A Study of the Creation of Safe Zones Using Link Removal in Epidemiology

Ashna Dagam[a], Sanjay Dagam[a], Laura P. Schaposnik[b], Gabriela Ravenna[c]

(Pre-print publication - still undergoing revisions)

Keywords:

I. INTRODUCTION

While Hollywood may often exaggerate the differences and separation between different social cliques in a high school, distinct and indistinct friend groups are common throughout schools. Much like nodes connected by edges, high school cliques form distinct clusters within the larger social graph, each demonstrating unique patterns of interaction and influence. Additionally, these such social cliques consist of varying numbers of members, each who have a varying number of connections with other groups, similar to cliques in a graph. Each unique arrangement causes an infection such as gossip to spread based on its location of origin and the structure of the graph.

So how hard is it to isolate an individual or group of individuals from the entire social network? What is the maximum size group of individuals that we can prevent from receiving this gossip in a specific period of time? By constructing a graph and analyzing an efficient way to construct safe zones through link removal, we are able to better understand and answer the questions above.

This paper is dedicated to the introduction and study of safe zones within separable graphs. The aim is to maximize the number of nodes within these safe zones while minimizing the number of links removed. Beginning with simple graphs and gradually extending the findings and observations to larger and more complex graph configurations, we are able to formulate a systematic approach to establish safe zones.

By drawing upon the link removal strategies defined by Bellingeri et al.'s "A Comparative Analysis of Link Removal Networks in Real Complex Weighted Networks" [1], our study analyzes these strategies and determines which strategies are most effective under which circumstances. In the real-world, the safe zones formed by link removal might represent groups of individuals shielded from infections, with the nodes and the links themselves each representing an individual person and relationship respectively.

A. The Four Optimal Link Removal Strategies

From the six link removal strategies described in Section III C, we found that there are four types of link removal strategies that work best. These include weak link removal, strong link removal, BC link removal, and BCw link removal. Following are brief summaries of real life examples where each link removal strategy works best.

Weak Link Removal. The first strategy is weak link removal. It works best in graphs that contain many cliques with few, weak connections to the other cliques. For example, in a community such as a bigger town, there are strong-knitted families. However, there are many weak links between the families and other families. Therefore, when an infection spreads, if one family member gets infected, it is assumed the entire family will get infected. Hence, we want to remove the weak links between the families and other families.

Strong Link Removal. The second strategy is strong link removal. This works best in graphs that have many cliques with highly weighted edges connecting the many cliques. For instance, a popular kid in high school may have closer, personal relationships with other popular kids, but less personal relationships with new classmates. Connections between popular kids have strong weights, and must be prioritized as most of the information will run through such students. Therefore, strong link removal is the best link removal strategy.

BC Link Removal. The third strategy is betweenness centrality (BC) link removal. This works best when one node is connected to a majority of other nodes or one node connects several separate cliques. In a scenario where a person is the "bridge" between two groups of people (such as a popular teen at school), much of the interactions between the groups of people will run through the person. Thus, regardless of the weights of any of the links, the most optimal link removal strategy will be removing the links connecting the person who is the "bridge" and the groups. These links will have the highest BC of all links in the network.

BCw Link Removal. The last strategy is weighted betweenness centrality (BCw) link removal. BCw seemingly combines BC link removal with strong link removal. An example of when it is best would be in a school setting, where teachers have more personal interactions other teachers that teach other classes, but students primarily only interact with the students within their class. Therefore, the teachers are considered "connecting" vertices, yet it is noted that the students have more frequent interactions with each other than the teachers have with other teachers. Hence, BCw link removal is best because it will isolate the infection in a clique. (Note: strong link removal is not best as the students have stronger weighted connections with their classmates.)

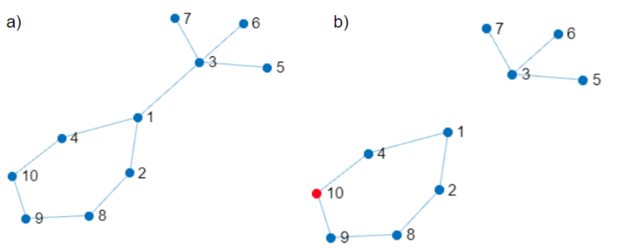

FIG. 1: Panel A depicts a simple graph with 10 nodes. Panel B shows two disconnected graphs with one red vertex denoting the infected vertex.

B. Background

As safe zones are simply parts of graphs, it is important to understand some basic graph theory terms. Specifically, nodes and edges are the fundamental units of graphs. In the context of epidemiology, nodes will most often represent individuals, either as infected, susceptible, or safe. Those who are susceptible or safe are not infected; susceptible persons are those who are able to be infected while safe persons are located within a safe zone and are thus unable to become infected. Additionally, an edge will often represent a connection whose specific details depend on the type of infection.

An infection can be used to describe anything that spreads such as a disease, gossip, etc. as discussed in Definition 3. It will also be assumed that the infection can spread to any node adjacent to an infected node. The time it takes an infection to spread to a neighboring vertex is denoted by T.

Furthermore, the term "link weight" (as defined in Definition 5) will be used to demonstrate how strong a connection is (ie. how fast an infection travels through the edge). Generally, an edge with a link weight of n indicates that the infection travels faster through that edge than through another edge of link weight $n-1$ and slower than through an edge with link weight $n+1$. For example, given three different edges of link weight 4, 10, and 16, an infection will travel the fastest through the edge with link weight 16 and slowest through the edge with link weight 4. Note that the term "link weight" generally refers to a number that is assigned to a link which is then used to represent relative costs, distances, etc. However, "link weight" will only be used to indicate the strength of the relationship between the two nodes the edge connects.

C. Definitions

In order to fully understand each link removal strategy explained in Section III C, it is imperative to be familiar with some common terminology.

Definition 1 explains a "graph" as defined in Chang and Nichols' *Introduction to Graph Theory* [2]. This is essential as safe zones are simply specific sections of graphs.

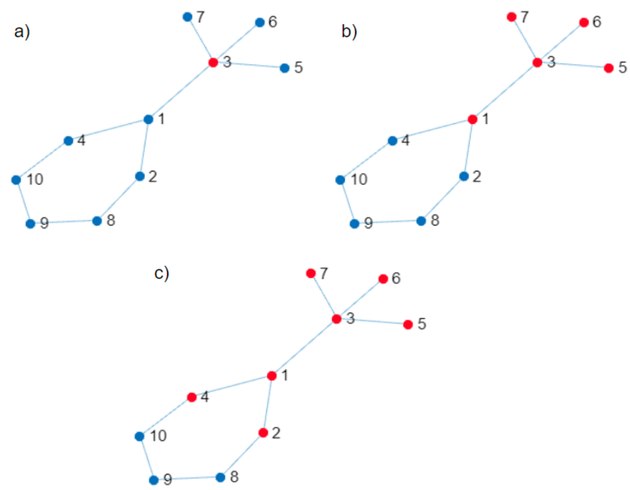

FIG. 2: Panels A, B, and C show the same graph, G_1, at times $T=1$, $T=2$, and $T=3$ respectively. The red vertices denote infected vertices while the blue vertices are vertices susceptible to infection. As T increases, the infection spreads and more vertices become infected.

Definition 1 (Graph)**.** A graph $G=(V,E)$ is a mathematical structure consisting of two sets V and E. The elements of V are the vertices of G, and the elements of E the edges of G. For our purposes, each edge will have a set of two vertices associated to it, which are called its endpoints.

Remark 1. *It can be seen that all the nodes and edges in Panel A of Figure 1 form a graph consisting of 10 nodes and 10 edges.*

In Definition 2, the term "safe zone" is defined. Since this paper centers around the topic of safe zones, it is necessary to first define exactly what a safe zone is.

Definition 2 (Safe Zone)**.** The safe zone of a graph is defined as a subset of vertices that cannot be infected, and makes up one connected component.

Remark 2. *For example, in Panel B of Figure 1 if the red vertex, Vertex 10, is defined to be infected, then Vertices 3, 5, 6, and 7 form a safe zone since they cannot be infected.*

An "infection" is defined in Definition 3. As safe zones are areas of a graph that cannot be infected, it is necessary to explain exactly what an infection is.

