model (**SIS** model). This model consists of a node beginning as **susceptible**. When said node comes into effective contact with the virus — when there is sufficient contact to transfer infection between an infected node and this susceptible node — it can become **infected**. Such an infected node then has the possibility of recovery without resistance, becoming **susceptible** again (see [9]). This process is demonstrated in Figure 1. We assume that the viruses we are studying are ones for which a proper virus removal tool has not been created, implying that the only way to get rid of a virus is the removal of all infected parts of the computer. We note that our immunization is a proactive, preventative measure, representing anything from employee training to better firewalls and anti-virus software. This makes it harder for a virus to infect a computer.

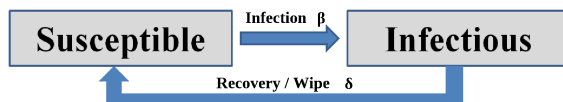


FIGURE 1. The standard SIS model is governed by infection and recovery likelihoods.

Next, we present some properties of both the virus and a hypothetical immunization.

- The strength of the virus  $\bar{\beta}$  is the likelihood that the virus will successfully infect an unprotected node that it attacks.
- The *Fan* of the virus is its reproductive strategy. For example, the virus might be programmed to attack all possible nodes ( $Fan = \text{all}$ ), or it may reproduce some number of times and let the network carry the virus’s duplicates to the node’s neighbors (that is, there exists some natural number  $n$  such that  $Fan = n$ ).
- The recovery rate  $\delta$  is the likelihood that the virus will be removed from a node in a given time step.
- The strength of the immunization is the amount that the immunization reduces  $\bar{\beta}$ . We designate as  $\underline{\beta}$  the virus’s reduced likelihood of successfully infecting this node. Furthermore, we assume the application of the immunization to be binary; either a node is immunized, or it is not.

**2.2. Graphs.** A graph is an ordered pair  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  consisting of the sets  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  of vertices (or nodes) and  $\mathcal{E} = \{(v_i, v_j) : v_i, v_j \in \mathcal{V}\}$  of edges. Any graph can be represented by its **adjacency matrix**  $A \in \mathbb{R}^{n \times n}$ , where  $a_{ij} \in A$  is given by 1 if  $(v_i, v_j) \in \mathcal{E}$ , and 0 otherwise. Graphs’ edges have two additional properties, weight and direction. A graph is weighted if the graph is of the form  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ , where  $\mathcal{W} = \{w_{(v_i, v_j)} : (v_i, v_j) \in \mathcal{E}\}$  and entries  $\{w_{(v_i, v_j)}\}$  correspond to weights of the edges in  $\mathcal{E}$ . In this case, the adjacency matrix

is represented by  $A \in \mathbb{R}^{n \times n}$ , where  $a_{ij} \in A$  is the corresponding weight in  $\mathcal{W}$ . If a graph is unweighted, then its edges' weights are considered to be 1. A graph is undirected if for all edges  $(v_i, v_j) \in \mathcal{E}$ ,  $(v_j, v_i) \in \mathcal{E}$ . A directed graph lacks this property. Finally, a graph is **connected** if there exists a path from any node to any other node. We assume that our graphs are weighted, connected, and undirected.

Graphs also have additional properties that can be computationally expensive to determine; therefore, they can only be performed on a graph that is small enough for these computations to be done quickly. In effect, 'small' is based on how much time the user has available to compute these properties. We will focus on **centrality measures**, which measure how central or important a given node or cluster of nodes is to the overall structure of the graph. A large number of such measures exists, each with a differing definition of importance. The two of particular importance to the methodology in Section 3 are **degree centrality** and **eigenvector centrality**.

**Definition 2.1** (Degree Centrality). *For a node  $v_i$  in an undirected graph, its **degree** is defined as the number of edges that originate from it. For a directed graph, we also define its **outdegree** and its **indegree**. The outdegree is simply the degree as just defined. The indegree is the number of edges that arrive at a given node. Note that for an undirected graph, degree, outdegree, and indegree are all equal for any given node.*

**Definition 2.2** (Eigenvector Centrality). *For a graph, the largest eigenvalue of the adjacency matrix is guaranteed to correspond to an eigenvector with all positive entries. For a node  $v_i$ , its **eigenvector centrality** is the  $i$ th entry in this all positive eigenvector (see [5]).*

### 3. IMMUNIZATION OF SMALL GRAPHS

**3.1. Methodology.** As discussed in Section 1, it has been claimed and debated that the overall structure of a graph is strongly predictive of the order in which nodes should be immunized. Specifically, despite the relative success from the usage of some centrality measures, it is demonstrable that for some cases, these measures fail significantly. In examining the literature, several questions became apparent, such as the accuracy of  $\varepsilon$  as a predictor of a virus spreading across a graph and any dependencies of  $\varepsilon$  on the virus's strength, the system's ability to recover, and the immunization's strength, as established in [7]. Therefore, this section begins by establishing a simulation upon which to work in Section 3.1.1. Then, in Section 3.1.2, we turn our focus towards calculating  $\varepsilon$  and its applications towards a brute force scheme, to which we can compare. Next, in Section 3.1.3, we examine  $\varepsilon$ 's relationship to immunization strength, viral strength, and recovery rate, before checking the accuracy of  $\varepsilon$  as a predictor of viral performance in 3.1.4. To do this, we analyze the relationship between  $\varepsilon$  and the number of times the virus wins by using various regressions for the overall data and for data drawn from specific immunization set sizes. We then create and discuss three immunization schemes in Section 3.1.5 before concluding by discussing the results of these schemes in Section 3.2.

**3.1.1. Simulation.** Our small scale simulation model is constructed using the following assumptions.

- (1) The virus follows a simple **SIS** model as discussed in Section 2.1.
- (2) The virus spreads too quickly for reactionary immunization. Hence, our emphasis is on a predictive defense, rather than a reactive defense. This corresponds to immunization before the flu season for people, rather than a cure for those already exposed.

- (3) Since we are modeling a computer system, we assume that the worst case scenario of wipe and reload is the only real recovery method for an infected computer until anti-virus software catches up technologically to the virus. Hence, there is no increase in resistance for recovered nodes.
- (4) Finally, as a simplifying assumption, we assume that we do not need to worry about the virus coming in from an outside source once the infection process has started.

Our simulation thus proceeds according to the following steps after all setup (e.g. immunizations) has been performed. Let  $\beta$  be the probability that an infection is successful, and let  $\delta$  be the probability that recovery is successful. Let  $P_\beta$  and  $P_\delta$  be the standard uniform random variables corresponding to likelihood of infection and recovery, respectively.

- (1) Begin with a random single node infected.
- (2) Identify all infected nodes.
- (3) If  $Fan = \text{all}$ , then for a given infected node:
  - (a) Find all neighbors of this node.
  - (b) Generate  $P_\beta$  for all such neighbors.
  - (c) If a neighbor's  $P_\beta < \beta$ , then we update the neighbor as infected.
- (4) If  $Fan \in \mathbb{N}$ , then for a given infected node:
  - (a) Generate a series of  $P_{\beta_i}$  for  $i = 1, 2, \dots, Fan$ .
  - (b) Randomly determine which neighbors the  $P_{\beta_i} < \beta$  correspond to, discarding the others.
  - (c) Update these neighbors to be infected.
- (5) Then for all infected nodes, generate a  $P_\delta$ . If  $P_\delta < \delta$ , then the infected node recovers.
- (6) Collect data for a given round and repeat from step 2 until the virus is completely successful, completely removed, or a user chosen number of rounds have passed.

