Genetic Algorithm Approach to Optimizing the Air Traffic Network during Spatial Hazards

JI HOON "ANDY" KIM (JKIM4223)

1 INTRODUCTION AND PROBLEM STATEMENT

With the explosive increase in the number of passengers that utilize the air transportation network each year, it is critical to assess the quality of the network as well as improving its efficiency to facilitate the rising number of travelers per year. The air transportation network has not only geographical impacts with human migration but also significant economical and political influences as well. In literature, analysis has been done to assess the network structure as well as to investigate the vulnerability and resiliency of these air traffic networks under spatial hazards. In this paper, spatial hazards include two types of hazards: natural disasters which target a cluster of nodes in the network, as well as isolated terrorist incidents which target nodes with high clustering and/or high degree.

Although the literature is comprehensive in the analysis of the network structure and resiliency of the air transportation network, minimal work has been done to generate and evaluate optimal network configurations during these spatial hazards. We set out to find the solution to the problem: what method can we use to generate optimal network configurations in the event of a spatial hazard? We will define optimality of a network under various metrics such as average shortest flight path distance and robustness. The approach that will be used to generate these optimal networks will be to investigate the genetic algorithms mentioned by Lordan et al. to breed the best network configurations as per the metrics above. In addition, the simulations for spatial hazards will be similar to that used by Wilkinson et al.

2 LITERATURE AND BACKGROUND

Before diving into understanding the optimal network configurations, we will first define some of the work completed in the field as well as metrics for evaluating our network. Guimer et al. [R. Guimer and Amaral 2005] demonstrated that the world shows a global community structure in the air traffic network where different airport hubs across the world are scattered linking these community structures. The importance of each node in the graph was evaluated using different measures of centrality and betweenness. Centrality in this case is the average degree centrality of a network (number of connected neighbors given a node i); betweenness is defined as the average betweenness of the nodes in the network (given a node v, how many shortest paths s and t pass through v). In addition, measures of degree distributions were calculated across the air transportation network, showing a scale-free power law-abiding network. We will be using similar metrics to evaluate the important hubs in our air transportation network.

Lordan et al. [O. Lordan 2014] reiterates much of what was seen by the results from Guimera, noting that there are a group of hubs that seem to dominate the worldwide air transportation network. However, in this paper, an important metric is described: the definition of robustness. We note that an air transportation network is robust if there is minimal service disruption with certain nodes in the network removed. This means that if node A is connected (has a path) to node B, the network is robust if, after removing a portion of the network, node A and B are still connected for all nodes across the graph. We will be using the same metric to measure robustness and better define it using notation and mathematical terms.

Wilkinson et al. [S. M. Wilkinson 2012] describes how to measure the vulnerability of a network to spatial hazards. A circular spatial hazard (such as a hurricane or volcano) is placed on the map with a certain radius. Any nodes within this radius geographically is removed from the network. Although Wilkinson describes how to model this along with various network statistics, there is little work on how to reconfigure the network to mitigate delays and impacts in the event of these spatial hazards. In this paper, we will be describing how to optimize the network with regards to the metrics described above.

Hu et al. [X. Hu 2007] demonstrates how some of these genetic algorithms can be used to optimize air traffic networks. For this paper, we will be adapting the practices from the results and algorithms created by Hu, and use this to optimize our networks in the event of a spatial hazard.

3 NOTATION AND METRICS

We will describe how some of the metrics are computed as well as notations that will be used for the paper.

3.1 Robustness

As described in the previous section, a network is robust if there is minimal disruption (i.e. most of the nodes are connected) when a certain number of nodes are removed. To empirically measure robustness, we first record the fraction of number of nodes in the largest strongly connected component (SCC) compared to the number of nodes in the graph. Using the metrics of degree centrality, PageRank and degree of the nodes, we define the most important (highest values for the metrics) nodes. Then, we remove the nodes in order of highest to lowest importance for all three metrics and compute the fraction of number of nodes in the largest SCC for each removal. This is performed until all of the nodes are removed from the network.

We then generate a curve of the fraction of number of nodes against percentage of the network removed [O. Lordan 2014] as seen in Figure 10. The robustness metric, R, is then the area under the curve which is computed through numerical methods (Riemann sums). We give intuition to this metric by understanding that the most robust networks will show high amounts of connectedness (nodes connected to each other) even if large portions of the network are remove. This is precisely what this metric is encapsulating.