Definition 3 (Infection)**.** An infection in a graph is a timed process where at each time, the infection will be transferred to an available vertex (a connected vertex that is able to be infected, e.g. not a safe zone) which will then carry the infection as well.

Remark 3. *An infection can simply be a plague/disease as well as the spread of gossip, information, etc. The progression of an infection from $T=1$ to $T=3$ can be seen in Figure 2. Panels A, B, and C depict the graph G_1 at each point in time. Note that the infection spreads over time which is shown by the increase of infected (red) vertices.*

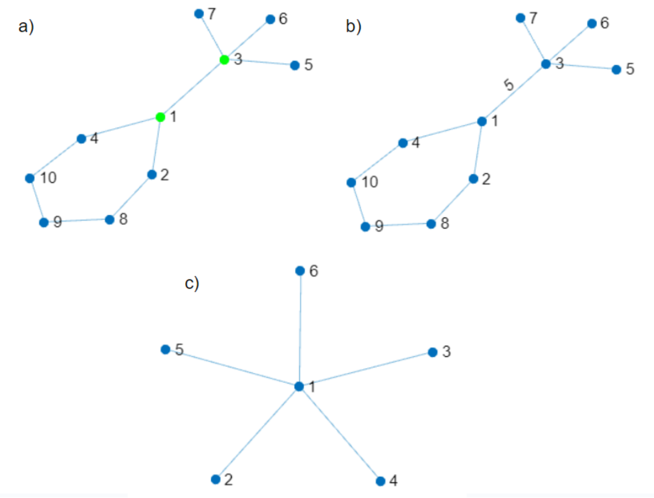

FIG. 3: Panel A shows a graph with 10 nodes where 2 nodes, Vertices 1 and 3, are colored green. Panel B depicts a graph also of 10 nodes where the edge connecting Vertex 1 and Vertex 3 is given a link weight of 5. All other edges are unweighted. Panel C shows a 6-node-graph where Vertices 2 through 6 are connected to Vertex 1.

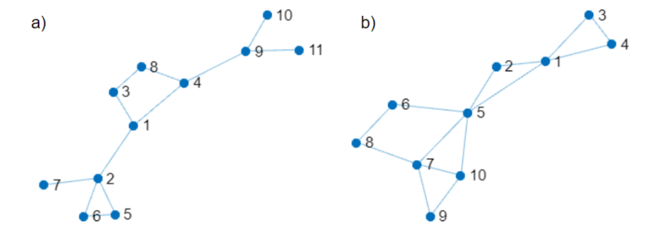

FIG. 4: Panel A and Panel B both show a graph comprised of 10 nodes. The graph in Panel A is defined to be a one-connected-graph while the graph in Panel B is defined to be a two-connected-graph.

As it is essential to classify types of nodes within a graph, Definition 4 defines a "network".

Definition 4 (Network)**.** A network is represented by a graph G whose vertices have different labels corresponding to the type of role the person/thing plays in the network.

Remark 4. *In Panel A of Figure 3, it could be defined that the blue vertices represent the employees of a company while the green vertices represent the company's managers.*

Definition 5 defines "link weight". This is essential to understanding how strong, weak, and weighted betweenness centrality link removal strategies are classified (as in Section III C).

Definition 5 (Link Weight)**.** The weight of a link is merely a number assigned to the link (edge). A higher weight indicates a stronger link, which means that the infection travels faster through the link from one connecting node to the next.

Remark 5. *For instance, in Panel B Figure 3, the edge connecting vertices 1 and 3 has link weight 5 while the other edges are not assigned weights.*

Definition 6 explores Betweenness Centrality as defined in Motoki Watabe's exercise on Betweenness Centrality [3]. This term is essential as it is a key term in two link removal strategies (as seen in Section III C).

Definition 6 (Betweenness Centrality of a node or of a link)**.** The betweenness centrality of a link or node models a way of calculating the amount of influence a single link or node has in a graph in regards to the flow of information. It is calculated by first determining the link or node being examined. Then, we take every pair of nodes in a graph, and determine how many of the pairs of nodes have their shortest path running through the link or node we are examining. Therefore, the higher the betweenness centrality, the more influence the node or link has on the graph.

Remark 6. *For example, Vertex 1 in Panel C of Figure 3 has the highest BC because the most paths connecting two vertices go through it. This can be noticed simply by observation. Thus, this node has the most influence on the graph.*

Definition 7 defines the term "K-connected-graph" as seen in David Guichard's Combinatorics and Graph Theory [4]. As the term is used to define the general types of graphs seen in Section II A and Section II C, it is imperative to explain it first.

Definition 7 (K-Edge-Connected-Graphs)**.** If a graph G is connected, any set of edges whose removal disconnects the graph is called a **cut**. G has edge connectivity k if there is a cut of size k but no smaller cut; the edge connectivity of a one-vertex graph is undefined. G is k-edge-connected if the edge connectivity of G is at least k. The edge connectivity is denoted λ(G).

Remark 7. *For instance, Panels A and B in Figure 4 depict two different k-edge-connected graphs; Panel A shows a one-edge-connected-graph, and Panel B shows a two-edge-connected-graph. This is because the graph in Panel A can be disconnected by exactly 1 cut (such as cutting the edge between Vertices 1 and 2). However, at least 2 cuts are necessary to disconnect the graph in Panel B (such as cutting the edge between Vertices 1 and 2 and the edge between Vertices 1 and 5).*

Definition 8 (Graph Density)**.** The density of a graph is defined as the ratio of the number of edges E in a graph to the maximum number of total edges that could be in the graph. It can be concluded that for a graph G with n nodes where there is only a maximum of one edge between two given vertices, the maximum number of edges in G is $\binom{n}{2}$.

Definition 9 (Weighted Network)**.** A weighted network is a network where each edge contained in the graph has a weight assigned to it.

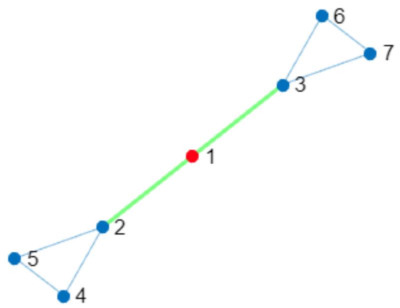

FIG. 5: A simple graph with 2 cliques of 3 vertices each and 1 connecting vertex.

Definition 10 (Minimum Cut)**.** In an unweighted network, the minimum cut is the minimum number of edges removed such that the graph is disconnected. In a weighted network, the minimum cut is the minimum sum of the weights of edges that, when removed, disconnect the graph.

Remark 8. *Definition 10 defines the term "minimum cut" in both an unweighted and weighted network, as seen in*

II. EFFICIENT LINK REMOVAL

Before forming general strategies of efficient link removal for any such graph, it is necessary to first look at some examples of different types of graphs with limited numbers of nodes. Example 1 is a one-edge-connected graph while Example 2 is a two-edge-connected graph.

A. One-Edge-Connected Graph

A one-edge-connected graph is one which can be disconnected by exactly one cut as seen in Figure 5. This can be seen in Definition 7 which defines a one-edge-connected graph as a graph for which there exists a cut of exactly 1 (but no smaller cut) which disconnects the graph. In Figure 5, the two cliques of the graph are clear (the leftmost 3 vertices and the rightmost 3 vertices). The red-colored vertex (Vertex 1) is the connecting vertex of the two cliques, and the two green edges are the connecting edges between the two cliques and Vertex 1. The green edges are the most important when there is a disease because if either of those such edges are removed, the two cliques will disconnected and will form a safe zone. Thus, the best strategy can be found through casework depending on where the infection originates.

Case 1: the infection starts at Vertex 1. In order to create a safe zone, the connection between Vertex 1 and the leftmost clique or the rightmost clique must be severed. To do this, at least one of the green edges must be removed. This would create a clique that is a safe zone.