In order to remove bias, it is recommended that the infected nodes be traversed in a random order. Values of interest include order of infection, number of times infected, number of rounds, if the virus wins by infecting all non-immune nodes, and number of nodes a given node is responsible for infecting.

**3.1.2. Calculations of  $\varepsilon$  and a Brute Force Scheme.** Let  $A$  be the adjacency matrix, and let  $B = \text{diag}(\beta_i)$  and  $D = \text{diag}(\delta_i)$ , for  $i$  corresponding to the nodes in the graph. Additionally, let  $\lambda_1$  be the function that returns the largest eigenvalue of a matrix. As described in [7], when

$$\lambda_1(BA - D) \leq -\varepsilon < 0,$$

$\varepsilon$  can be used to establish a bound on the exponential decay of a virus in a graph. Henceforth, when we refer to  $\varepsilon$ , we are actually referring to this bounding equation. This bound was used to comparatively establish the effectiveness of the immunization schemes for small networks. Note that there is a weakness in the calculation of  $\varepsilon$ ; the calculation relies on an in-depth knowledge of the virus's ( $B$ ) and recovery mechanism's ( $D$ ) effects on individual nodes. In predictive immunization, it is not necessarily guaranteed that this knowledge is available.

Using  $\varepsilon$ , a brute force immunization scheme can be created. Let  $t$  be the number of nodes to immunize. Then:

**Scheme 3.1** (Brute Force).

- (1) Create a list of all combinations of  $t$  nodes of the graph.
- (2) For each combination, treat the nodes as immunized and calculate  $\varepsilon$ .
- (3) Pick the combination with the least  $\varepsilon$  value.

As with any brute force scheme, this is computationally expensive. Note that a combination with the best  $\varepsilon$  at a step  $t$  may be unrelated to a best combination at  $t + 1$ . The best combination, for example, on  $t = 3$  may be  $\{1, 5, 10\}$ , but on  $t = 4$ , it may be  $\{3, 4, 7, 8\}$ . As such, it is frequently found that for a sequential algorithm, it is impossible to match the brute force recommendation, as each calculation of the next choice is dependent upon the nodes already immunized. In such a case, at  $t = 3$ , the sequential algorithm may match the brute force with  $\{1, 5, 10\}$ , but proceed to  $\{1, 5, 7, 10\}$  on  $t = 4$ , with a substantial loss of relative effectiveness. Note also that if an algorithm does a node by node immunization, with a brute force decision made on each step, then that algorithm is a greedy immunization scheme, as proposed in [7].

3.1.3. *Dependencies of  $\varepsilon$ .* Recall from Section 3.1.2 that the  $\varepsilon$  bound is determined by

$$\lambda_1(BA - D).$$

First, we assume that all nodes have the same initial infection probabilities before immunization and that all nodes gain the same level of resistance due to immunization. Let  $\underline{\beta}$  stand for the infection probability of an immunized node (when the infection likelihood is at its lowest). Then it follows that  $\bar{\beta}$  stands for the infection probability of an un-immunized node (when the infection likelihood is at its highest). Together, these form the  $\beta$ s in Section 3.1.1. Finally, let  $\delta$  stand for the likelihood of recovery, as in Section 3.1.1.

The question that we wish to answer is what happens to  $\varepsilon$  when we modify these values. As an example, we take the Experimentation Matrix from Appendix A. This matrix is described in Section 3.1.4. The graphs in Figure 2 were generated with default values of  $\underline{\beta} = 0.001$ ,  $\bar{\beta} = 1$ , and  $\delta = \frac{1}{3}$ . Each graph takes two of these values to be constant and modulates the other from 0.001 to 1 in steps of 0.001. These graphs use the arbitrarily chosen immunization set  $\{1, 2, 4, 5, 7, 8, 9, 11\}$ .

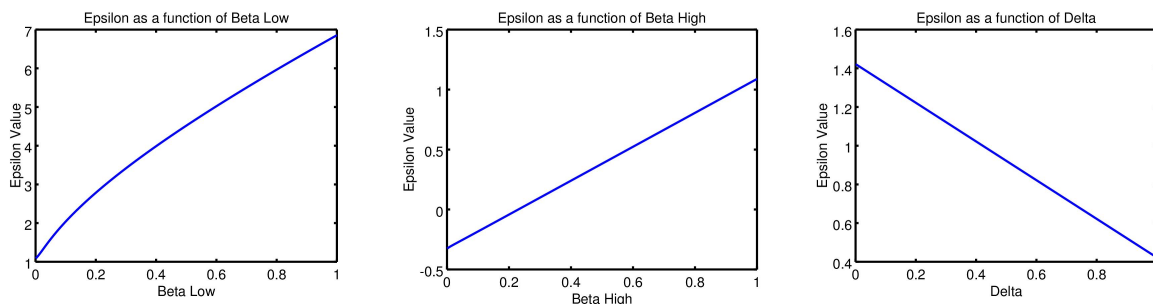


FIGURE 2. Effects of varying the parameters  $\underline{\beta}$ ,  $\bar{\beta}$ , and  $\delta$  on  $\varepsilon$ .

Figure 2 shows that for a given set combination, the relationship for any parameter is roughly or exactly linear, with goodness-of-fit values  $R^2$  of  $R_{\underline{\beta}}^2 = 0.9896$ ,  $R_{\bar{\beta}}^2 = 1$ ,  $R_{\delta}^2 = 1$ .

Next, we will examine the change in the “best” sets when these parameters are varied. We now specify a size of 8 and choose the set for the smallest values of  $\varepsilon$ . These sets are represented by the differently colored segments of the curves in Figure 3.

First we note that the last image in Figure 3 uses a single set for the entire line. We conjecture that, if recovery could be modified on a node by node basis with  $\bar{\delta}$  and  $\underline{\delta}$  values, it would result in a similar situation as with the  $\beta$  values. Hence, it appears likely that  $\varepsilon$  depends on both  $\beta$  and  $\delta$  in similar ways. Variation of  $\underline{\beta}$  has the most noticeable effect on

## B. FAGAN

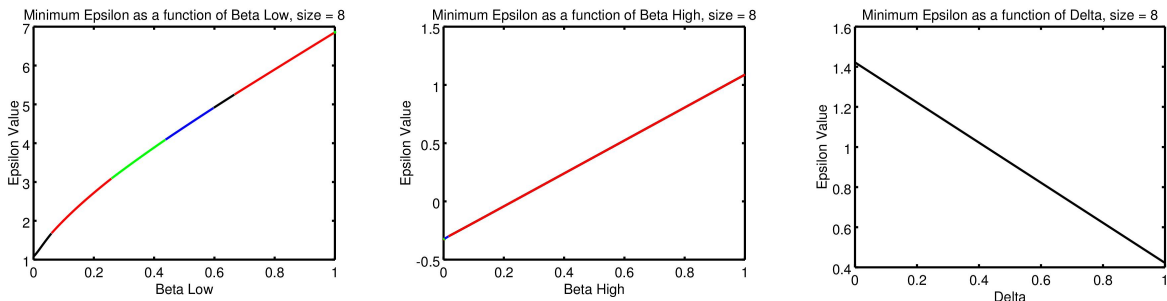


FIGURE 3. Effects of varying the parameters  $\underline{\beta}$ ,  $\bar{\beta}$ , and  $\delta$  on the immunization set corresponding to the lowest  $\varepsilon$ .

which choices result in the smallest  $\varepsilon$ . Variation of  $\bar{\beta}$  only has strong effects on the optimal choices near 0.001. We conjecture that this is due to proximity with  $\underline{\beta}$ .