3.2 Average Shortest Flight Path Distance

The Shortest Flight Path Distance is defined as the shortest distance to fly between two cities. For example, if city A and city B are 500 miles apart and there is a direct connection between the cities, then the shortest flight distance is 500 miles. If the shortest connection between city$A$ and city B is through a node C (i.e. a layover), then the shortest flight distance is the flight distance from city A to C plus the flight distance from C to B. We will call this function $SFPD(A, B)$. For a graph $G = (V, E)$, The average shortest flight path distance, K, is then [X. Hu 2007]:

$$K = \frac{1}{N(N-1)} \sum_{i,j \in V, i \neq j} SFPD(i, j)$$

where $N = |V|$ is the number of nodes in the graph.

4 DATA OVERVIEW

With the metrics outlined in the previous section, we now provide an overview of the data collection process and a preliminary analysis of the network and its structure.

4.1 Data Collection

The data is collected from the public database on https://transtats.bts.gov which maintains all the flights in a given year. For the purpose of this dataset, we collected the information about a single day, June 1st, 2017. The data is given in a CSV format and includes 16482 entries/rows corresponding to the 16482 flights that occurred in the United States airspace on that given day. In this dataset, we have six features per row: origin airport ID, destination airport ID, longitude of origin airport, latitude of origin airport, longitude of destination airport, and latitude of destination airport. Using these features, we load the data into the SNAP framework with each node in the graph corresponding to an airport and each edge corresponding to a flight path that occurred on June 1st, 2017.

Loading in the graph, we first note that multiple flights on the same path equate to one edge. Furthermore, since very rarely flights are non-reciprocal, we force the condition that our graph is undirected [R. Guimer and Amaral 2005]. Furthermore, we note that there are 291 airports/nodes in our dataset spread across the United States with 1884 unique flight path/edges in our network. A visual representation of the data is shown in Figure 1.

4.2 Preliminary Analysis and Statistics

Before we dive into tackling the optimization of the network, we take a look at a couple of network statistics to get a better idea of the underlying structure. First, we note the log-log plot of the degree distribution as shown in Figure 2. We confirm that the network is scale-free considering that the points seem to follow the power law distribution. Deriving a maximum likelihood estimate for our parameter α, we show that $\alpha = 1.659$ in our network.

We also compute the degree centrality, PageRank score and degree of our network for each node and order the nodes from the highest value for the metric

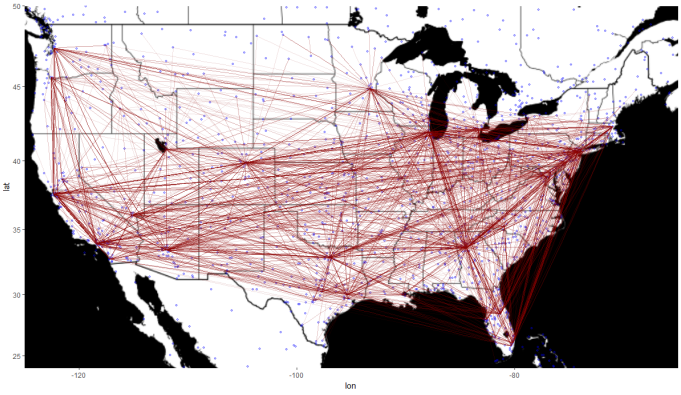

Fig. 1. The United States airline network. The airports/nodes of the graph are marked with a blue circle and the flights are marked with a red line.

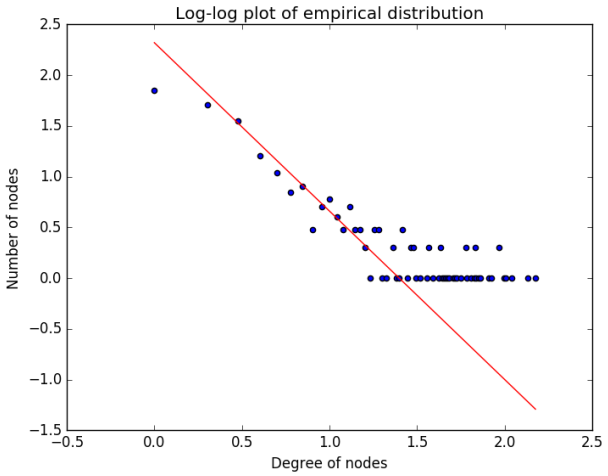

Fig. 2. Log-log plot of the degree distribution. An estimate of the power α is plotted on the figure as a red line.

to the lowest. We deem the nodes with the highest values for these metrics to be the most "important" nodes and vice versa. The five most important airports/cities found by all three metrics are Atlanta, Chicago O'Hare, Denver, Minneapolis-St. Paul and Detroit respectively.