Case 2: the infection starts at Vertex 2 or 3. Since the graph in Figure 5 is symmetrical, Vertex 2 and Vertex 3 can be considered identically. Without loss of generality, assume the infection originates at Vertex 3. The rightmost clique is susceptible to infection due to two available infection pathways. Securing the leftmost clique is an easier task since it is non-adjacent to the infected vertex. Consequently, the removal of the connection between Vertex 1 and 3 or Vertex 1 and 2 establishes a safe zone. Optimal safety ensues from severing the edge between Vertex 1 and 3, isolating Vertex 1 from infection. However, if the infection spreads too fast, severing the edge between Vertex 1 and 2 will give more time. In a mirrored scenario where the infection begins at Vertex 2, the same rationale applies, except in reverse, due to the symmetry depicted in Figure 5.

Case 3: the infection begins at Vertex 4, 5, 6, or 7. Again, because the graph shown in Figure 5 is symmetrical, Vertices 4, 5, 6, and 7 can be considered identically. Without loss of generality, assume the infection starts on either Vertex 6 or 7. The rightmost clique will most likely succumb to the infection, as there are two paths for the infection to spread. Thus, it will be much easier to secure the leftmost clique. By severing the edge between Vertex 1 and 3, the most logical safe zone will be created. Similarly, if the leftmost clique was where the infection started, the same logic applies, but only reversed, as Figure 5 is symmetrical. This leads us to a conclusion:

Lemma 1. *If a graph G is symmetrical, then two cliques, C_1 and C_2, in G of an equal number of nodes and where each node in C_1 has exactly one corresponding node in C_2 of equal degree, then each of the two cliques has an equal chance of becoming a safe zone.*

Proof. If G is symmetrical, then C_1 and C_2 may be considered identical due to their equivalent properties. Therefore, they will not be considered differently in any way in the process of selecting which edges will be cut. Thus, C_1 and C_2 will have equal chances of becoming a safe zone. □

Conclusion. It can be noticed that Cases 2 and 3 have extremely similar safe zone creation strategies. This is because of how the graph is structured: it includes two symmetrical cliques. Therefore, if the infection originates in one of the cliques, the best course of action is to isolate the infection in the clique in which it originated. This similarity leads to an important conclusion:

Lemma 2. *In a one-edge-connected graph where the minimum cut is a cut in a central location of the graph, if the location of the infection is unknown or undetermined, removing the minimum cut is the best or nearly the best way to create a safe zone.*

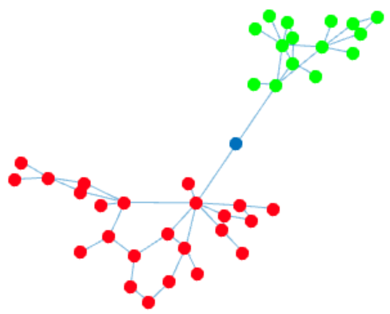

FIG. 6: A computer generated one-edge-connected graph

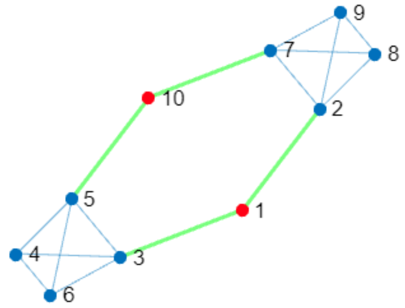

FIG. 7: A simple graph with 2 cliques of 4 vertices each and 2 connecting vertices.

Proof. We know a minimum cut is contained in the center of the graph. Since a minimum cut is contained in the center of the graph, cutting the graph along its minimum cut results in a disconnected graph with the closest number of vertices in each portion of the disconnected graph as can be. Since we do not know the location of the infection, it is best to to disconnected the graph into equal or nearly equal portions of vertices.

The above statement can be found from an expected value problem, where we let x and y be the number of vertices in both disconnected graphs. The expected number of nodes in the safe zone (the disconnected graph without the infection) is $x * \frac{y}{x+y} + y * \frac{x}{x+y} = \frac{2xy}{x+y}$. (if a safe zone with x vertices is created, the infection must start in the section with y vertices) From the AM-GM Inequality, we have that $2\sqrt{xy} \leq x + y$, with equality when $x = y$. Therefore, it follows that to maximize $\frac{2xy}{x+y}$, we must have $x = y$. Additionally, we see that getting x and y as close together as possible without changing $x + y$ increases $\frac{2xy}{x+y}$. □

B. A General One-Edge-Connected Graph.

Figure 6 is a computer generated one-edge-connected graph. There are two sections of the graph which we will call G_1 and G_2. G_1 will denote the subsection of the graph composed of all the red vertices while G_2 will denote the subsection of the graph composed of all the green vertices.

A one-edge-connected graph can be disconnected with 1 link removal (the minimum cut). We have been considering removing this link (the minimum cut) in the specific example of a one-edge-connected graph. However, this does not necessarily optimize the link removal because it may create a safe zone that does not contain many vertices. It will create a safe zone, regardless.

Furthermore, our aim is to examine the connections characterized by the highest betweenness centrality, as described in Definition 6. These connections play a critical role in potentially fragmenting the graph which when disconnected, creates a safe zone containing a large number of the total nodes. Moreover, through brainstorming and logical analysis, it appears that links possessing the highest betweenness centrality are likely to be part of the minimum cut or a cut that does not substantially exceed the minimum cut.

We also note that a one-edge-connected graph can have any number of cliques greater than or equal to 2. The computer generated graph has 2 cliques. However, no matter the number of cliques, we want to examine the edge part of the minimum cut (the minimum cut is 1) as well as the edges with high betweenness centrality.

Theorem 1. *Given a one-edge-connected graph, the optimal cut is one which disconnects the graph and removes the links with high betweenness centrality.*

Proof. Since our aim is to create the largest possible safe zone, our optimal cut must create a safe zone and disconnect the graph.

From Lemma 2, if the minimum cut is in a central location of the graph, the optimal link removal strategy is the removal of that link. We can see that this link has high betweenness centrality due to its location and will thus create at least one safe zone containing a large number of the total nodes. Therefore, removing this link is optimal.

Furthermore, if the minimum cut is not in a central location, then the removal of this link will not result in a safe zone consisting of a high percentage of the total number of vertices. Thus, removing this link is not optimal.

Therefore, we must consider BC in the optimal cut. If we make a cut that removes links with high BC and disconnects the graph, we will form a safe zone consisting of a high percentage of the total number of nodes. Thus, even if the number of links removed exceeds the minimum cut, a more optimal safe zone will be formed. Note that how many links we can remove depends on the time restrictions of the given graph. Thus if graphs G_1 and G_2 are identical graphs but the infection in G_1 spreads to more nodes in time T than the infection in G_2 in time T, the optimal cut may differ between the two graphs. □

C. Two-Edge-Connected Graph

In example 2, we will study Figure 7. Figure 7 is a two-edge-connected graph. As defined in Definition 7, a two-edge-connected-graph is one which can be disconnected by a cut of size 2 (and no smaller). Figure 7 has two cliques with 4 vertices, and there are 2 connecting vertices.

1. Case 1 - the infection starts at either of the red vertices

If the infection starts at either of the two red vertices (Vertex 1 or 10), everything is susceptible to the infection. To start, in order to create a safe zone, either the leftmost edge that is adjacent to the infected red vertex, or the rightmost edge that is adjacent to the infected red vertex must be removed. It is important that this is shut down. Otherwise, the infection will spread to both cliques. Therefore, we must assume that one of the green edges adjacent to the infected red vertex is removed. Although this may not be the case as the edge is adjacent to the infected vertex, we must assume that this edge gets removed before time T, when the infection spreads. Using the symmetry of the graph, without loss of generality, let's assume we removed the leftmost edge that is adjacent to the infected red vertex. Note that if we removed the rightmost edge, our logic would follow, just reversed. Finally, call the infected vertex Vertex A. After this, we have two methods for creating a safe zone:

Method 1: remove a green edge not adjacent to Vertex A. After we removed the leftmost green edge adjacent to Vertex A, we still need to remove another edge. One way is to remove a green edge not adjacent to Vertex A. There are two options - either the rightmost green edge not adjacent to Vertex A, or the leftmost green edge not adjacent to Vertex A. Opting to remove the leftmost green edge, which is not connected to Vertex A, would be the simpler choice since it is further from the source of infection. However, opting to eliminate the rightmost green edge, which is not adjacent to Vertex A, would result in saving more individuals (specifically, the red vertex that is not Vertex A). This option is more challenging, as the rightmost green edge not adjacent to Vertex A is closer to the infection compared to the leftmost green edge.