Figure 3 also helps demonstrate the possible changes in best choices. It is likely that there exist graphs whose best choices change much more and much less frequently than this example graph's do, increasing the difficulty of accurate decision making. This additionally highlights the usefulness of the greedy immunization scheme used in [7].

As a final point of contrast, three brute force immunizations are presented in Tables 1-3. Included are the parameters for these immunizations, the best set for a given step sorted by the size of the immunization set, and the corresponding  $\varepsilon$  value. Note that there are many other possible solutions and that some solutions may be identical to solutions for other parameter combinations. Recall that when the  $\varepsilon$  value is negative, we are guaranteed exponential decay of the virus.

$\underline{\beta}$	$\bar{\beta}$	$\delta$	Size 1	Size 2	Size 3	Size 4	Size 5	Size 6	Size 7
0.001	1	$\frac{1}{3}$	{15}	{8, 15}	{2,8,9}	{1,2,8,9}	{1,2,5,8,9}	{1,2,4,8,9,15}	{2,4,5,7,8,9,11}
			6.05382	5.26975	4.43918	3.70066	3.01748	2.45853	1.77683
0.5	1	$\frac{1}{3}$	{15}	{8, 15}	{2,8,15}	{2,8,9,15}	{2,7,8,9,15}	{2,6,7,8,9,15}	{2,5,6,7,8,9,15}
			6.46485	6.07987	5.75349	5.42941	5.16656	4.91523	4.64014
0.001	0.5	$\frac{1}{3}$	{15}	{8, 15}	{2,8,9}	{1,2,8,9}	{1,2,5,8,9}	{1,2,4,8,9,15}	{2,4,5,7,8,9,11}
			2.86066	2.46904	2.05450	1.68593	1.34554	1.06614	0.72631

TABLE 1. This table shows the immunization sets resulting from different parameter values and the lowest  $\varepsilon$  values for sets of sizes 1-7.

Finally, Tables 1-3 demonstrate the impact of the changes in  $\underline{\beta}$  and  $\bar{\beta}$ . Changing  $\bar{\beta}$  changes only the corresponding  $\varepsilon$  value for a given choice. On the other hand, changing the effectiveness of the immunization,  $\underline{\beta}$ , changes a number of the nodes chosen to be immunized. These two trends hold for most values. The value of  $\bar{\beta}$  begins to affect nodes chosen when it approaches the value of  $\underline{\beta}$ . Note that many choices remain the same despite the change in parameters.

This section has shown the strength of obtaining information about the virus and the problem of relying on just the graph's organization. In contrast to this, it is important to note that much of the information used is not necessarily available. In a situation where a virus has not been well studied, the relevant parameters may be unknown and impossible to

$\underline{\beta}$	$\bar{\beta}$	$\delta$	Size 8	Size 9	Size 10	Size 11
0.001	1	$\frac{1}{3}$	{1,2,4,5,7,8,9,11}	{2,5,6,7,8,9,11,14,15}	{2,4,5,6,7,8,9,11,14,15}	{1,2,4,5,6,7,8,9,10,11,15}
			1.08861	0.67669	-0.18942	-0.22152
0.5	1	$\frac{1}{3}$	{2,5,6,7,8,9,11,15}	{2,4,5,6,7,8,9,11,15}	{1,2,5,6,7,8,9,10,11,15}	{1,2,4,5,6,7,8,9,10,11,15}
			4.41304	4.17786	3.97542	3.76926
0.001	0.5	$\frac{1}{3}$	{1,2,4,5,7,8,9,11}	{2,5,6,7,8,9,11,14,15}	{2,4,5,6,7,8,9,11,14,15}	{1,2,4,5,6,7,8,9,10,11,15}
			0.38164	0.17672	-0.23099	-0.25371

TABLE 2. This table shows the immunization sets resulting from different parameter values and the lowest  $\varepsilon$  values for sets of sizes 8-11.

$\underline{\beta}$	$\bar{\beta}$	$\delta$	Size 12	Size 13	Size 14
0.001	1	$\frac{1}{3}$	{1,2,4,5,6,7,8,9,10,11,13,15}	{1,2,4,5,6,7,8,9,10,11,12,13,15}	{1,2,3,4,5,6,7,8,9,10,11,12,13,15}
			-0.24149	-0.26116	-0.26900
0.5	1	$\frac{1}{3}$	{1,2,4,5,6,7,8,9,10,11,13,15}	{1,2,3,4,5,6,7,8,9,10,11,13,15}	{1,2,3,4,5,6,7,8,9,10,11,12,13,15}
			3.59728	3.46860	3.3347
0.001	0.5	$\frac{1}{3}$	{1,2,4,5,6,7,8,9,10,11,13,15}	{1,2,4,5,6,7,8,9,10,11,12,13,15}	{1,2,3,4,5,6,7,8,9,10,11,12,13,15}
			-0.26784	-0.28176	-0.28749

TABLE 3. This table shows the immunization sets resulting from different parameter values and the lowest  $\varepsilon$  values for sets of sizes 12-14.

estimate. While a greedy approach is strong in well studied cases, an approach based on a graph's organization would provide a broader application.

3.1.4. *Experimentation on  $\varepsilon$ .* In order to validate the applicability of the  $\varepsilon$  parameter, we compare  $\varepsilon$  against both weighted and unweighted graphs and the effects of a virus. The conditions for this experiment were as follows:

- A graph of size 15, with an average number of approximately 7 edges per node, was randomly generated. The graph's weighted adjacency matrix can be seen in Appendix A. We use a single matrix here and in Section 3.1.3 in order to make controlled changes. This varies from our approach in Section 3.2.
- Using this graph, every possible combination of nodes for immunization was generated and assigned an  $\varepsilon$  value, using the parameters of  $\bar{\beta} = 1$ ,  $\underline{\beta} = 0.001$ , and  $\delta = \frac{1}{3}$ .
- Then the simulation was performed 250 times for each combination, as detailed in Section 3.1.1, with  $Fan = \text{all}$ .
- The following test cases were used to demonstrate the importance of weighting on a graph:
  - Case 1:** The graph was tested as unweighted, while the  $\varepsilon$  calculation used the unweighted adjacency matrix.
  - Case 2:** The graph was tested as weighted, while the  $\varepsilon$  calculation used the unweighted adjacency matrix.

**Case 3:** The graph was tested as weighted, while the  $\varepsilon$  calculation used the weighted adjacency matrix.

Recall from 3.1.2 that  $\varepsilon$  is bounded by  $\lambda_1(BA - D)$ , where  $A$  is specified as the adjacency matrix. In [7], this  $A$  was the *unweighted* adjacency matrix, which is determined by whether contact occurs between two nodes or not. In contrast, a computer network can more precisely determine the amount of contact that occurs between two computers. This information could be used to assign weights to the graph of the computer network.

As noted in Section 3.1.1, one parameter of interest is the number of times the virus wins, or how many times it infects every non-immune node in the graph. We note that no node is immune under the circumstances of our test. This value was then compared to  $\varepsilon$  using [8]. The general results for the unweighted-unweighted case are shown in Figure 4.

