Reducing Influenza Propagation Through Air Travel Pathways*

Juan Ricardo Grau[1] and Marisa Pia Kwiatkowski[2]

Abstract— **During "flu season", influenza, a common infectious disease, spreads rapidly across the United States and throughout the world. In this work, we explore methods to curb the spread of the flu throughout air traffic networks, a frequent culprit of flu transmission. Using link-removal of edges in real air traffic network datasets, we model a delay of the spread of the flu throughout the network. In this work, we analyze some of the existing methods for link-removal that use properties inherent to the network, including Jaccard index, betweeness centrality, and MinAtRisk, and evaluate their performance using discrete-stochastic simulations. We use these methods to create simulations to measure impact, and look toward optimal ways to inhibit the spread of infection through networks.**

I. INTRODUCTION

A. *Background on Influenza*

Influenza, commonly known as the flu, is a mild to severe upper respiratory virus that infects people across the world, hitting the United States particularly hard between December and March. Because of the virus' swift ability to mutate and resulting mismatch between the vaccine specimen and the circulating virus, the flu vaccination doesn't fully protect against illness. Thus, in the US winter of 2017-2018, approximately 900,000 people were hospitalized from the flu and 80,000 deaths occurred from the flu and related complications [1].

In general, the flu is easily transmitted through human-to-human interactions, with virus molecules caught in water droplets from coughs or sneezes or transmitted via surfaces. Furthermore, people infected with the flu are most contagious a day before the start exhibiting symptoms, increasing the chances of flu spreading unbeknownst to the host.

B. *The Airport Problem*

The spreading patterns of the flu make public places danger zones for flu transmission. In particular, airports are a prime breeding ground for the virus, because of the sheer number of people from different origins and different destinations that pass through airports daily, and the close proximity of people in the airplane's closed environment.

There are a variety of ways in which flu transmission can be curbed throughout the airport system. On the individual scale, these methods include: preventing or discouraging people with visible symptoms from flying, cleaning and disinfecting public surfaces, and providing masks for people on flights. Zooming out, on a higher level, these include: cancelling flights and shutting down airports to see more widespread impact.

Looking at the ways in which flu propagation can be curbed, we can help set the groundwork for protecting people across the world from other serious epidemics, ones that spread faster or have a higher infection rate, and where swift action can be critical to saving lives.

Overall, we will look into existing methods such as the Jaccard index and betweenness centrality to selectively remove the links which are most susceptible to transmitting diseases and evaluate them on real flight-traffic network data subsets. Based on these methods, we compare various ways of modeling spread and inhibiting the spread of infection.

II. RELATED WORK

A. *Link Removal*

There are several existing papers attempting to accomplish the task of reducing flu transmission

---

*Project for CS224W, Stanford University

[1] J. Grau is with the Department of Computer Science, Stanford University. `jrgrau at stanford.edu`

[2] M. Kwiatkowski is with the Department of Computer Science, Stanford University. `marisapi at stanford.edu`

through link removal.

In 2014, Nandi and Medal used mixed-integer-programming formulations (instead of the non-linear formulations used in other papers) of different models to inhibit the spread through a network [2]. The first two models are aimed at optimizing the number of connections between nodes. These models are the MinConnect model, which minimizes the connections between infected and susceptible nodes, and the MinAtRisk model, which minimizes the number of susceptible nodes. The second two models try to optimize the transmission paths in the network. The MinPaths model tries to maximize the number of transmission paths removed from the network, while the MinWPaths model tries to minimize the weighted number of transmission paths between all infected nodes and all susceptible nodes. They implemented greedy algorithms for each of these models which allow them to get within 5 percent of the optimal solution with a reasonable run time. They compare their four different models in a variety of starting simulations. Each simulation starts with a different percentage of infected nodes and they are allowed to remove only a certain percentage of links. They compare their models with a model that randomly deletes edges, a model that greedily removes links with the highest contamination degree, and a model that removes based on betweenness centrality scores.