Method 2: remove a green edge adjacent to Vertex A. This way is clearly optimal. This would isolate the infection. However, unless there is absolute certainty that the infection will not spread outside of Vertex A, this method could be disastrous. This is utterly important as at every time T, the infection spreads. Nevertheless, this method is optimal - the whole population besides Vertex A would be safe from the infection.

Conclusion of Case 1. In both methods, we removed a green edge. This is because it was part of the minimum cut of the graph, and since we initially assumed we removed an edge adjacent to Vertex A, we just needed to remove one more green edge.

Lemma 3. *In a graph where a vertex connected to at least one edge on at least two sides is infected, the edges are part of a minimum cut, and the minimum cut is near the middle of the graph, removing the minimum cut or removing edges from the minimum cut so in the next time T the infection cannot spread into the area of the graph which we want to create a safe zone is the best strategy to create a safe zone.*

Proof. In Figure 7, when the red vertices were infected, the best strategy was to immediately remove one of the green edges connected to the infection vertex. This edge was part of the minimum cut, and it prevented the infection from spreading into both cliques. After doing so, we were left with a one-edge-connected graph, and we could remove one edge to create a safe zone with nearly half of the total graph's vertices. □

2. Case 2 - the infection starts in a vertex adjacent to a red vertex

Without loss of generality, assume the infection started in Vertex 3 (we can do this because of the left-right symmetry and the up-down symmetry of the graph). Now, for this case, unlike previous cases, we will work out the problem based on assumptions on how many edges we can remove in time T, before the graph spreads.

Assume we can remove 4 or more edges every T. However unlikely this is (since in Section II A, we had said it is hard to remove just one edge in time T), we must consider this case. If this were to be true, we simply remove the four adjacent edges of Vertex 3. The rest of the network is safe.

Assume we can remove 3 edges every T. If we are removing 3 edges, one of the adjacent vertices of Vertex 3 will get infected. We want this adjacent vertex which will get infected to have the least degree so the infection can then be isolated in the next time T. The only adjacent vertex of Vertex 3 that has degree greater than 3 is Vertex 5. Thus, we must remove the edge connecting Vertex 3 to Vertex 5. Other than that, we can remove any two of the other 3 adjacent edges of Vertex 3. After the time T, the infection will spread to one of the adjacent vertices. From there, since every vertex that the infection could spread to has degree 3 or less, we can simply remove the adjacent edges of this newly infected vertex. Now, the entire graph is safe except for Vertex 3 and one other vertex.

Assume we can remove 2 edges every T. This is case requires a little bit of guess and check. It will not be as straightforward as before. However, after a few tries, we can find the best method: remove the green edge adjacent to Vertex 3 and the edge connecting Vertex 5 to Vertex 3. Now after time T, the infection will spread to Vertex 4 and 6. Then we can remove the two edges that are still connected to Vertex 5. This will keep the infection contained in three vertices - Vertex 3, 4, and 6.

Assume we can remove 1 edge every T. This case is very straightforward. We first remove the green edge adjacent to Vertex 3, the infected vertex. After time T, the infection will spread to all vertices in the leftmost clique. Next, remove the other green edge adjacent the infected clique. This will effectively keep the infection contained within a clique. The rest of the graph will be safe.

Conclusion for Case 2. Our goals with each guideline (remove 2 edges every T, 1 edge every T, etc.) were to remove the green edges to save the rightmost clique. Then, we would set that as our base, and try different ways to save the more people.

Additionally, our method of assuming a set number of edges that we could remove every T is the best method for larger cases. As we progress, we will implement this new method to keep organized.

Lemma 4. *In a graph where the minimum cut is near the center of the graph, if the infected vertex is a vertex that is connected to at least one edge, which is part of the minimum cut on only one side, then removing those edge(s) is the best way to create a safe zone, or start to create a safe zone.*

Proof. In Figure 7, when a vertex adjacent to the red vertices was infection, the best strategy was to immediately remove the green edge connected to the infected vertex. This edge was part of the minimum cut, and it prevented the infection from spreading to the middle part of the graph (the part containing the minimum cuts and the red vertices). After doing so, we were left with a one-edge-connected graph, with which one edge could easily be removed to create a safe zone.

Additionally, if we did not immediately remove the green edge connected to the infected vertex, the infection would have spread to the middle part of the graph, threatening a potential spread to the clique in which the infection did not originate. □

3. *Case 3 - the infection starts on a vertex which is neither red nor adjacent to a red vertex*

Without loss of generality, let the infection start in Vertex 6. We will proceed how we did in the above subsubsection. Note that our logic for creating a safe zone can be applied to the other 3 vertices in Figure 7 simply by reversing the logic depending on whether the infection starts at Vertex 4, 8, or 9. This is due to the graph's symmetry.

Assume we can remove 3 or more edges every T. We simply remove all 3 edges adjacent to Vertex 6. The entire rest of the graph is safe.

Assume we can remove 2 edges every T. There are 3 edges adjacent to Vertex 6. If we can only remove 2 edges every T, one of the adjacent edges cannot be removed, so an adjacent vertex will get infected. Now, let's look at the adjacent vertices of Vertex 6. Two of the three adjacent vertices have 4 adjacent vertices. However, there is one which only has 3 adjacent vertices - Vertex 4. Now, we can come up with a strategy. First, we remove the two adjacent edges of Vertex 6 which do not connect to Vertex 4. After doing so, the infection will spread only to Vertex 4. Next, we remove the two adjacent edges of Vertex 4 which do not connect back to Vertex 6. Moreover, the infection cannot spread anywhere else, and is simply contained in Vertex 4 and Vertex 6.

Assume we can remove 1 edge every T. This case will require some guess and check, but it is very simple. Simply we remove any one of the 2 green edges connecting the leftmost clique to the rest of the graph. After this, the infection will spread to all of the leftmost clique. Next, we simply remove the second green edge connecting the leftmost clique to the rest of the graph. This will isolate the infection in the leftmost clique, saving the rest of the graph.

Other cases of Case 3 and Conclusion. There are other variations to consider (one can remove 1 edge, then 0 edges, then 2 edges, etc.) However, we still have a strategy: Find the degree of the initially infected vertex. Use logic and sometimes a little guess and check to figure out which edges to remove to save the most amount of people (ie. save the most amount of vertices).

Lemma 5. *In a graph where the minimum cut is near the center of the graph and the infected vertex is not connected to any edges part of the minimum cut, if the infection cannot be isolated within a couple of vertices and can only be isolated within the clique the initially infected vertex is located in, then removing the minimum cut is the best way to create a safe zone.*

Proof. In Figure 7, when a vertex which was not a red vertex nor adjacent to a red vertex was infection, the best strategy was to either isolate the infection within a couple of vertices, or if the infection could only be isolated within the clique the initially infected vertex was located in, then removing the minimum cut was the strategy to create a safe zone. No matter which minimum cut was used, a safe zone was created with more than half of the vertices, half of the vertices, or nearly half of the vertices. □

Proposition 1. *Given a two-edge-connected graph with the minimum cut near the center of the graph, if the infection cannot be isolated within a few vertices, removing the minimum cut is the optimal way to create a safe zone.*

Proof. In all cases in Section II C, no matter where the infection started, the first strategy of creating a safe zone

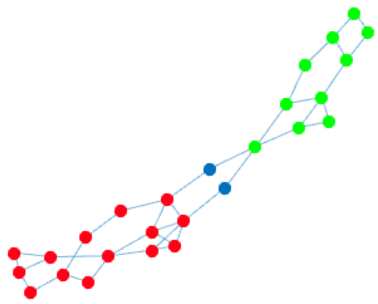

FIG. 8: A computer generated two-edge-connected graph

was to remove the minimum cut. If the infection could easily be isolated within a few vertices before spreading to the middle portion of the graph, we would like to isolate the infection. However, if a quick isolation was not possible, the optimal link removal strategy had us removing the minimum cut as Figure 7 had its minimum cuts in the center of the graph, or near the center of the graph. □

D. A General Two-Edge-Connected Graph

The graph shown in Figure 8 is a computer-generated two-edge-connected graph, comprising two primary sections referred to as G_1 and G_2. Specifically, G_1 encompasses all the red vertices, while G_2 denotes the green vertices within the graph.