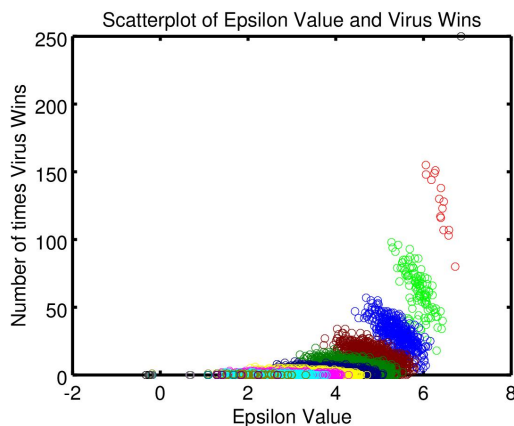


FIGURE 4. This plot of  $\varepsilon$  and the number of times the virus wins uses data drawn from Case 1: Unweighted-Unweighted. Note that points of the same color are drawn from immunization sets of the same size.

We seek to determine the nature of the relationship between the value of  $\varepsilon$  and the number of times the virus wins. To do this, we use R, a statistical computation software (see [8]). Three different regressions were used to fit a model to the data, as described in [4]:

- linear regression,
- Poisson regression, and
- zero-inflated Poisson regression.

A linear regression seeks to test whether a linear relationship exists between the value of  $\varepsilon$  and the number of times the virus wins. The Poisson regression instead tests whether the relationship is better fit by a Poisson model. Finally, the zero-inflated Poisson regression tests using two models by assuming that some of the zeros can be modeled separately from “true” zeros, with all remaining values modeled by a Poisson model. We present the models and their probability values. In each model, we test against the null hypothesis that the parameter has no statistically significant impact. The tables in Figure 5 show the results.

Notably, each cases’ models are extremely similar to one another. As such, we consider the unweighted-unweighted case primarily. Between the Poisson and zero-inflated Poisson models, we can apply Vuong’s test, as demonstrated in R in [4], to determine if either model is superior. As seen in Table 4, the zero-inflated Poisson model is superior for all cases. Next, we examine the linear and zero-inflated models in Figure 6. This figure was generated by



Case 1: Unweighted-Unweighted

Model:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	-6.47	-5.53	-3.88, 3.68
Intercept p-value	0.00	0.00	0.00, 0.00
$\varepsilon$	2.67	1.56	1.26, -1.11
$\varepsilon$ p-value	0.00	0.00	0.00, 0.00

Case 2: Weighted-Unweighted

Model:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	-6.54	-5.55	-3.90, 3.67
Intercept p-value	0.00	0.00	0.00, 0.00
$\varepsilon$	2.70	1.57	1.27, -1.11
$\varepsilon$ p-value	0.00	0.00	0.00, 0.00

Case 3: Weighted-Weighted

Model:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	-6.48	-5.54	-3.89, 3.65
Intercept p-value	0.00	0.00	0.00, 0.00
$\varepsilon$	2.68	1.57	1.26, -1.10
$\varepsilon$ p-value	0.00	0.00	0.00, 0.00

FIGURE 5. These tables present the results of regressions applied to the three cases of weighted and unweighted graphs and weighted and unweighted  $\varepsilon$  calculations.

using the predict function in R. It is readily observed that the zero-inflated Poisson regression line provides a more accurate fit. Returning to Figure 4, we note that the immunization sets, when compared by size, appear to have negative linear relationships. To investigate this, we next consider the  $\varepsilon$  values corresponding to immunization sets of a specific size.

Poisson v. Zero-Inflated:	Raw	AIC-Corrected	BIC-Corrected
Unweighted-Unweighted	<, 0.00	<, 0.00	<, 0.00
Weighted-Unweighted	<, 0.00	<, 0.00	<, 0.00
Weighted-Weighted	<, 0.00	<, 0.00	<, 0.00

TABLE 4. This table shows the results of Vuong’s test, which can be used to compare the effectiveness of the Poisson and zero-inflated Poisson models. In all cases, the zero-inflated Poisson model is more effective.

We use data from immunization sets of sizes 5-9 drawn from Case 1: Unweighted-Unweighted. The distributions of  $\varepsilon$  for sizes 5-9 are shown in Figure 7. It is clear that as the size increases, the distribution of  $\varepsilon$  shifts left and the number of times the virus wins decreases on average. It is also evident that the distributions are likely to be more poorly modeled by the regressions used earlier. Modeling the number of wins of the virus as dependent on the

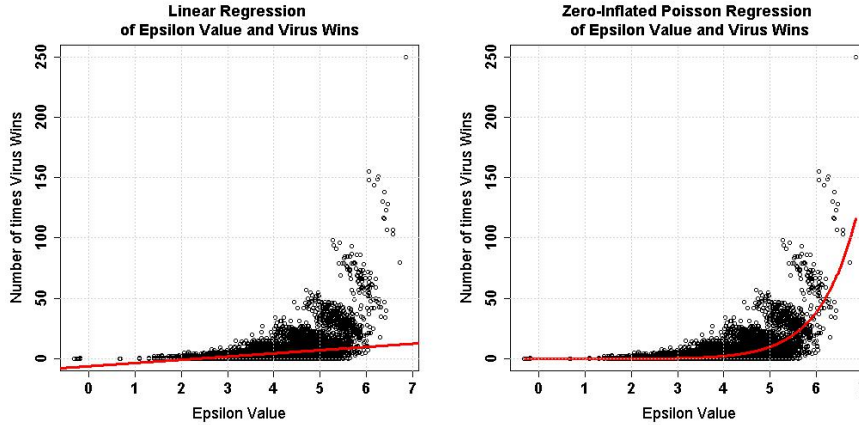


FIGURE 6. The predictions from linear and zero-inflated Poisson models are superimposed on a plot of  $\epsilon$  and the number of times the virus wins. It is obvious that the zero-inflated Poisson model has a more accurate fit.

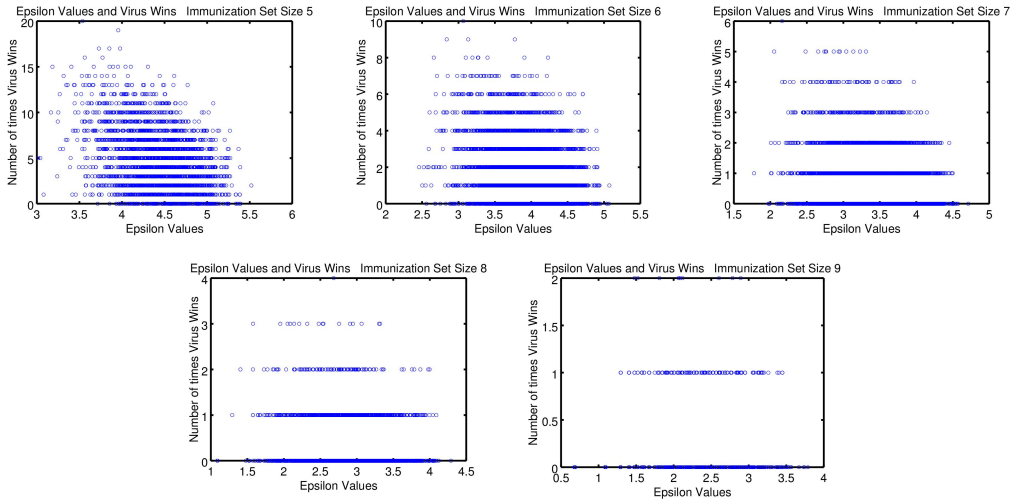


FIGURE 7. These graphs show the distribution of  $\epsilon$  and the corresponding number of wins of the virus for immunization sets of sizes 5-9.

value of  $\epsilon$  using the previous regressions, we do indeed observe that the models' parameters become less statistically significant as the size increases. The results are compiled in Table 5. These models also do not entirely resemble the overall models in that the signs of the values are opposite for any given parameter, except for the Poisson and zero-inflated Poisson models for immunization sets of size 9. Given that the immunization sets of size 9 resulted in the most binary distribution of virus wins in Figure 7, this comes as no surprise.