Using these rankings, we can compute the robustness of the network by removing random nodes and measuring the fractional size of the largest strongest connected component as described in the Metrics section. A plot of this is found in Figure 10. We note that the three metrics are largely in agreement with one another.

Furthermore, we compute the average network eccentricity. This is done by averaging the eccentricity values for each node (eccentricity is defined as the largest shortest-path distance from the node to any other node

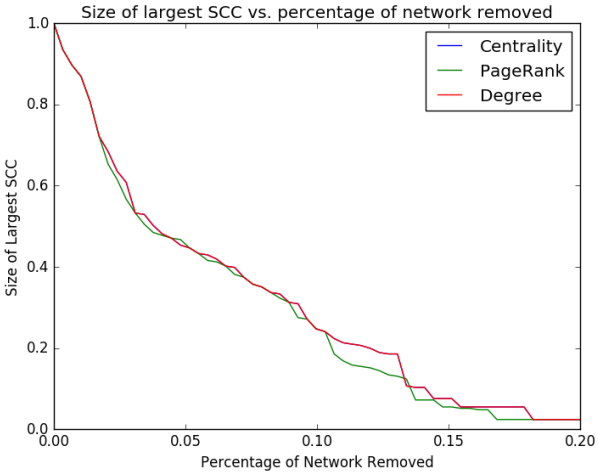

Fig. 3. Robustness of the current US Airline network

in the graph. With our metrics, we determined that the average eccentricity of the network is 3.759.

Lastly, we compute the average Shortest Flight Path Distance (from now on will be referred as SFPD). However, we note that this computation requires $\binom{291}{2}$ graph searches and will computationally take much too long. Therefore, we choose 1000 pairs across our network and compute SFPD and average this across these 1000 results. We will expound upon chi-squared statistical significance in the Results section.

5 MODEL ALGORITHMS

With this preliminary analysis, we move onto how we will be optimizing the network using genetic algorithms [X. Hu 2007].

5.1 Genetic Algorithm Background

5.1.1 Background. First, we define the notion of a chromosome. In genetic algorithms, a chromosome is an array that represents an object we wish to optimize; in our case, this chromosome corresponds to a network configuration. The array is composed of 0's and 1's and is of length L. Each cell in this chromosome is called a gene. Therefore, referring to the 10-th gene will mean the 9-th index in the chromosome.

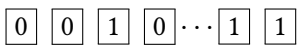

Fig. 4. A single chromosome

For the start of the model, we generate C chromosomes. We define parents as two chromosomes that will

be chosen to mate, i.e. paired together. Initially, we divide the C chromosomes into D groups such that D is a power of 2.

5.1.2 Selecting the Best Chromosome and Pairing up. We then compute the metric that we are optimizing with respect to (either SFPD or Robustness) for each chromosome and pick the best chromosome for each group. This will yield D chromosomes. We then pair each of the D chromosomes up as respective parents (yielding $\frac{D}{2}$ pairs).

5.1.3 Generate Children. We take each pair of two chromosomes and randomly select an index, i in the array, at this point, we make a "cut". A child chromosome is then created by concatenating the array from indices 0 to i of the first parent with the array from indices i to L (length of the chromosome) from the second parent. We generate $2D$ children per pair.

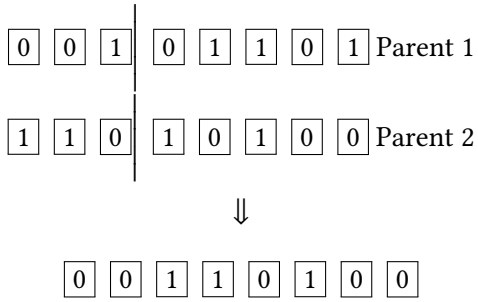

Fig. 5. An example of a child created through the process described above. The top two arrays are parents and the last array is the resulting child. Cut is made at index = 2

5.1.4 Mutation. In order to add randomness to model, after we generate the children, we mutate each one to add genetic diversity in our model. In order to perform this, we loop across our chromosome for each of the children and mutate a gene with a probability $p = \frac{1}{L}$. Note that this mutation rate is very low (expected value of 1 per chromosome) since too high of a rate will simply add too much noise in our model and will essentially perform a random search. A mutation is defined as such:

If gene i is to be mutated, we look for a random index in our chromosome that is not equal to i and swap the contents of these genes with each other. We do not flip the bits because we wish to constrain the number of 1's we have in our chromosome (not too many nor too little).