Similarly to one-edge-connected graph, the minimum cut will consist of 2 edges being removed. In our specific example of a two-edge-connected graph, we have been considering this minimum cut. However, we cannot assume this is the most optimal link removal for all graphs. It will create a safe zone, nevertheless.

Additionally, similarly to the one-edge-connected graph, removing links with high betweenness centrality will fragment the graph into disconnected parts with high amounts of vertices. From logic and testing, we hypothesise that edges with high betweenness centrality will be part of the minimum cut or a cut that does not exceed the minimum cut considerably.

We also note that a two-edge-connected graph can have any number of cliques greater than or equal to 2. The computer generated graph has 2 cliques. However, no matter the number of cliques, we want to examine the minimum cut as well as the edges with the highest betweenness centrality.

Theorem 2. *Given a two-edge-connected graph, the optimal cut is one which disconnects the graph and removes the links with high betweenness centrality.*

Proof. From Proposition 1, if the minimum cut is in a central location of the graph, the optimal link removal strategy is the removal of those links. We can see that those links have high betweenness centrality due to their location, and therefore, removing them will create a safe zone containing a large number of the total nodes. Therefore, removing those links is optimal.

Furthermore, if the minimum cut is not in a central location, then the removal of the links will not result in a safe zone consisting of a high percentage of the total number of vertices. Thus, removing the links is not optimal.

Therefore, we must consider BC in the optimal cut. If we make a cut that removes links with high BC and disconnects the graph, we will form a safe zone consisting of a high percentage of the total number of nodes. Thus, even if the number of links removed exceeds the minimum cut, a more optimal safe zone will be formed. (The minimum cut is not in a central part of the graph, but a cut of edges with high BC is). □

Remark 9. *The similarity of Theorem 1 and Theorem 2 and their proofs shows that a one-edge-connected-graph builds off a two-edge-connected graph.*

Comparison of a general one-edge and general two-edge connected graphs. In both cases, we find that examining the minimum cut of the graph is key, as by definition, this is the cut that will create a safe zone while removing the least number of vertices. Additionally, for both cases, we know that removing links with high betweenness centrality will fragment the graph into disconnected parts, each with a high amount of the total number of vertices. Additionally, we hypothesise that in most graphs, the edges with high betweenness centrality will either be part of the minimum cut, or they will be part of a cut that does not include substantially more vertices than the minimum cut.

However, the only difference is simply that the minimum cut of a one-edge connected graph is 1, while the minimum cut of a two-edge connected graph is 2. Additionally, in general a two-edge connected graph will contain more vertices and more links than a one-edge connected graph, but that is not necessarily the case in all example. Therefore, a two-edge connected graph will be more complicated than a one-edge connected graph. Nevertheless, the strategies of examining the minimum cut as well as edges with high betweenness centrality still apply to both one-edge and two-edge connected graphs.

E. A General Three-Edge Connected Graph

Let's consider a general three-edge-connected graph such as the one in Figure 9.

We can start by examining the three-edge-connected graph. Since there are less obvious groups to divide the graph into based on the minimum cut, as there are many minimum cuts which result in subgraphs with an approximately equal number of vertices. Furthermore, the minimum cut may not be the most optimal way to create safe zones. This is shown in Figure 10 in Panels A, B, C, and D.

Additionally, our link removal strategies depend much more on the location of the initially infected vertex, and

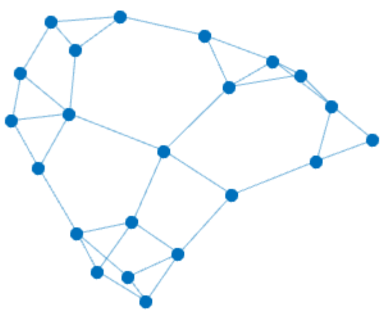

FIG. 9: A computer generated three-edge-connected graph

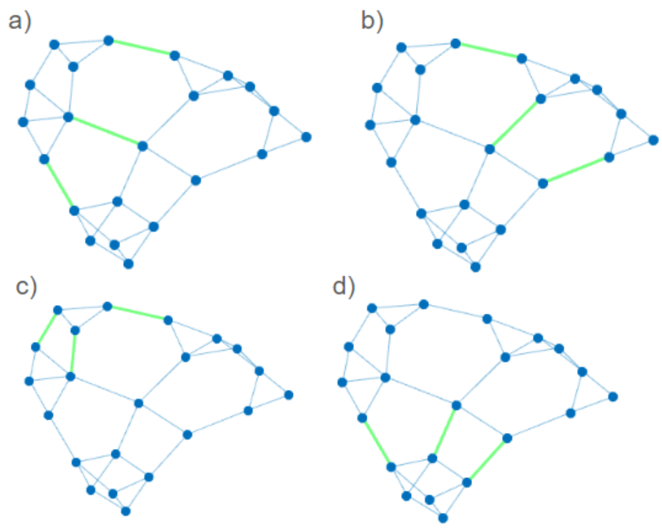

FIG. 10: Four examples of minimum cuts for the three-edge-connected graph shown in Figure 9. The edges that would be severed as a part of each minimum cut are shown in green

the time restrictions we have when cutting edges. This is again because of the numerous ways to create safe zone based on the minimum cut, and the similarity of the amount of vertices saved based on each way to create a safe zone.

In general, three-edge-connected graphs are more complex than one and two-edge-connected graphs. This means that in more cases, such as the computer generated graph, removing a minimum cut will not be optimal, or it would depend on the location of the initial infection to determine which minimum cut to remove.

Lemma 6. *Given a three-edge-connected graph, the link removal strategy generally depends more heavily on the initial position of the infection.*

Proof. Figure 10 shows different ways to create a safe zone based on removing the minimum cut. Each panel shows a different set of links removed. However, in Panels A, B, and D, the graph is disconnected into two sections that have somewhat similar amount of vertices, but not the same or nearly the same amount of vertices. Furthermore, in the three mentioned panels, the disconnected portions of the graph contain similar amounts of vertices. This means there is not a clear optimal link removal strategy for any starting infection, and therefore the optimal link removal strategy depends more heavily on the initial position of the infection. □

F. K-edge-connected graph lemmas

Lemma 7. *Given a k-edge-connected graph and one edge is removed, a k-1-edge-connected graph is created if and only if the edge removed is part of all the minimum cuts.*

Proof. The minimum cut of any k-edge-connected graph will contain k edges by definition. If we remove one edge such that it is part of all the minimum cuts minimum cut, we are left with all minimum cut of k-1-edges. Therefore, since the new graph has a collection of k-1 edges that disconnect the graph, it is by definition a k-1-edge-connected graph.

Additionally, there is no other way to remove just one edge from a k-edge-connected graph and result in a k-1-edge-connected graph than mentioned in the lemma. This is the second part needed to be justified for the if and only if to hold in the lemma. □

Lemma 8. *Given a k-1-edge-connected graph and one edge is added, a k-edge-connected graph is created if and only if the edge added adds one edge to all minimum cuts*

Proof. The minimum cut of any k-1-edge-connected graph will contain k-1 edges by definition. If we add one edge such that it adds one edge to all minimum cuts, we are left with all minimum cuts of k-edges. Therefore, since the new graph has a minimum collection of k edges that disconnect the graph, it is by definition a k-edge-connected graph.

Additionally, there is no other way to add just one edge from a k-1-edge-connected graph and result in a k-edge-connected graph than mentioned in the lemma. This is the second part needed to be justified for the if and only if to hold in the lemma. □

Remark 10. *The two lemmas above show that every k-edge-connected graph builds off a k-1-edge connected graph. The lemma proves that any graph can be examined through its minimum cut.*

Lemma 9. *The overall trend within k-edge-connected-graphs is the greater k is, the greater its graph density.*

Proof. Consider a graph G with l vertices. Since graph density (see Definition 8) is defined as the number of edges over the total number of possible edges, the total number of possible edges in G is $\binom{l}{2} = \frac{l(l-1)}{2}$.

Now we note that if G is a k-edge-connected graph then there is a way to disconnect k by removing exactly k edges. This will create two distinct subgraphs which we will call S_1 and S_2. Thus S_1 and S_2 are connected by exactly k edges.