As the size of the immunization set increases, the models notably become less effective. All results are statistically significant with  $p < 0.001$  except for immunization sets of size 8 (intercept) and size 9 ( $\epsilon$  significant at the 0.1 level). This is due to the increasingly binary nature of the ability of the virus to win, which severely challenges the effectiveness of linear and Poisson regressions. Furthermore, the magnitude of the effect of  $\epsilon$  decreases as the size increases. This implies that the effect of  $\epsilon$  for later choices becomes less important than for

Size 5:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	17.85	4.22	3.99, -8.90
Intercept p-value	0.00	0.00	0.00, 0.00
$\varepsilon$	-2.96	-0.61	-0.54, 1.35
$\varepsilon$ p-value	0.00	0.00	0.00, 0.00
Size 6:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	6.13	2.97	2.76, -4.14
Intercept p-value	0.00	0.00	0.00, 0.00
$\varepsilon$	-1.12	-0.62	-0.52, 0.64
$\varepsilon$ p-value	0.00	0.00	0.00, 0.00
Size 7:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	1.85	1.51	1.80, -1.05
Intercept p-value	0.00	0.00	0.00, 0.095
$\varepsilon$	-0.37	-0.60	-0.61, -0.029
$\varepsilon$ p-value	0.00	0.00	0.00, 0.882
Size 8:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	0.53	0.14	0.33, -1.33
Intercept p-value	0.00	0.433	0.339, 0.169
$\varepsilon$	-0.12	-0.65	-0.58, 0.21
$\varepsilon$ p-value	0.00	0.00	0.00, 0.553
Size 9:	Linear	Poisson	Zero-Inflated (Poisson, Zero-Inflated)
Intercept	0.079	-2.38	-0.89, 1.94
Intercept p-value	0.00	0.00	0.33, 0.266
$\varepsilon$	-0.014	-0.30	-0.70, -1.00
$\varepsilon$ p-value	0.0325	0.0298	0.037, 0.227

TABLE 5. This table has the three regression models applied to  $\varepsilon$  values corresponding to immunization sets of sizes 5-9.

earlier choices. Indeed, comparing the number of times the virus wins as a function of  $\varepsilon$  corroborates this, as seen in Figure 4. This is also evident from the change of the intercept’s values in all models in Table 5. It is readily apparent from Figure 4 and Table 5 that as  $\varepsilon$  decreases, both the number of times the virus wins and the rate of change of this value decrease. Contrary to one’s expectations however, and as reflected by the statistical models (see the sign of the  $\varepsilon$  values in Table 5), if the size of the immunization set is fixed, it appears that as  $\varepsilon$  *increases*, the virus wins *less often*. As such, minimizing the  $\varepsilon$  value may actually help the virus to succeed.

We examine one additional regression in order to corroborate this result and generalize it to the whole data set. Consider the ordinal probit regression, again using R, in which whether the virus wins *at all* for a given immunization set is predicted by the value of  $\varepsilon$  and

the size of the immunization set. Then we obtain the values seen in Table 6. All values are statistically significant with  $p < 0.001$ , with proportionally small standard errors. Examining the estimates of the coefficients, we can see that the probability of the virus’s winning is *less* when more nodes have been immunized or when the value of  $\varepsilon$  is high. This is consistent with our results seen in Table 5.

	Estimate	Std. Error	$p$ -value
Intercept	8.35	0.16	0.00
$\varepsilon$ Value	-0.42	0.02	0.00
Immunization Set Size	-1.02	0.01	0.00

TABLE 6. This table contains the coefficient’s values, standard errors, and  $p$ -values for a probit regression.

3.1.5. *Centrality Experimentation.* In this section, we will discuss the creation of some immunization schemes based on the usage of centrality measures. The effectiveness of these schemes will be discussed in Section 3.2. A variety of centrality measures were examined in order to determine if there was a correlation between the measurements and the “best choices.” Here, we define the best choices as the number of times a strict brute force scheme would choose a node to be in an immunization set of some size across all sizes. Recall that such a scheme seeks to *minimize*  $\varepsilon$ , as in [7]. Given the lengthy computation time, graphs of sizes 10, 15, and 20 were chosen for the initial sample with simplifying assumptions of  $\bar{\beta} = 1$ ,  $\underline{\beta} = 0.001$ , and  $\delta = 0$  for all nodes. It was determined very quickly that both the literature scheme in [7] and the brute force scheme correspond relatively well with the degree for the first few choices. Of those nodes with high degree, it was often, but not always the case, that they had high eigenvector centrality. Other centrality measures were examined, but in subsequent testing, these failed to provide strong results. Thus, degree and eigenvector centralities were used as the basis for the initial immunization scheme.

As the number of nodes immunized increases, degree centrality corresponds less to the best choices. It was conjectured that, from the virus’s point of view, the graph’s immunized nodes would appear to be disconnected from the graph. Hence, the next iteration of the immunization scheme examined what happened when nodes that were immunized were removed from the graph. The following scheme forms the basis for our first set of results in Section 3.2. Note that a “sufficient number” reflects reaching the user’s desired number of immunizations. Recall from Section 3.1.2 that, as described in [7], we have bounding of exponential decay only when  $\lambda_1(BA - D) \leq -\varepsilon < 0$ .

### Scheme 3.2.

- (1) *Begin with no nodes immunized.*
- (2) *Let  $A$  be the initial adjacency matrix of the graph.*
- (3) *Let  $A_t = A$ .*
- (4) *Take  $D = \{v \in A : Degree(v) = \max(Degree(A_t))\}$  and order  $D$  by Eigenvector( $A_t$ ) in descending order.*
- (5) *Immunize the node  $v_t = d_1 \in D$ , where  $d_1$  is the first node in  $D$ .*
- (6) *Assign  $A_{t+1} = A_t \setminus \{v_t\}$ .*
- (7) *Repeat (4) through (6) until a sufficient number of nodes are immunized or*

$$\max(Degree(A_{t+1})) = 0.$$

- (8) *Immunize the remaining nodes in descending order of Eigenvector( $A$ ) until a sufficient number of nodes are immunized.*

As seen in Section 3.2, the results for this initial scheme were strong. The failures of this initial scheme appeared most commonly in the last few choices. In order to improve this scheme, Step 8 was examined. A brute force scheme was used to replace Step 8. It became readily apparent that, in order to improve the last choices, it was necessary to improve the prior choices, as the brute force scheme showed little to no improvement.

In order to determine when to begin Step 8, Scheme 3.2 was run with a variety of other computable values, found in [1]. Of these, the clustering coefficient, defined as the number of cycles of length three divided by the number of connected triples, was noted to have a precipitous decline before the general results of Scheme 3.2 fell off. We use this as the replacement for the condition in Step (7), forming Scheme 3.3.