5.1.5 Collect the Children. We now merge all the children together from all the pairs into one large group, which again has size C. We then repeat the above process for a user-defined number of iterations.

ALGORITHM 1: Genetic Optimization Algorithm

```
function Simulation (chromosome, C, D, I);
Input  : A chromosome initialized from our network
         chromosome, Number of chromosomes per
         generation C, number of groups D, and number of
         iterations I
Output : The optimized chromosome
//Generates C chromosomes using mutations of chromosome
generation = generate_chromosomes(C)
count = 0
while count < I do
    groups = split(generation, D) //List of Lists, splits into D
      groups
    parents = Empty list
    for group in groups do
        //Per the metric
        chromosome = best_chromosome_in_group(group)
        parents.add(chromosome)
    end
    //Place one half of the list into one parent1 //And the
      other into parent2 parent1, parent2 = split(parents, 2)
    //Mate the parents and perform mutation
    children = mate(parent1, parent2)
    children = mutate(children)
    generation = children
    count++
end
chromosome = best_chromosome_in_group(generation)
return chromosome
```

5.2 Genetic Algorithms for Air Traffic Networks

Now, how can we encode this algorithm for use in our model? We simply have to figure out how to represent the graphs in our model into chromosomes and vice versa. This is done by creating an adjacency matrix A of the graph. If there is a flight from Airport 1 to 2, we place a 1 in the $(1, 2)$ entry in the matrix A and 0 otherwise. However, since our graph is undirected, the adjacency matrix is symmetric. Therefore, we only need to encode the upper triangle of the adjacency matrix without the diagonals (Since a flight will never depart and arrive at the same airport). We flatten this upper triangular matrix into an array by looping through the rows of the upper triangular matrix and appending the rows sequentially into a single array. We then take this array to be the chromosome representation of our graph. This process is shown in Figure 6.

If we have N nodes in the graph, the array is essentially choosing a pair out of N elements. Therefore, the

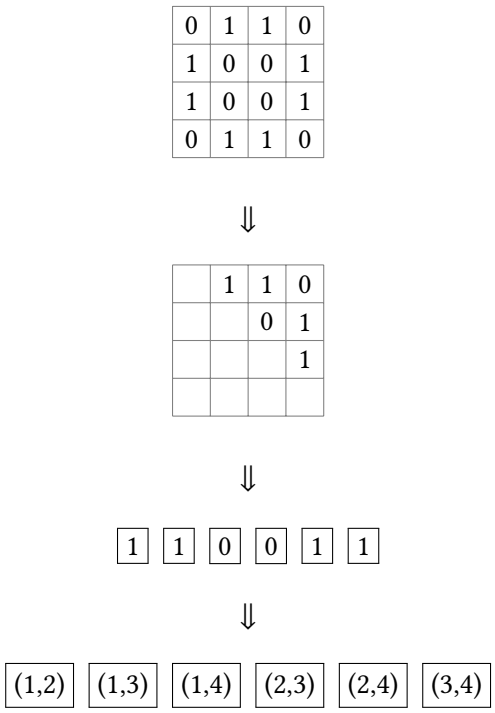

Fig. 6. Adjacency matrix turned into an array. Top down is as follows: The adjacency matrix, the reduced upper triangular matrix, flattened array, positions of each cell in the flatten array in the original adjacency matrix

length of the resulting chromosome is $\frac{N(N-1)}{2}$ elements. Note that this chromosome also provides an implicit mapping back to a graph (if we take the first element to be the $(1,2)$ entry of the adjacency matrix, the second to be the $(1,3)$ entry and so on). Therefore, to run our genetic algorithm, we convert our base graph into a chromosome and run the algorithm. With the resulting chromosome from the output of the algorithm, we convert it back into a graph which is our optimized network.