One flaw in this paper is that all of the networks that they tested their algorithms on were artificially generated. As a result, it is difficult to draw a result that could apply to real-world situations such as the spread patterns of real infectious diseases over flight networks. On a purely mathematical basis, their methods seem to work well. However, the simulations they use (susceptible-infectious and susceptible-infectious-recovered) are quite simplistic. In their simulations, nodes will infect a neighboring node with a pre-determined transmission probability.

In addition, in 2012, Marcelino and Kaiser built on the research behind link removal [3]. Similar to other contemporary papers studying flu spread in airline networks, they recognized that removal of hubs or high-traffic airports is not the most practical solution, and instead worked to identify key flights and remove those so as to cause less of a disruption on the system. They also modeled the spreading of H1N1 through the network, taking into account the population of the airports surrounding cities, time for the flu to be at peak infection power, and interactions between individuals. For the edge removal strategies, they employed a model to remove a certain amount of connections depending on characteristics of the edges using betweenness centrality and the Jaccard coefficient. With a target of removing 25 percent of the connections and measuring impact, the results showed a decrease in infected population of 37 percent for edge betweenness centrality and 23 percent for the Jaccard coefficient, compared to only 18 percent for the hub removal strategy [3].

This paper has limitations, including computationally heavy measurements, a limited data set that treats all airport connections with equal weights and frequencies, and unrealistic assumptions of the flu originating in Mexico city, when in reality its likely that the flu enters the airline network from multiple starting places and spreads in parallel from those multiple origins. We overcome some of these limitations in our work.

B. Betweenness Centrality

Focusing in on using betweenness centrality as a metric, in 2001, Brandes discusses new methods of optimizing for betweenness centrality [4]. Brandes outlined the method of determining central nodes by looking at nodes that are hit in the most shortest paths between all start points and end points on the graph. The article also makes comparisons to other centrality measures including: graph centrality, closeness centrality, and stress centrality. Brandes introduces the concept of dependencies of one node on another when calculating shortest paths which means that one can determine the betweenness centrality index by solving one single-source shortest-paths problem for each vertex, a great improvement [4]. And another tactic for increasing speed is to split the graph into biconnected components and solving for those first. Brandes also provided evidence based on real tests for the algorithms developed in the article.

Overall, Brandes tested these new algorithms and measured the speed on undirected and unweighted random graphs as well as on one weighted directed graph. However, it would be more helpful to see if this holds for not only randomly generated graphs and the naturally-occuring social network, but also real human-engineered networks like flight patterns, and on diverse sets of data with both high and low clustering / density / diameter / etc. The network sample that these tests were run on also had a low density and low out degree on average for nodes, making harder to extend this runtime optimization to other network types, and making the evaluation less robust or realistic for this case.

III. METHODS

A. *Data Collection*

We gathered our data from two sources. First, the sample data set used in the Marcelino study with 500 global airports, and second, a weighted network of flights between US airports in 2002, weighted by available seats between two airports over the course of the year.

Having data from both of these sources was helpful for comparison, both between a global and US dataset, and between the constructed weighted and unweighted networks from these two sources. Working with both of these sources allowed for a wider view of general flu persistence and growth, in conjunction with the effects of edge removal.

B. *Network Construction*

The first step was constructing the network structures from the data sets, using the Snap.py and NetworkX frameworks. To work with the Marcelino unweighted global dataset, we processed the csv files downloaded from the research site. To work with the US weighted dataset, we created loaded the edge list, and stored alongside it, a separate dictionary of edges to their respective weights (weights calculated from amount of traffic along that flight path). We calculated constant metrics for both of the two graphs, using the edge weights for the weighted graph, allowing us to compare the two.

We decided to also test our link removal methods on a randomly generated graph with 50 nodes and 150 edges. We decided to do this after we discovered that the min-at-risk algorithm would not run in reasonable time on a graph with 500 nodes and thousands of edges (this was also mentioned by Nandi et al. [2]). While this graph is not as accurate of a representation of our problem as the US airlines weighted graph or the global airline graph, we thought it would give us a sense of how the min-at-risk algorithm would compare to our other link removal methods in reducing the number of infections of the flu virus. The random graph is directed and unweighted, and generated as an Erdos Renyi random graph.