**Scheme 3.3.**

- (1) *Begin with no nodes immunized.*
- (2) *Let  $A$  be the initial adjacency matrix of the graph.*
- (3) *Let  $A_t = A$ .*
- (4) *Take  $D = \{v \in A : Degree(v) = \max(Degree(A_t))\}$  and order  $D$  by Eigenvector( $A_t$ ) in descending order.*
- (5) *Immunize the node  $v_t = d_1 \in D$ , where  $d_1$  is the first node in  $D$ .*
- (6) *Assign  $A_{t+1} = A_t \setminus \{v_t\}$ .*
- (7) *Repeat (4) through (6) until a sufficient number of nodes are immunized or*  

$$ClusteringCoefficient(A_t) \notin \mathbb{R}_+ \setminus \{0\}$$
- (8) *Immunize the remaining nodes in descending order of Eigenvector( $A$ ) until a sufficient number of nodes are immunized.*

Additionally, one other scheme on small organizations was attempted. In contrast to the preceding schemes, this scheme focuses on a divide and conquer approach.

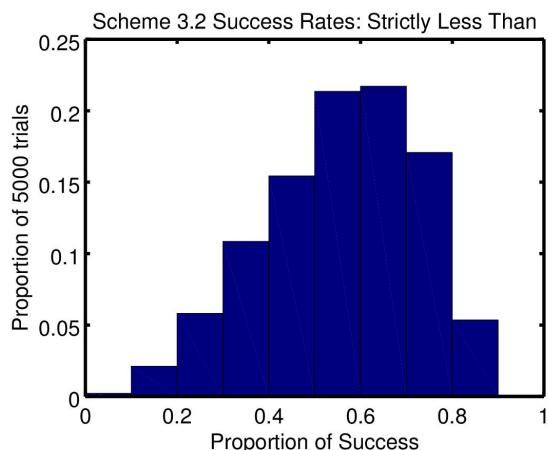
**Scheme 3.4.**

- (1) *Begin with no nodes immunized.*
- (2) *Let  $A$  be the initial adjacency matrix of the graph.*
- (3) *Let  $A_t = A$ .*
- (4) *Take  $D = \{v \in A : Degree(v) = \max(Degree(A_t))\}$  and order  $D$  by Eigenvector( $A_t$ ) in descending order.*
- (5) *Immunize the node  $v_t = d_1 \in D$ , where  $d_1$  is the first node in  $D$ .*
- (6) *Determine if  $A_t \setminus \{v_t\}$  is connected. If it is not, break it into its connected subgraphs.*
- (7) *Of all connected subgraphs, let  $A_{t+1}$  be the graph with the most nodes.*
- (8) *Repeat steps (4) through (7) until a sufficient number of nodes are immunized.*

The idea behind Scheme 3.4 was to modify Scheme 3.2 such that it no longer needed a failure condition and could be applied on subgraphs. The comparison of the results between all three schemes is discussed in Section 3.2.

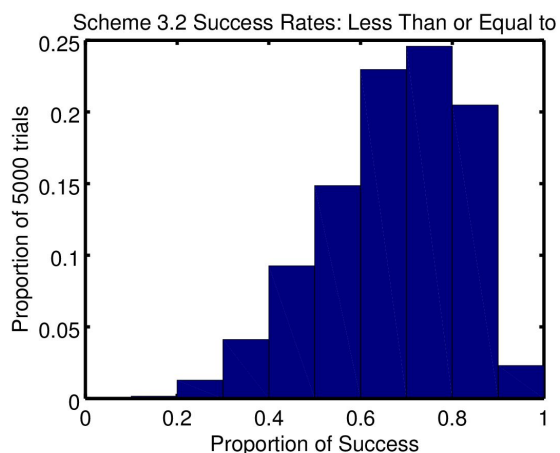
**3.2. Results.** For the performance analysis using  $\varepsilon$ , we created 5000 randomly generated connected graphs of size 50, with 25 as the average degree of the nodes. We adopted this approach in order to reduce the possibility of the comparisons being done on biased graphs,

as some test trials indicated a much stronger performance for one scheme over another on a specific graph. Additionally, we used  $\bar{\beta} = 1$ ,  $\underline{\beta} = 0.001$ , and  $\delta = 0$ . This represents a virus for which the population has no natural resistance and no means of recovery. In order to compare the schemes proposed, we compared each to the “reverse greedy” scheme proposed in [7] twice, the first awarding equality in favor of the literature’s scheme. The comparison proceeds by allowing each scheme to choose one node to immunize, and then calculating  $\varepsilon$  for the entire graph, as in accordance with the previous literature. Dividing the number of times the proposed schemes  $\varepsilon$  is less than, in the first comparison, or less than or equal to, in the second, the literature schemes  $\varepsilon$  by the total number of runs gives that trials proportion of success, giving that trial’s proportion of success. This can be seen in Figures 8, 10, and 12 for the case when the comparison is strictly less than, and in Figures 9, 11, and 13 for the case when the comparison is less than or equal to. Note that these figures and results do not measure *relative* success; how far apart the two  $\varepsilon$  values were did not influence the result.



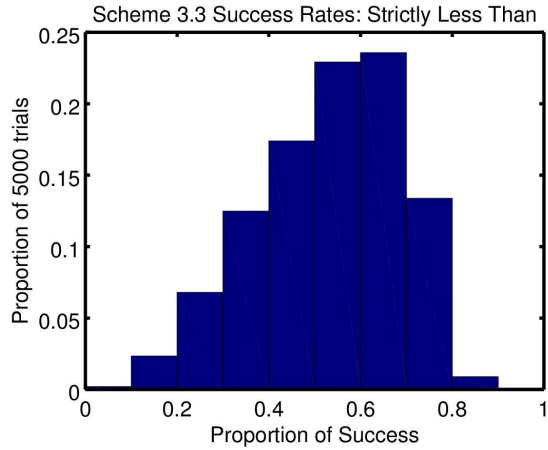
minimum	0.0400
1st quartile	0.4600
median	0.5800
3rd quartile	0.7000
maximum	0.9200
mean	0.5677
std. dev.	0.1690
results < .15	0.0064
results > .5	0.6554
results > .9	0.0002

FIGURE 8. This figure shows the results of a strictly better performance analysis with quartile analysis using Scheme 3.2.



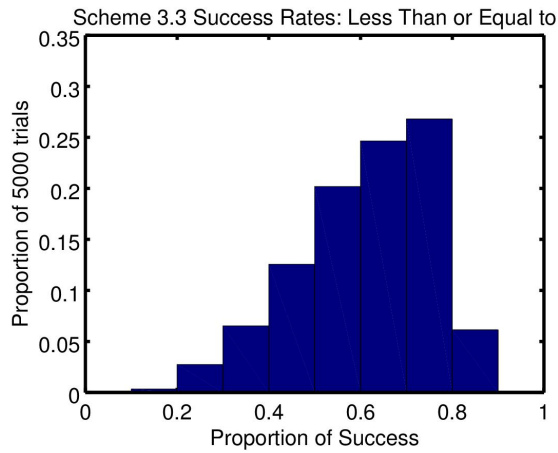
minimum	0.1400
1st quartile	0.5800
median	0.7000
3rd quartile	0.8000
maximum	0.9800
mean	0.6811
std. dev.	0.1518
results < .15	0.0006
results > .5	0.8518
results > .9	0.0230

FIGURE 9. This figure shows the results of a better or equal performance analysis with quartile analysis using Scheme 3.2.