If we now consider a graph G_2 where S_1 and S_2 are connected by exactly $k+1$ edges, then there will be one more edge in G_2 than G_1. Furthermore, each graph has the same number of nodes so the graph density of G_2 is greater than the graph density of G_1.

Then, if G_2 is a $k+1$-edge-connected graph, then it can be concluded that the trend for graph density increases as k increases. However, if G_2 is not a $k+1$-edge-connected graph, then there is a cut of exactly k which disconnects the graph and G_2 is still a k-edge-connected graph. Thus to transform G_2 to a $k+1$-edge-connected graph, additional edges must be added which will in turn, increase the graph density of G_2. Thus, there is a trend of increasing graph density of a k-edge-connected graph as k increases. Note that not all $k+1$-edge-connected graphs will have a greater graph density than all k-edge-connected graphs; however there is a one-to-one correspondence apparent leading to a trend of increasing graph density with an increasing k value. □

G. Optimal link Removal Strategy proposition

Proposition 2. *Consider the cut of x edges that disconnects the graph into 2 disjoint sections. Let the non-infected section of the graph consist of y vertices. The optimal link removal strategy minimizes $\frac{x}{y}$.*

Remark 11. *The proposition above considers not just maximizing the number of vertices saved (y) or minimizing the number of edges removed (x), but takes into account both by minimizing $\frac{x}{y}$.*

We note that this strategy is based of the sparsest cut problem [5], although we are considering the non-infected section of the graph. The sparsest cut problem considers the smaller section of the graph once it has been cut. The non-infected section is not necessarily the smaller section of the cut graph.

III. LINK REMOVAL

Extensive research has been conducted on various strategies for link removal within social networks. In epidemiology, the primary focus has centered on optimizing node removal through vaccination strategies. However, not every disease or infection has a vaccine or cure. This makes the creation of safe zones using link removal a crucial step in limiting the spread of a disease. Since our primary objective is to sever links, not nodes, our emphasis lies in restricting or ideally entirely stopping transportation between nodes. Therefore, it is crucial to compare the advancements and strategies of link removal in social networks to our conclusions from the above two figures to find similarities.

The biggest challenge we face in our adaptation is that in social networks, it is possible for information to strictly be able to transfer in one direction. For instance, given two arbitrary nodes A and B, it may be possible for information to be able to travel from A to B but not from B to A. However, in epidemiology, diseases can always transfer from either node A to node B or from node B to node A. Thus, similar strategies used in social networks may be applied to epidemiology, but the strategies must be adapted to fit the new restrictions.

A. History of Link Removal

This subsection utilizes Bellingeri et al.'s [6] summary of link and node removal in real, social networks.

One of the first and most famous studies of link removal was conducted by Mark Granovetter in 1973 and titled "The Strength of Weak Ties"[7]. His research originally focused on sociological theory and represented groups of people as graphs with each individual as a node and each relation or interaction between two individuals as edges but has been since applied more generally to other types of networks. Furthermore, he characterized each relation (edge) as "strong", "weak", or "absent", similar to how a weighted network (as defined in Definition 9) is constructed. Remember that a weighted network is one in which each edge is assigned a weight. For instance, a "strong" link may be between two close friends or family members while a "weak" link may be a relation between two acquaintances. Finally, an "absent" link may refer to a relationship that is non-existent or a relationship which is nothing more than knowledge of the other person's name. Next, Granovetter argued in his paper that weaker weighted links act as bridges between stronger weighted links in strong-knit communities, as stated in his "weak-link hypothesis." Additionally, Granovetter explained that weak links function as a type of glue, and they may be the most important links in the network, as the weak links connect the communities of strong weighted links together. Without the weak links, information would not transfer from community to community. Hence, Granovetter's main point was his theory that social networks consist of weak links which are more important than strong links.

B. Advances in Link Removal

This subsection utilizes Bellingeri et al.'s [6] summary of link and node removal in real, social networks.

After Granovetter's "The Strength of Weak Ties" [7], there were some more studies that aimed to prove the "weak-link hypothesis." These included Onnela et al.'s [8] research and Garas et al.'s [9] study. The former built a social network by collecting mobile phone call records and by describing the nodes as individuals and their phone calls as links, weighting the phone call links by their duration. The latter assigned a weighted link between two nodes representing different stocks according to the cross-correlation between the return time series of each stock in the New York Stock Exchange. Both studies confirmed the "weak-link hypothesis", since both found that removing weak links decreases the largest connected cluster (LCC) more than removing strong links.

Following these two studies, Pan and Sarmaki performed analysis on the co-authorship network in the field of physics[10]. The network was formed by scientists, which were represented as nodes, and links weighted by the number of co-authored papers. Unlike previous analysis in social networks, removing strong links shrunk the LCC faster. Although the data contradicts the "weak-link hypothesis," it makes sense given the type of data, as Pan and Sarmaki's analysis revealed that dense, local neighborhoods contain weaker links, yet there were stronger links between senior scientists leading different research groups.

Following Pan and Sarmaki's research, Pajevic and Plenz performed a comprehensive analysis of science co-authorship and cinema collaboration social network robustness[11]. In their analysis, all 4 science co-authorship showed the LCC was more vulnerable to strong link removal. However, in Pajevic and Plenz's analysis of cinema collaboration (where nodes represent actors, and weights of links represent the amount of movies in which they appeared), weak link removal fractured the LCC more than strong link removal. Therefore, a true conclusion could not be drawn because the strength of the ideal link removal depended on how the network was composed.

The last major and most recent study occurred with Bellingeri et al.'s "A Comparative Analysis of Link Removal Strategies in Real Complex Weighted Networks" [1]. This paper attempted to finally determine whether the "weak-link hypothesis" or the "strong-link hypothesis" was more accurate. To do this, the study took 6 networks and applied 11 different link removal strategies. The study found that a new method of link removal: betweenness centrality (Definition 6) was the most efficient way to divide the LCC. The result that betweenness centrality (BC) link removal was more efficient than weak or strong link removal indicates a new insight on the weak-strong link debate.

Additionally, Bellingeri et al. did calculate and analyze two other measurements in the networks - Total Flow (TF) and Efficiency (EFF), with a specific interest in EFF. Bellingeri et al. found that removing strong links and links with higher weighted betweenness centrality (Section III C) reduced the networks' EFF the greatest. Contrastingly, these strategies barely fractured the graph's LCC. Bellingeri et al. then came to the conclusion from the results of the EFF and LCC that removing strong links and following the weighted betweenness centrality (BCw) link removal strategy would leave the network connected, but inefficient.

C. 6 Link Removal Strategies

This section introduces the six main strategies used for link removal in epidemiology. These strategies are adapted from Bellingeri et al.'s "A Comparative Analysis of Link Removal Networks in Real Complex Weighted Networks" [1]. Since this article mainly uses these strategies for social networks, most of the strategies have been adapted for our research into safe zones.

1. Random: Links are randomly removed without regard to weight, importance, degree, etc. Random link removal is typically not used because of its variability and large degree of uncertainty

2. Strong: Links are removed in decreasing order of link weight (Definition 5). The link with the highest weight is removed first.

3. Weak: Links are removed in increasing order of weight. This strategy is inversely related to strong link removal. Thus the link with the lowest weight is removed first.

4. BC (betweenness centrality): Links are removed according to their betweenness centrality (Definition 6). Thus, the link with the highest betweenness centrality is removed first.

5. BCw (weighted betweenness centrality): Weighted betweenness centrality is similar to "unweighted" betweenness centrality, except that it takes into account the weights of links. Links are assigned weights and are removed according to their weighted betweenness centrality, i.e. links with higher weighted betweenness centrality are removed first. Therefore, to calculate weighted betweenness centrality, we find the minimum sum of the inverse of the weights of the links connecting a pair of nodes, and if the link we are examining is used in the path, we add 1 to a sum. We do this for all pairs of nodes, adding 1 to our sum if our desired link is in the path of the minimum sum of weights of links from the pair of nodes. This gives us the weighted betweenness centrality of one link.

6. Degree: Links are removed according to the sum of the degree of its adjacent nodes. (i.e the link with the highest sums of the degrees of the two nodes it connects is removed first).