C. *Equations*

Nandi et al. model the problem of inhibiting the transmission of a virus through a network as a mixed-integer-program, where the goal is to minimize the number of susceptible nodes in the network by repeatedly removing edges [3]. They refer to this approach as the MinAtRisk model.

The MinAtRisk mixed-integer-program is as follows:

$$\begin{aligned}
min \quad & \sum_{i \in S} z_i \\
s.t. \quad & x_{ij} + y_{ij} \geq 1 \ \forall (i,j) \in A \\
& x_{ki} - x_{kj} + y_{ij} \geq 0 \ \forall (k,i) \in \Omega, \ \forall j \in N_i, \\
& \qquad\qquad k \neq j, \ j \notin I \\
& \sum_{(i,j) \in A} y_{ij} \leq b \\
& x_{ij} \geq 0 \ \forall (i,j) \in \Omega \\
& y_{ij} \in 0,1 \ \forall (i,j) \in A \\
& z_i - x_{ij} \geq 0 \ \forall i \in S, \ j \in I
\end{aligned} \tag{1}$$

While Nandi et al. don't explicitly solve this optimization problem, they present a greedy algorithm to solve it in a more reasonable amount of time.

D. *MinAtRisk Algorithm*

We use the Greedy Algorithm to solve this MinAtRisk problem [2]. In this algorithm, we take a set of infected nodes I, a budget for how many edges we are allowed to remove b, and the number of times we are going to randomly remove a set of links M. In [2], I has a predetermined size and is then randomly selected from N, and M is always set to 100. The set of susceptible nodes S is the

Algorithm 1 Greedy Algorithm to minimize the number of nodes at risk of infection in a network

```
procedure GREEDYMINATRISK(G)
    $I \leftarrow$ set of infected nodes in N
    $b \leftarrow$ number of edges to remove
    $M \leftarrow$ sample size
    $L \leftarrow \emptyset$
    while $|L| < b$ do
        for $i \in [1, |E|]$ do
            $E \leftarrow E \backslash E_i$
            $TR_{E_i} \leftarrow 0$
            for $j \in [1, M]$ do
                $TL \leftarrow b - |L| - 1$ Rand edges
                $E \leftarrow E \backslash TL$
                $R_{E_i} \leftarrow \{E_{ij} | i \in I$ and $j \in S\}$
                $TR_{E_i} \leftarrow TR_{E_i} + R_{E_i}$
                $E \leftarrow E \cup TL$
            $E = E \cup E_i$
            if $TR_{E_i} < TR_{best}$ then
                $i_{best} \leftarrow i$
        $L \leftarrow L \cup E_{i_{best}}$
        $E \leftarrow E \backslash E_{i_{best}}$
    return $L$
```

set of all nodes that have one or more connections to an infected node.

Essentially, this algorithm goes through all the edges of the network, and removes the edges that contribute most to the number of at risk nodes. At each iteration of the while loop, we remove the edge that results in the greatest decrease in the number of susceptible nodes when it is removed from the network. (See **Algorithm 1**)

It is important to note here that we do not remove any edges that would disconnect the graph. We decided that it is important for the graph to stay connected to make sure people can travel between any two vertices in the graph. This applies both to the MinAtRisk algorithm and also to all subsequent link-removal methods - betweenness centrality and Jaccard algorithm.

E. Betweenness Centrality Algorithm

One measure used by Marcelino et al. to determine which edges to remove from the network is betweenness centrality. Centrality scores are meant to measure the 'importance' of vertices or edges in a network. The betweenness centrality score calculates the importance of a node or edge using the concept of 'shortest paths.' In an unweighted network, the shortest path between to vertices is the path that traverses the fewest edges. For a weighted network, the shortest path between two vertices is the path that minimizes the total sum of the edge weights along the path.