5.3 Spatial Model and Optimizing the Network

Once we have encoded the genetic algorithm to produce optimal networks, we expand our model to incorporate the spatial hazards in order to tackle the main problem. We include two types of spatial hazards in our network: natural disaster and targeted attacks. Natural disasters are events that impact airline networks in a large scale geographically: A volcanic eruption, a hurricane, a tsunami, etc... We will also be modeling targeted attacks. Targeted attacks are coordinated events that take down critical/important airports in the network. This may be through events such as terrorism or disease outbreaks and typically put the network under the most stress. In the following sections we will be describing how these are being modeled.

5.3.1 Natural Disasters. The method of incorporating natural disasters in our network will primarily mimic the work done by Wilkinson et al. [S. M. Wilkinson 2012]. For our model, we will be simulating a volcanic eruption at Yellowstone National Park (44.4280° N, 110.5885° W). The method of simulating the disaster will be as follows:

(1) Place a point on the map at the location of the spatial hazard
(2) Set a radius around the point that represents the range of impact. To simulation our volcanic eruption, we will be using a radius of 500 miles.
(3) Remove all nodes in the network that are within the radius of the impact

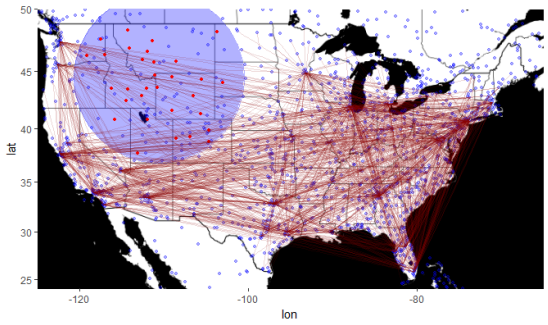

Fig. 7. Network after simulating the volcanic eruption of Yellowstone. The impacted nodes are in red with the impact radius colored in blue. Note that not all the airports in the radius are red since those airports did not have chartered flights on that day.

Figure 7 demonstrates the air traffic network after implementing the natural disaster hazard model. Note that Denver International Airport, one of the critical airports in the network, is removed and therefore we see a loss of many edges in the graph.

5.3.2 Targeted Attacks. To simulate the targeted attacks in our network, we simply compute the PageRank of all the nodes in the network. This will yield a basic idea of which airports are the most important in our network. From this point, we will see take the top five airports that have the highest PageRank values and remove them from the network. The resulting network is shown in Figure 8.

5.3.3 Optimizing the Network. We optimize the network after simulating both these spatial hazards using the genetic algorithms described in Section 5.2. In our

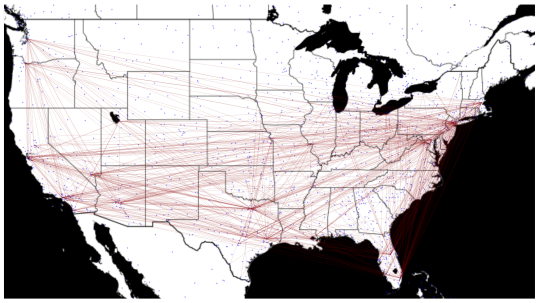

Fig. 8. Network after simulating the targeted attacks across the five most important cities by PageRank.

optimization, we make the assumption that the number of flights that the impacted network can have is the same number as the non-impacted network. This is to allow the network to reroute the flights in a different configuration rather than losing edges to the spatial hazard. In the next section, we will describe some of the experimental results after implementing these models.

6 EXPERIMENTS AND RESULTS

6.1 Genetic Algorithm Results

6.1.1 Optimization with respect to Shortest Flight Path Distance. We took the network of 291 nodes and 1884 edges and converted it to chromosome form as described in the previous section. The inputs to our model are $C = 16$, $D = 4$ and $I = 500$ (500 generations). The metric that we use to optimize with respect to is SFPD. Using this, we see the following mean SFPD statistics with 1000 pairs sampled from both the current and new networks:

$$\mu_{current,SFPD} = 1910.12\ mi$$

$$\mu_{optimized,SFPD} = 1606.09\ mi$$

With the following standard deviations:

$$\sigma_{current,SFPD} = 1486.7\ mi$$

$$\sigma_{optimized,SFPD} = 512.1\ mi$$

Although the means seem drastically different, the variances of the two populations are also very large so how can we tell if the two means are significantly different? We perform a Chi-Squared test on a t-statistic defined as:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$

where μ_1 and μ_2 are the two means respectively, s_1 and s_2 are the two sample variances (squared our standard deviation value) and $n_1 = n_2 = 1000$. Plugging in our results, we see that $t = -6.112$ with a p value of less than 10^{-4}. This means that with 99.99% confidence level that the means are statistically different. Therefore, our algorithm successfully bred a network that optimizes our target metric of SFPD.