D. Conclusion of Link Removal in Social Networks

There has many explorations made concerning node removal as noted in Section III A and Section III B. Thus, we can use inspiration from such strategies and observations to develop optimal strategies for link removal. Additionally, by analyzing observations from social network link removal, we can adapt and apply those techniques towards graphs concerning epidemiology.

We have also concluded from Lemma 2 and Section II D that by disconnecting the graph, at least one safe zone will be created, assuming minimal infection spread.

Finally, using the strategies listed in Section III C we will show which ones are ideal and in what circumstances they are ideal. Since the example graphs are of limited size and only consist of fewer than 20 nodes, we will only consider what will happen after the first few links removed. However, such logic can easily be applied to larger graphs by applying another strategy to disjoint subgraphs or sections of a larger graph.

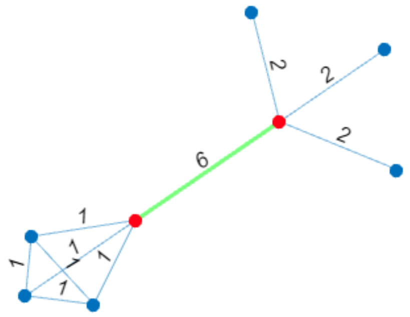

FIG. 11: In this figure, the red-colored nodes represent "senior" nodes while the blue-colored nodes represent "junior" nodes. Additionally, the green link represents a strongly weighted edge of weight 6. The blue connected edges are lighter with weights 1 and 2 as labeled. The 2 cliques are connected by the 2 senior nodes and the green edge.

Random Link Removal. To start, we can easily rule out random link removal as a general ideal strategy. It is trivial that random link removal is realistically never the best option. Randomly removing links relies stopping the infection based on chance. It is clear that it is far more effective to logically attack a certain section of the graph to stop the infection from spreading to more areas of the graph.

Strong Link Removal. In this situation, a graph with two or more cliques would work best, as shown in Figure 11. One node in each clique is labeled the "senior" node: typically a singular node within a clique which is important because it has the most strongly weighted connections to other nodes in other cliques. A real life example of a "senior" node would be a popular kid at school who is close friends with another popular kid. Within the cliques, there are also many other nodes referred to as "junior" nodes: nodes which have relatively weakly weighted ties with their connecting nodes. Such example of a "junior" node in a high school setting would be the loosely connected groups of new kids at the beginning of the year who do not know many people yet. Consequently, these "junior" nodes have weak connections with their main "senior" node as well as other "junior" nodes within their same clique. However, these "junior" nodes typically have few to no connections with any other nodes outside their own clique. Then, edges would be removed beginning with the highest weighted edge: typically one of the edges connecting two "senior" nodes. It is thus imperative to sever the strongly weighted links to form safe zones.

Weak Link Removal. This strategy works best when there are several cliques which are weakly interconnected, but the nodes within a singular clique are strongly intraconnected. Such a graph can be seen in Figure 12. Thus prioritizing removing weak connections will isolate cliques and create larger safe zones.

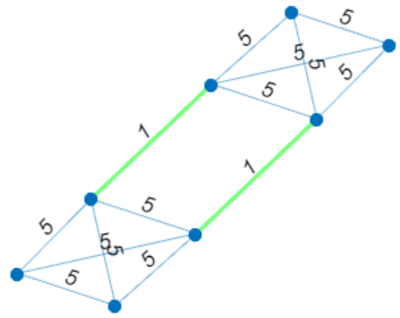

FIG. 12: A graph with 2 cliques showing the weight of the links. The green links each have a weight of 1 as shown, while each of the blue links have a weight of 5.

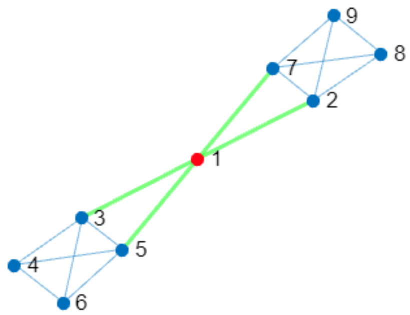

FIG. 13: A graph with 2 cliques showing the highest link and node BCs. The red node has the highest node BC while the green links have the highest link BC.

Such a graph may represent the connections between two or more family friends. Each clique would represent an individual family unit with strong ties to each family member and weak ties to other families.

BC Link Removal. In this case, link weights are not considered. This is because either all the link weights are the same or nearly identical and can thus be disregarded. Typically this strategy is the best when we have a graph where several cliques containing many nodes are connected by a single node. Thus, in order to limit infection spread, we must remove links surrounding such nodes. The best link removal strategy is clearly BC as link weights are not considered, and the connecting vertices have the highest BC, so they are the ones we want to remove. Therefore, BC link removal is the best.

An example where BC Link removal is the best is located in Figure 13. This graph shows the shows the node with the highest BC in red, and the links surrounding it in green. It should be noted that the green links have the greatest link BC because they are a part of the most direct paths connecting two pairs of vertices.

Such a graph may depict a social group. Each clique represents a separate friend group while the red node represents a person who is friends with two people from each group.

BCw Link Removal. In weighted betweenness centrality link removal, we combine the principles of both strong link removal and BC link removal. While BCw and strong link removal strategies can overlap in their approach, there are scenarios where BCw emerges as the superior strategy. Such a situation can be found in the faculty environment in a high school setting, where teachers maintain moderately weighted connections with their peers, while their students foster stronger relationships outside the school setting through frequent interactions.

However the most important aspect of such graphs are the differences in link weights. While some of the links with a large BC are weighted highly, some are weighted far less. For instance in our example, the highly weighted personal connections between teachers are key. Therefore, BCw links removal is the best strategy, as we want to remove the connections between the teachers, which are the links are simultaneously the highest in weight and the highest in BC.

This example is clearly shown in Figure 14. In the figure, the teachers have strong connections with each other (green links). The teachers have a weaker personal connections with their employees. However, schoolmates in the same class have extremely strong connections with other students. Additionally, some students have a connection to another student in a different class, albeit an extremely weak connection.

Clearly, BCw link removal is the best. It removes the links between the teachers. The links between the teachers are the most important to remove as the infection will spread quickly between them. We also see that the connections between the students from different classes have extremely low weight, so information will take a long time to spread from those edges. Therefore, it is more important to remove the links between the teachers before removing the low weight links. Hence, BCw link removal is the best option. Note that BC link removal will remove the low weight links, strong link removal will remove the links between coworkers, and weak link removal will remove the lowest weighted links.

Degree Link Removal. After much thought and experimentation, we have found that degree link removal always comes short to other methods. Since it is very general, there is typically a better, more specific method depending on the type of graph as seen in the other parts of this section.

Additionally, Bellingeri et al.'s paper (A Comparative analysis of Link Removal Strategies in Real Complex Weighted Networks[1]) verifies our hypothesis, as our "degree link removal" is very similar to their Degree Product (DP) link removal. In that paper, a study showed that DP link removal never was the best link removal strategy in any of the 6 networks analyzed. No matter the percentage of links removed or the measurements (total flow, efficiency, and largest connected cluster) used to test the strategy, DP link removal always fell short. Therefore, we can confirm that our hypothesis about degree link removal, and there are always better methods.

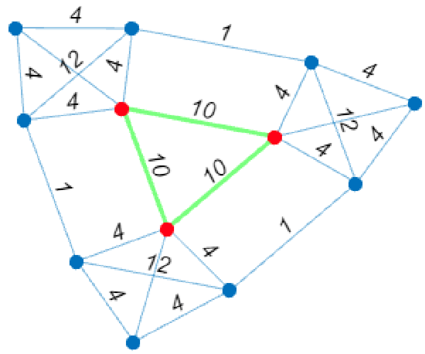

FIG. 14: A graph with 3 cliques. The green links are very strong and have a weight of 10 each. The other three edges connecting the cliques are relatively weak and have a weight of 1 each. The edges within each clique are weighted in the middle at 4 each except for one connection per clique. It is weighted the strongest at a link weight of 12. Using these weights, we have calculated that the green links have the highest Weighted Betweenness Centrality.