The betweenness centrality of a vertex v in a network is given by the following formula:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

The betweenness centrality of a vertex is a measure of how many shortest paths that pass through a given node. Similarly, the betweenness centrality of an edge in a graph is the number of shortest paths that traverse that edge. Here, we will use edge betweenness centrality to provide fewer disruptions to the airline networks (removing nodes would equate to shutting down entire airports, which would be a large disruption for travellers).

F. Jaccard Coefficient Algorithm

Marcelino et al. also remove edges based on their Jaccard coefficient. The Jaccard coefficient between two neighbors is given by the following formula:

$$J(x,y) = \frac{|\Gamma_x \cap \Gamma_y|}{|\Gamma_x \cup \Gamma_y|}$$

Where Γ_x is the set of neighbors of a node x. This is defined for undirected graphs, but since we are dealing with directed graphs, we needed a way to modify this so that it applies to the networks we are using. At the moment, we can calculate separate Jaccard coefficients for in and out degrees. So, instead of using just Γ_x, we calculate two coefficients, the first using Γ_x, in (ie: neighbors of x that have edges ending at x) and the second using Γ_x, out which is similarly defined. We try multiplying these coefficients together to create a Jaccard coefficient for each edge, but plan to look into other ways of distilling these two values into a similarity measure.

G. Model

In order to see the change before and after the link removal interventions, we constructed a flu spread simulation model based loosely off of Epstein et al.'s 2007 model [6]. We use the weighted and unweighted networks of airline traffic separately to predict spread of travellers, and we use these alongside stored data about the current number of people in a city who are flu infective, as well as the total number of people in a city who have been infected throughout the duration of the simulation. We will detail the algorithm for our model in the following sections.

In our model, we use a Susceptible-Infective-Recovered (SIR) model because, even though the flu can mutate, making it SIS in the long-term, it's almost impossible for it to change drastically enough to cause someone to become infected twice within the 30-day time frame of our simulation. Therefore, similar to Epstein et al., we look at the infective period as the 1 day when a person is infected with the flu, but not yet exhibiting symptoms. This is because on this 1 day, they are the most contagious, and may unwittingly spread the flu during travel. In contrast, when they have visible symptoms, they are much more likely to remain in their homes and postpone travel plans, because the flu is more severe than say a common cold. Therefore, we use this 1-day infective period for our model [6].

We also use the reproduction number of 1.7 from the Epstein et al. paper, to remain consistent for comparison in the field. We calculate the expected number of additional people infected per day, β, using the definition of reproduction number, R_0, where γ is the mean infective period [8]:

$$R_0 = \beta/\gamma$$

Other assumptions include setting a fixed city population for each of the cities in the model, constant probability of a city being infected at the start of the simulation, and a fixed probability of travelling per day for each person in the city. We also use a starting percentage of people infected in an infected city, of 0.032, based on a historical CDC report of people infected with the flu at a given time in the same year as our flight data network dataset [7].

Thus, we use the following starting values for our simulation:

$NUM.IN.CITY = 100000$
$PROB.TRAVEL.PER.DAY = 0.01$
$PROB.START.CITY = 0.10$
$START.PERCENT.INF.IN.CITY = 0.032$
$SIM.NUM.DAYS = 30$

At the onset of the simulation, we initialize the flu for a set number of individuals in multiple start cities. We ensure that each method of link removal on a given dataset uses the same set of start cities using a seeded random generator, so that we can consistently compare results. From there, those subset of travellers in all cities "travel" during the day, meaning move to a neighbor of their city of origin, chosen corresponding to the edge weights of the outgoing edges. As part of the "travel" group, travellers who are infected are now in a new city. At the end of the day, the "infecting" occurs, and all the people in a city who are infected (as noted before, all individuals are infective for one day only) pass on the flu to others in the city. At the start of the next day, we repeat the process of people travelling to new cities, and this iteration occurs for all days of the simulation.