We generate a plot of which flights were added and removed from the network in Figure 9

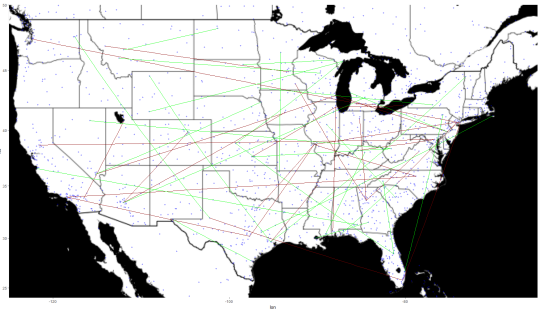

Fig. 9. Lines in green show flights that were added by the genetic algorithm. Lines in red show flights that were removed by the optimization

6.1.2 Optimizing with respect to SFPD and Robustness. Although we optimized in the last section SFPD, we note that this not the only metric we should be considering in our model. For example, if we remove all the important nodes, many parts of our network will become fragmented. By fragmenting our network, we will see the SFPD will be naturally small because many of the nodes that do not connect will not be included in the average SFPD metric. Thus, minimizing SFPD will simply yield an optimized, but fragmented, network.

Thus, in order to also reduce the fragmenting we see in this network, we add the robustness metric into our calculations. A very robust network will remain connected and not fragmented. Thus, for our genetic algorithm, we use a new metric which we will now refer to as the *score* of a graph. In our case, the score is:

$$score = \alpha K + \beta R$$

In this case, K and R, as defined in our Metrics section, are the average shortest flight path distance and robustness metrics respectively. $\alpha, \beta \in \mathbb{R}$ are coefficients that are some weighting to give each score. We determine these coefficients by sampling K and R over various generated graphs and perform ordinary least squares. We determine these coefficients to be roughly $\alpha = 2.65 \times 10^{-}6$ and $\beta \sim 1$.

We see a couple of results from optimizing with respect to this metric. We see that

$$\mu_{current,SFPD} = 1717\ mi$$

$$\mu_{optimized,SFPD} = 1786\ mi$$

With the following standard deviations:

$$\sigma_{current,SFPD} = 1226\ mi$$
$$\sigma_{optimized,SFPD} = 1250\ mi$$

We see that the flight path is no longer shortened with a significant t-test result. This demonstrates that our algorithm no longer successfully reduces flight distance. However, taking a look at robustness, we see the following results:

$$\mu_{current,R} = 0.0154\ mi$$
$$\mu_{optimized,R} = 0.0170\ mi$$

We see that the robustness of the network has significantly increased with the optimization. This may be an artifact from running ordinary least squares and determining coefficients. Regardless, tuning the parameters in the score will yield different results from run to run. However, we see that we were able to optimize robustness. Creating the same type of plot in Figure 10 for our optimized network and for the original network:

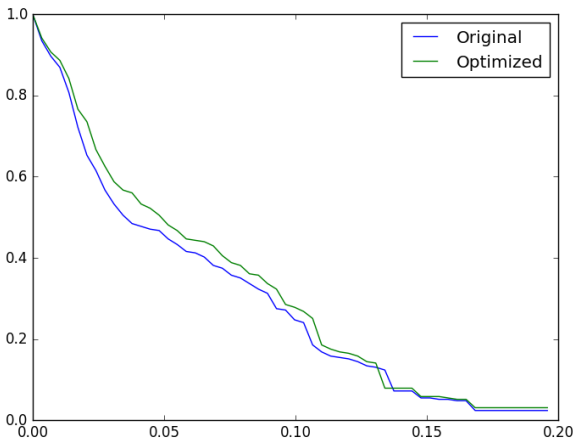

Fig. 10. Robustness of the optimized US Airline network versus the robustness of the current US Airline network.

This further confirms the increase robustness of our network. Since we have not incorporated the spatial model; our analysis for robustness is cursory in this section and will be more thorough in the next section (Thus, we will not cover significance tests in this section).

6.2 Optimizing the Spatial Hazards Networks

We now incorporate the models described in Section 5.3 and attempt rerouting the network in the event of these spatial hazards.