IV. CONCLUDING REMARKS

A. Future Work

Further work may include applying these four optimal link removal strategies to a real-life dataset. For example, we may apply these strategies using an air traffic dataset (such as one from the International Air Traffic Association) to form a graph representing air travel patterns. The practicality of such a graph was demonstrated by the COVID-19 pandemic when airports were closed in an effort to limit infection spread. Moreover, this dataset can be used to construct a network illustrating air travel connections, demonstrating a practical application of link removal strategies, similar to the concept of vaccine-driven node removal. Real data minimizes the risk of false correlations which can sometimes be observed with computer-generated data. Another noteworthy observation is the direct relationship between airport sizes and city populations, with larger cities typically hosting larger airports. This correlation underlines the direct link between a city's population and the scale of air travel activity it experiences.

Affiliations.
(a) Brookfield Academy High School, Brookfield, WI 53005, USA.
(b) University of Illinois at Chicago, Chicago, IL 60607, USA.
(c) Universidad Nacional de Mar del Plata, Mar del Plata, Argentina.

[1] M. Bellingeri, D. Bevacqua, F. Scotognella, R. Alfieri, and D. Cassi, Scientific Reports **10** (2020), URL https://www.nature.com/articles/s41598-020-60298-7.
[2] P. Zhang and G. Chartrand, *Introduction to graph theory* (Tata McGraw-Hill, 2006).
[3] M. Watabe (1998), URL https://www.sscnet.ucla.edu/soc/faculty/mcfarland/soc112/cent-ans.htm#:~:text=To%20calculate%20betweenness%20centrality%2C%20you,)(n%2D2)%2F2.
[4] D. Guichard, *An introduction to combinatorics and graph theory*, URL https://www.whitman.edu/mathematics/cgt_online/cgt.pdf.
[5] S. Arora, J. R. Lee, and A. Naor, in *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, USA, 2005), STOC '05, p. 553â€"562, ISBN 1581139608, URL https://doi.org/10.1145/1060590.1060673.
[6] M. Bellingeri, D. Bevacqua, F. Scotognella, R. Alfieri, Q. Nguyen, D. Montepietra, and D. Cassi, Frontiers in Physics **8** (2020), URL https://doi.org/10.3389/fphy.2020.00228.
[7] M. S. Granovetter, American Journal of Sociology **78**, 1360 (1973), ISSN 00029602, 15375390, URL http://www.jstor.org/stable/2776392.
[8] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó , D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási, Proceedings of the National Academy of Sciences **104**, 7332 (2007), URL https://doi.org/10.1073%2Fpnas.0610245104.
[9] A. Garas, P. Argyrakis, and S. Havlin, The European Physical Journal B **63**, 265 (2008), URL https://doi.org/10.1140%2Fepjb%2Fe2008-00237-3.
[10] R. K. Pan and J. SaramÃ¤ki, EPL (Europhysics Letters) **97**, 18007 (2012), URL https://doi.org/10.1209%2F0295-5075%2F97%2F18007.
[11] S. Pajevic and D. Plenz, Nature Physics **8**, 429 (2012), URL https://doi.org/10.1038%2Fnphys2257.
[12] R. Bhansali and L. P. Schaposnik, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **476**, 20190826 (2020), URL https://doi.org/10.1098%2Frspa.2019.0826.
[13] L. Schaposnika and A. Zhang, Letters in Biomathematics **5** (2018), URL https://doi.org/10.30707%2Flib5.1schaposnika.
[14] V. Colizza, R. Pastor-Satorras, and A. Vespignani, Nature Physics **3**, 276 (2007), URL https://doi.org/10.1038%2Fnphys560.

Appendix A: Unused Definitions

Definition 11 (Infection Expression 1)**.** (Taken from Bhansali and Schaposnik's "A trust model for spreading gossip in social networks" [12]) Consider a finite or infinite graph G and a set A_t of infected vertices, where $t \in \mathbb{N}$ and t represents the time. Additionally, let there be m amount of different types of people in a graph (such as a taxi driver, a parent, etc.) and let $k = (k_1, \cdots, k_m) \in \mathbb{N}^m$ trust vectors. Use this to assign a label in $\{1, \cdots, m\}$ to each vertex. Additionally, we define k_i for each label $i = 1, \cdots, m$ to be the minimum amount of adjacent infected vertices to vertex v where the infection is able to spread to vertex v. Finally, let $N_i(v)$ denotes the set of adjacent vertices to v in G with label i:

$$A_{t+1} = A_t \cup \{v \in V(G) : |N_i(v) \cap A_t| \geq k_i\}$$

Definition 12 (Infection Expression 2)**.** (Taken from Bhansali and Schaposnik's "A trust model for spreading gossip in social networks" [12]) Using the previously defined terms in Definition 11 as well as some new terms, we have another definition. In this definition, we define another parameter that will determine if the infection will spread — j: type of gossip to categorize a person (spreading mild, medium, etc. gossip). Additionally, let $T = \{K^1, \cdots, K^n\}$ the trust family, and $K^j = \{k^j_1, \cdots, k^j_m\} \in \mathbb{N}^m$ trust vectors. We then have a new definition using our additional parameter j:

$$A_{t+1} = A_t \cup \{v \in V(G) : \exists K_j \in T s.t. |N_i(v) \cap A_t| \geq k^j_i\}$$

Definition 13 (Safe Zone Expression)**.** (Taken from Bhansali and Schaposnik's "A trust model for spreading gossip in social networks" [12]) A definition equation, using terms defined from Definition 11 and Definition 12 above can be determined. Our definition for the *immune set* I is now:

$$I := \{v \in : \forall K^j \in T, \exists i \in [m] s.t. |N_i(v)| < k^j_i\},$$

Definition 14 (Propagate)**.** If in a graph an infection is transmitted by a minimum number of vertices with different labels to a new vertex, then the infection is said to have propagated to a new vertex. (taken from Bhansali and Schaposnik's "A trust model for spreading gossip in social networks" [13])

A term that is used to define the spread of an infection throughout a graph.

Appendix B: IATA MarketIS Data

MarketIS (MarketIS from IATA) is a data service provided by the International Air Traffic Association (IATA). MarketIS provides real, in depth data about airplane transportation. The following papers have also used the data from MarketIS:

- Bellingeri et al's "A Comparative Analysis of Link Removal Strategies in Real Complex Weighted Networks" [1] used the top 500 United States airports to create the graph in one of the networks the paper analyzed. It cited a particular paper for their graph of the top 500 US airports from Colizza, Pastor-Satorras, and Vespignani[14].
- Colizza, Pastor-Satorras, and Vespignani[14] which used the top 500 US airports to create a realistic metapopulation model, cited the IATA website (IATA).

Given the fact that these papers used the MarketIS dataset, it is evident that this dataset is reliable. Following are the reasons why the MarketIS data will be useful:

- Using the dataset, a real map can be generated instead of random, computer generated maps. Thus, these strategies can be directly applied to real world situations.
- In epidemiology, lots of infections are transferred by way of air travel. For example, COVID-19 shut down airports and air travel (especially international air travel) as an attempt to limit the spread of the virus.
- Given that vaccines are seen as the practical application of node removal, this dataset can be used to form a network regarding air travel, a practical application of link removal.
- Computer generations can sometimes lead to false correlations. With a real data set, false correlations are far less likely.
- Airport sizes are very reflective of the population size of a city. Typically, the larger cities are the ones with larger airports. Thus, larger cities have more passengers and more flights passing through it, leading to a direct correlation between population and airport size.

Using a real dataset, such as one representing air travel patterns, offers a significant advantage over computer-generated maps. These authentic maps can be directly applicable to real-world scenarios. In the context of epidemiology, it's evident that many infections spread through air travel, as exemplified by the COVID-19 pandemic, which prompted the shutdown of airports and international travel to curb the virus's transmission.

***[Note that this paragraph is in the "Future Work" subsection] Moreover, this dataset can be used to construct a network illustrating air travel connections, demonstrating a practical application of link removal strategies, similar to the concept of vaccine-driven node removal. Real data minimizes the risk of false correlations which can sometimes be observed with computer-generated data. Another noteworthy observation is the direct relationship between airport sizes and city populations, with larger cities typically hosting larger airports. This correlation underlines the direct link between a city's population and the scale of air travel activity it experiences.