In the implementation of our model, we use the Epstein et al. referenced equation "Infection of Susceptibles" to model the passing of the flu within cities. This equation guides calculation for the number of newly infective individuals [6]. We scale the expected number of individuals infected in a day, using the ratio of susceptible individuals, $S_i(t)$, over total individuals in the city, $T_i(t)$, where $\sum_{\tau=1}^{\tau_2} I_i(\tau,t)$ is the number of infectious individuals in the city on the current day, and λ is the rate of infection. We can use this equation, because like Epstein et al., we assume homogeneous mixing with uniformly distributed contact between individuals [6]:

$$E_i(0,t+1) = \lambda \frac{S_i(t)}{T_i(t)} \sum_{\tau=1}^{\tau_2} I_i(\tau,t)$$

IV. RESULTS

To evaluate our results, we compare the statistics and infected individuals before and after the described simulation and intervention, as well as use network visualizations to portray the effectiveness of the various strategies.

In order to visually track the changes and compare different methods, we created a visualization technique where a network's airport city nodes are labeled by color, red if "infected" and blue if "not infected". We use a threshold percentage of residents of a city with the flu in order for a node to qualify as "infected", default threshold: $THRESH = NUM.IN.CITY/1000$

We illustrate our experiments with the following: 1) table of totals of infected persons following the simulation, 2) visualization of the graph post- and pre- intervention, color-coded based on number of infected persons in the city relative to the threshold, and 3) summary statistics of each graph post- and pre- intervention.

A. Experiments

Note: The following are results from the different experiments, with the corresponding statistics below each visualization. You can note the changes between the initial and the interventions by noticing the decrease in red (cities infected above the threshold).

1) US airlines weighted graph

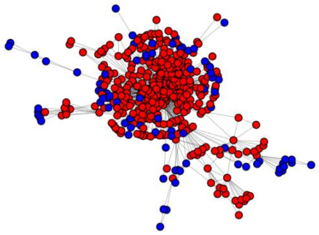

Fig 1. Initial graph. With statistics: Nodes: 500, Edges: 5960, Clustering Coef: 0.6174892374, Probability of path: 1.0, Effective diameter of graph: 3.77195306586, Diameter: 7, Avg shortest path length: 3.02814

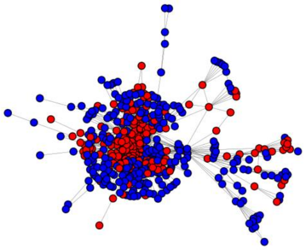

Fig 2. Betweenness centrality method. With statistics: Nodes: 500, Edges: 4470, Clustering Coef: 0.597068158343, Probability of path: 0.23, Effective diameter of graph: 3.48397260274, Diameter: 9, Avg shortest path length: 2.56248224936

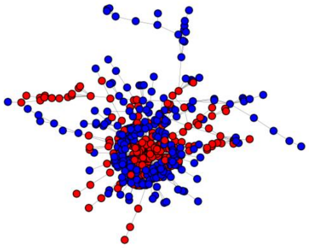

Fig 3. Jaccard method. With statistics: Nodes: 500, Edges: 4470, Clustering Coef: 0.308972242916, Probability of path: 0.41, Effective diameter of graph: 5.42301425662, Diameter: 12, Avg shortest path length: 3.60214894771

2) Global airline unweighted graph

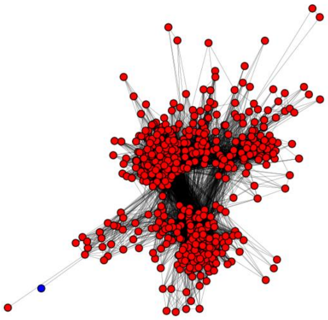

Fig 4. Initial graph. With statistics: Nodes: 500, Edges: 24009, Clustering Coef: 0.64884037988, Probability of path: 1.0, Effective diameter of

graph: 2.78642670084, Diameter: 5, Avg shortest path length: 2.30242

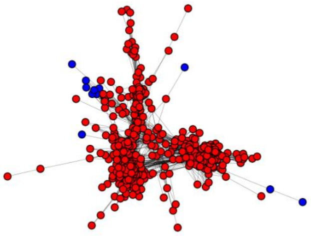

Fig 5. Betweenness centrality method. With statistics: Nodes: 500, Edges: 18007, Clustering Coef: 0.584374926898, Probability of path: 0.95, Effective diameter of graph: 4.20966214762, Diameter: 8, Avg shortest path length: 3.11229008101