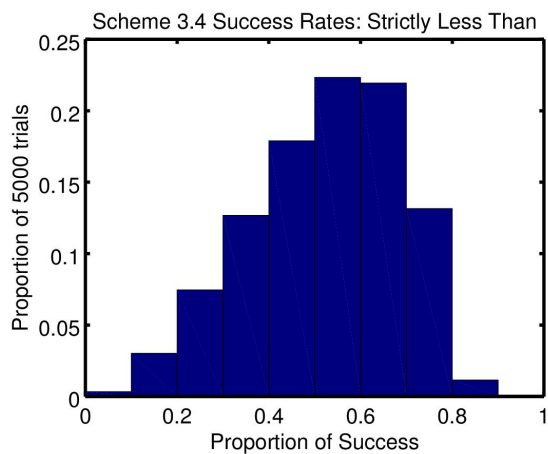
minimum	0.0400
1st quartile	0.4400
median	0.5600
3rd quartile	0.6600
maximum	0.9000
mean	0.5396
std. dev.	0.1568
results < .15	0.0076
results > .5	0.6078
results > .9	0.0000

FIGURE 10. This figure shows the results of a strictly better performance analysis with quartile analysis using Scheme 3.3.



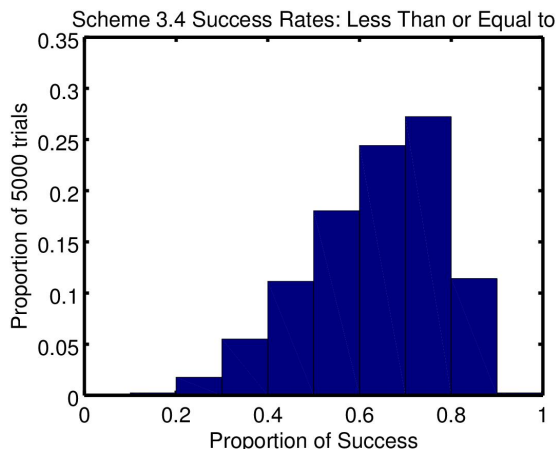
minimum	0.0800
1st quartile	0.5200
median	0.6400
3rd quartile	0.7400
maximum	0.9200
mean	0.6219
std. dev.	0.1454
results < .15	0.0016
results > .5	0.7782
results > .9	0.0006

FIGURE 11. This figure shows the results of a better or equal performance analysis with quartile analysis using Scheme 3.3.



minimum	0.0400
1st quartile	0.4200
median	0.5400
3rd quartile	0.6600
maximum	0.8800
mean	0.5322
std. dev.	0.1624
results < .15	0.0104
results > .5	0.5860
results > .9	0

FIGURE 12. This figure shows the results of a strictly better performance analysis with quartile analysis using Scheme 3.4.



minimum	0.1200
1st quartile	0.5600
median	0.6600
3rd quartile	0.7600
maximum	0.9800
mean	0.6462
std. dev.	0.1457
results < .15	0.0010
results > .5	0.8136
results > .9	0.0024

FIGURE 13. This figure shows the results of a better or equal performance analysis with quartile analysis using Scheme 3.4.

To determine which scheme is superior, we apply a two-sample  $t$ -test for unequal variances (see [11] for details) to Scheme 3.2 and Schemes 3.4 and 3.3. This test assumes that the samples are drawn from populations with unequal variance and tests for the difference of the samples' respective means. We choose this test due to the fact that the results from Scheme 3.2 have unequal variance to the results from Schemes 3.3 and 3.4 using the variance test in R: in the strictly less than cases  $p = 0.00$  and  $p = 0.01717$  and in the less than or equal to cases  $p = 0.002515$  and  $p = 0.003596$ .

Between Schemes 3.2 and 3.4, for the strictly less than case, the 95% confidence interval is  $[0.0291, 0.0420]$ . For the less than or equal to case, the 95% confidence interval is  $[0.0291, 0.0408]$ . Since both confidence intervals do not include 0 and are positive, we reject the null hypothesis that the difference of the population means is 0; that is, we reject the hypothesis that the two schemes are equally effective. Additionally, this test demonstrates that Scheme 3.2 is a stronger scheme. The authors conjecture that this is due to the effect of considering all possible next selections. For comparison, in Scheme 3.4, only the nodes that remain in the largest subgraph are under consideration. As such, it is likely that it is possible to improve Scheme 3.4 by examining the graph in a more holistic way, as Scheme 3.2 does.

Next, we apply a two-sample  $t$ -test for unequal variances on Scheme 3.2 and Scheme 3.3. We see that for the strictly less than case that the confidence interval is  $[0.0217, 0.0345]$ , and for the less than or equal to case it is  $[0.0534, 0.0650]$ . From both of these intervals, a similar analysis shows that Scheme 3.2 is the stronger scheme. The authors conjecture that this result is due to an unknown weakness within the eigenvector centrality used in Step 8. This conjecture stems from the fact that in our trials, it was observed that Scheme 3.3 ended earlier than Scheme 3.2, making the former more reliant on eigenvector centrality than the latter.

Lastly, we consider the likelihood that our results are statistically significant compared to the greedy scheme suggested in [7]. We perform a one-sample  $t$ -test on the strictly less than case in R and obtain the 95% confidence interval  $[0.5631, 0.5724]$ . Here, the null hypothesis is that the population mean would otherwise be 0.5, representing a 50% success rate and equivalency between the two schemes. Hence, we reject this null hypothesis and conclude that our scheme is not equivalent to the one in [7]. Since the 95% confidence interval's



lower bound is greater than 0.5, we have made statistically significant improvements on the literature’s suggested scheme of minimizing  $\varepsilon$ .

#### 4. CONCLUSION

In this paper, we have introduced a simulation for a virus in a network. Using the simulation described in Section 3.1.1, we demonstrated the impact of a virus’s parameters on the value of  $\varepsilon$ , showing that  $\varepsilon$  increases as a virus becomes more effective in spreading, as immunization becomes less effective, and as recovery becomes less likely across an entire graph. In contrast, for a given graph and immunization set size, as in Figure 4, we observe that as  $\varepsilon$  *increases*, the number of times the virus wins *decreases*. Indeed, *increasing*  $\varepsilon$  predicts a *reduced* probability of a viral win, as shown in Table 6 for the general case and Table 5 for specific immunization set sizes. This confounding effect implies that  $\varepsilon$  should be maximized, rather than minimized, contradicting the recommendation in [7], which followed from what one would expect given the effects of a virus’s parameters on  $\varepsilon$ . As such, as a predictor  $\varepsilon$  should be used carefully. Finally, three alternative schemes were proposed as methods of minimizing  $\varepsilon$ , in accordance with established literature. Of these schemes, Scheme 3.2 was shown to be able to outperform the other two schemes as well as the reverse greedy immunization scheme proposed in [7]. Hence, we conclude that graph organization can be used as a strong predictor of  $\varepsilon$  in order to produce more effective immunization schemes, especially for viruses for which there is no existing knowledge nor recovery mechanism. This is with the caveat that  $\varepsilon$  may perform the opposite role as that cited in the literature.

#### 5. FUTURE WORK

Areas that are of future interest include:

- refinement of the zero-inflated Poisson model in Figure 6 so as to better fit the data,
- use of real graphs and data sets to refine and validate results,
- addition and examination of recovery sets, as opposed to a general blanket recovery,
- improved brute force time requirement (if possible), and
- additional understanding of the usage of  $\varepsilon$  as a means of prediction.

Regarding the first item, ideally, we should be able to construct a model that can accurately predict the effectiveness of certain choices in immunizations. For this, the second item may be necessary in order to gather a large amount of realistic testing data. This would allow for verification of the existing models seen in Section 3.1.4, and, ideally, a refinement and combination of the models. We suspect it is possible to gather such data using low cost computers that are currently available on the market. Through clever networking and programming, it is extremely likely that a set of such computers would be able to report the growth of such an infection to some central computer in a modular way, allowing for a variety of tests to be conducted. This would also, conveniently, aid us in addressing the third item.

With respect to the fourth item, it need only be said that asymptotic analysis of algorithms has gifted us with many fine algorithms. If a quality brute force algorithm exists and is known for immunizing graphs, it would be of great use in the construction and comparison of immunization schemes.

Of special note is the last item. Our results show that there is a distinct danger in using the  $\varepsilon$  value as a predictor of the decay of the virus. Indeed, some of the results of this paper are counter-intuitive; the  $\varepsilon$  parameter scales as expected with viral parameters, but does not predict the success of the virus as expected. If there is an ideal parameter that could be

used to predict immunization effectiveness, many other solutions could be used. Suppose such a parameter  $\omega$  exists and consider the following variation of the optimal immunization problem.

Let  $\mathcal{I}$  be the Immunization Set Graph; that is, the graph representing all possible immunization sets. This graph  $\mathcal{I} = (\mathcal{V}_{\mathcal{I}}, \mathcal{E}_{\mathcal{I}})$  is a graph whose vertices represent immunization sets of a graph  $\mathcal{G}$ , and whose edges represent subset and superset functions, depending on direction. Each  $\mathcal{I}$  is thus dependent on a specific  $\mathcal{G}$ . Consider Figure 14. On the left, there is a graph which we call  $\mathcal{G}$ . Its Immunization Set Graph  $\mathcal{I}$  is to the right of it. Each vertex in  $\mathcal{I}$  represents immunizing a specific combination of nodes.

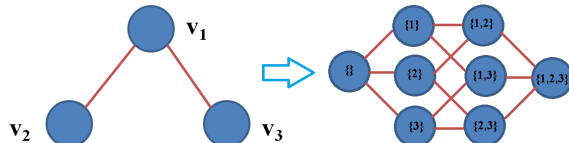


FIGURE 14. This figure shows a graph and its respective immunization set graph.

In order to begin using this for immunization schemes, we now consider a Weighted Immunization Set Graph. In addition to the above characteristics,  $\mathcal{I}$  now has  $\mathcal{W}_{\mathcal{I}}$ , composed of weights equal to the  $\omega$  value of immunizing that specific set of nodes. These weights can be assigned either to  $\mathcal{E}_{\mathcal{I}}$ , in which the weight is assigned to the edges entering a vertex, or to  $\mathcal{V}_{\mathcal{I}}$ , in which the weight is assigned to the node itself. Then, an optimal immunization scheme is one in which the lowest value path is taken, similar to a shortest path problem. In this light, the greedy algorithm proposed by Preciado et al. in [7] could be seen as a greedy shortest path algorithm. The complication to this is the calculation of the weights. Immunization schemes could be viewed as methods of estimating these weights. Furthermore, a field of algorithms already exists for shortest path problems; for example, apply Dijkstra's algorithm from [3] after a calculation of all weights. As such, estimations as seen earlier may provide ways to approximate the best path, which, when combined with work already done on graphs and shortest paths, may in the future yield more efficient and robust algorithms than have been proposed.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank the Center for Undergraduate Research in Mathematics at Brigham Young University, NSF grant DMS-1148695, and the Undergraduate Research Program at California State University, Fresno for helping fund this project; Mr. Aramays Orkusyan for his collaboration and assistance on this project; Dr. Steve Chung for his guidance on statistics and probability; and Dr. Doreen De Leon for her advice and guidance on the entirety of this project. Finally, the authors would like to thank the editors and reviewers for their constructive criticisms and feedback that helped improve the quality of this paper.

## REFERENCES

- [1] Gergana Buono. Octave networks toolbox first release, July 2014.
- [2] Fan Chung, Alexander Tsiatas, and Paul Horn. Distributing antidote using pagerank vectors. *Internet Mathematics*, pages 237–254, 2009.
- [3] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [4] UCLA: Statistical Consulting Group. Data analysis examples, May 2015.
- [5] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70, 2004.
- [6] Octave community. GNU Octave 3.8.1, 2014.
- [7] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. *CoRR*, abs/1303.3984, 2013.
- [8] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.
- [9] Emilia Vynnycky and Richard G White. *An Introduction to Infectious Disease Modelling*. Oxford University Press, Oxford, NY, 2010.
- [10] Chenxi Wang, J. C. Knight, and M. C. Elder. On computer viral infection and the effect of immunization. In *Proceedings of the 16th Annual Computer Security Applications Conference, ACSAC '00*, pages 246–256, Washington, DC, USA, 2000. IEEE Computer Society.
- [11] Bernard L Welch. The generalization of student’s’ problem when several different population variances are involved. *Biometrika*, pages 28–35, 1947.

## APPENDIX A. SMALL NETWORKS: MATRICES

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 0.16 & 0.47 & 0.03 & 0.28 & 0 & 0 & 0.74 & 0 & 0 & 0.48 & 0.28 \\
 0 & 0 & 0.20 & 0 & 0.56 & 0.90 & 0.71 & 0 & 0 & 0.86 & 0.76 & 0 & 0.14 & 0 & 0.55 \\
 0 & 0.20 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.45 & 0 & 0.70 & 0 & 0 & 0 & 0.86 \\
 0 & 0 & 0 & 0 & 0.58 & 0.85 & 0 & 0.54 & 0.63 & 0.57 & 0 & 0 & 0 & 0.23 & 0.75 \\
 0.16 & 0.56 & 0 & 0.58 & 0 & 0 & 0 & 0 & 0 & 0.93 & 0 & 0.30 & 0 & 0.01 & 0.55 \\
 0.47 & 0.90 & 0 & 0.85 & 0 & 0 & 0 & 0.35 & 0.52 & 0 & 0.95 & 0 & 0.15 & 0 & 0 \\
 0.03 & 0.71 & 0.25 & 0 & 0 & 0 & 0 & 0.17 & 0.16 & 0.53 & 0 & 0 & 0.93 & 0 & 0.93 \\
 0.28 & 0 & 0 & 0.54 & 0 & 0.35 & 0.17 & 0 & 0 & 0.48 & 0.70 & 0.62 & 0.20 & 0 & 0.97 \\
 0 & 0 & 0.45 & 0.63 & 0 & 0.52 & 0.16 & 0 & 0 & 0.78 & 0 & 0.80 & 0.11 & 0 & 0.64 \\
 0 & 0.86 & 0 & 0.57 & 0.93 & 0 & 0.53 & 0.48 & 0.78 & 0 & 0 & 0 & 0 & 0.23 & 0 \\
 0.74 & 0.76 & 0.70 & 0 & 0 & 0.95 & 0 & 0.70 & 0 & 0 & 0 & 0.97 & 0.26 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0.30 & 0 & 0 & 0.62 & 0.80 & 0 & 0.97 & 0 & 0 & 0 & 0.46 \\
 0 & 0.14 & 0 & 0 & 0 & 0.15 & 0.93 & 0.20 & 0.11 & 0 & 0.25 & 0 & 0 & 0 & 0 \\
 0.48 & 0 & 0 & 0.23 & 0.01 & 0 & 0 & 0 & 0 & 0.23 & 0 & 0 & 0 & 0 & 0 \\
 0.28 & 0.55 & 0.86 & 0.75 & 0.55 & 0 & 0.93 & 0.97 & 0.64 & 0 & 0 & 0.46 & 0 & 0 & 0
 \end{bmatrix}
 \tag{1}$$

Experimentation Matrix