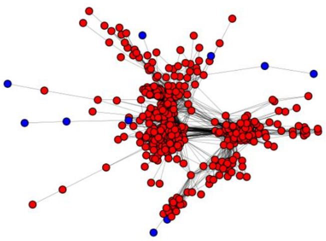

Fig 6. Jaccard method. With statistics: Nodes: 500, Edges: 18007, Clustering Coef: 0.623270376986, Probability of path: 1.0, Effective diameter of graph: 4.14071078431, Diameter: 9, Avg shortest path length: 3.09691200564

3) Random unweighted graph

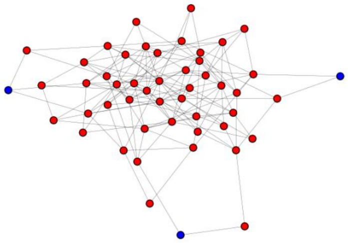

Fig 7. Initial graph. With statistics: Nodes: 50, Edges: 150 Clustering Coef: 0.144207126207, Probability of path: 0.84, Effective diameter of graph: 4.93425605536, Diameter: 10, Avg shortest path length: 3.43225806452

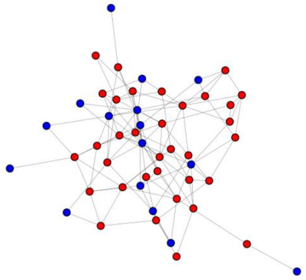

Fig 8. Betweenness centrality method. With statistics: Nodes: 50, Edges: 113, Clustering Coef: 0.144698412698, Probability of path: 0.15, Effective diameter of graph: 3.9152173913, Diameter: 7, Avg shortest path length: 2.47896440129

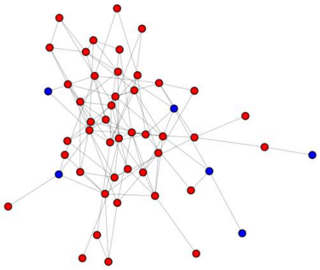

Fig 9. Jaccard method. With statistics: Nodes: 50, Edges: 113, Clustering Coef: 0.126841269841, Probability of path: 0.64, Effective diameter of graph: 4.79214876033, Diameter: 8, Avg shortest path length: 3.3877159309

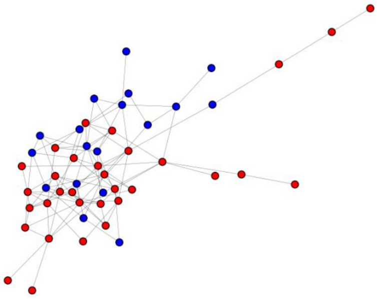

Fig 10. MinAtRisk method. With statistics: Nodes: 50, Edges: 112, Clustering Coef:

	Original	Jaccard	B-Centrality	Min-At-Risk
air 500	33.06	31.35	31.25	N/A
USAir Weighted	17.95	13.15	12.17	N/A
Random Graph	2.38	1.98	2,30	1.04

TABLE I

NUMBER OF INFECTED PEOPLE AFTER SIMULATION IN MILLIONS. (NOTE THAT EACH SIMULATION STARTED WITH 16,000 PERSONS INFECTED.)

0.123230880231, Probability of path: 0.68, Effective diameter of graph: 5.60268456376, Diameter: 10, Avg shortest path length: 3.69852034525

B. *Overall Findings*

Overall, we found that both Jaccard and Betweeness Centrality reduce the number of people infected by the flu, without drastically reducing the average shortest path. This is important because average shortest path is a metric for determining the amount of detriment cancelling these flights has on the system of travellers. (In this instance, it would be the number of connections a traveller has to take with the restricted number of edges.)

In addition, we experimented with different number of starting cities, and settled on using a percentage of the overall number, randomly selected, in order to make a more robust simulation, building off of the single initial disease outbreak starting point in the Marcelino et al. study.

Furthermore, we learned from comparing the different starting datasets, that the more densely connected graph, the Global airline network, made it much more difficult to decrease flu propogation, which points to an overall trend of: more connections and more direct travel makes it harder to curb outbreaks in an efficient and timely fashion. This has implications as the world becomes increasingly more physically connected as time goes on.

Finally, the MinAtRisk method depicts a real scenario of prior information (here, knowing the cities that are origins of the flu), which shows that detecting where the outbreaks start has great potential for being able to feed more information into the simulation model and using a method that makes more educated choices for efficient link removal.

V. DISCUSSION

A. *Limitations*

There are specific limitations that we encountered through the process, noted here:

- Evaluating effectiveness of our models on inhibiting the spread of viruses through the network. We use the metrics of total number of people infected throughout the time period, as well as global graph statistics, but our models make simplifying assumptions that make it difficult to evaluate real impact.
- Working with a limited set of airline datasets that, although they depict the interaction between major cities, don't account for within-community spreading in closer-proximity cities.
- Extrapolating the SIS model to fit spread for the networks we worked with. This included deciding to ignore the chances for someone to be vaccinated in our calculations of flu spread, as well as other simplifying assumptions about susceptible persons in the cities we modeled.

B. *Future Directions*

So far, we've gained a better understanding on what is needed to inhibit the flow of infectious diseases throughout a network, and begun to explore improvements on existing methods for network interdiction.

One interesting result that we found was that the MinAtRisk algorithm performed far better than our other two link removal methods on the random graph on 50 nodes. It would be interesting to see how the MinAtRisk algorithm performs on a real dataset, such as the US airline weighted graph. To do this, improvements would be needed in the runtime complexity of the algorithm in order to process the thousands of edges in those graphs

A direction for future work would be to improve on our simulation of the spread of the flu virus. One such improvement would be to model the populations of the cities in a more nuanced fashion. In our simulation, each city has 100,000 persons. In a more realistic simulation, some cities would have much larger populations than others, which would in turn affect how likely those nodes are to infect their neighbors (since higher population correlates with higher number of travelers).

GITHUB REPOSITORY

Visit https://github.com/marisapi/cs224w-proj to see the code for our project, including the disease simulation, algorithm implementations, data modeling, etc.

CONTRIBUTIONS

Marisa Kwiatkowski and Juan Grau contributed equally to the project, collaborating on components including crafting simulations, coding algorithms, running experiments, writing up the report and visualizing results.

ACKNOWLEDGMENT

Thank you to the CS224W instructors for all their support of this project, to openflights.org and Marcelino et al.'s research group for providing access to the datasets for this project, and to Overleaf for the research paper template.

REFERENCES

[1] National Foundation for Infectious Diseases (2018). Influenza. Retrieved from http://www.nfid.org/influenza

[2] Nandi, A. K., Medal, H. R. (2016). Methods for removing links in a network to minimize the spread of infections. Computers Operations Research, 69, 10-24.

[3] Marcelino, J., Kaiser, M. (2012). Critical paths in a metapopulation model of H1N1: Efficiently delaying influenza spreading through flight cancellation. PLoS currents, 4.

[4] Brandes, U. (2001). A faster algorithm for betweenness centrality. Journal of mathematical sociology, 25(2), 163-177.

[5] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[6] Epstein, J. M., Goedecke, D. M., Yu, F., Morris, R. J., Wagener, D. K., Bobashev, G. V. (2007). Controlling pandemic flu: the value of international air travel restrictions. PloS one, 2(5), e401.

[7] CDC (2002). 2001-02 Influenza Season Summary. Retrieved from https://www.cdc.gov/flu/weekly/weeklyarchives2001-2002/01-02summary.htm

[8] Wikipedia (2018). Basic reproduction number. Retrieved from https://en.wikipedia.org/wiki/Basic_reproduction_number.