6.2.1 Optimizing the Natural Disaster Model. As described in the previous sections, we simulate the Yellowstone eruption by removing all nodes within a 500 mile radius. This updates our network to include only 260 airports. Running the genetic algorithm 100 times across this network and generating 100 optimized networks, we generate the following statistics:

$$\mu_{current,SFPD} = 2157\ mi$$
$$\mu_{optimized,SFPD} = 1527\ mi$$

With the following standard deviations:

$$\sigma_{current,SFPD} = 3405\ mi$$
$$\sigma_{optimized,SFPD} = 3200\ mi$$

And with respect to robustness:

$$\mu_{current,R} = 0.01564\ mi$$
$$\mu_{optimized,R} = 0.01941\ mi$$

With the following standard deviations:

$$\sigma_{current,R} = 0.00623\ mi$$
$$\sigma_{optimized,R} = 0.00781\ mi$$

We see that we were significantly able to reduce the average shortest flight path distance; however, performing a significance test, we obtain a p value of 0.1791. Although the means are quite different, the variance is large enough that the results are not considered statistically significant. Performing the same significance test with respect to the robustness metric, the obtained p value is 0.0002. Therefore, we can say at a 95% confidence level that we have increased the robustness of the network. We see a plot of the robustness in 11 and a plot of the added and removed flights in Figure 12.

6.2.2 Optimizing the Targeted Attack Model. After removing the top 5 nodes according to PageRank in the model, we perform the genetic algorithm 100 times once more, generating 100 optimized networks. We obtain the following statistics with this model: With respect to the shortest flight path distance

$$\mu_{current,SFPD} = 1212\ mi$$
$$\mu_{optimized,SFPD} = 2335\ mi$$

With the following standard deviations:

$$\sigma_{current,SFPD} = 2805\ mi$$
$$\sigma_{optimized,SFPD} = 2308\ mi$$

Performing t-tests much like in the previous sections, we compute $p = 0.002$. Why does the shortest flight path distance significantly increase with the optimization?

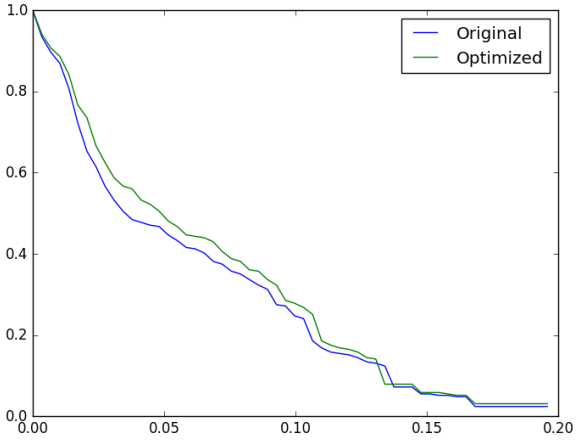

Fig. 11. Robustness of the natural disaster network versus the robustness of the optimized natural disaster network

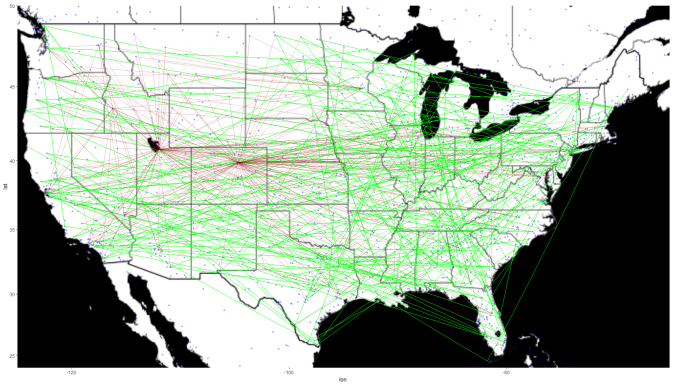

Fig. 12. Added and removed flights in green and red respectively between the original network and the natural disaster spatial hazard optimized network.

We note that the small SFPD for the non-optimized network is skewed. This is because the network is largely fragmented (since we removed the most important nodes) to start with; therefore, any connected networks naturally have a short flight path distance. Expectedly, rerouting the networking will connect more nodes together and thus, increase average shortest flight path distance. However, decreasing fragmentation can be see in the robustness metric:

$$\mu_{current,R} = 0.01138\ mi$$
$$\mu_{optimized,R} = 0.01539\ mi$$

With the following standard deviations:

$$\sigma_{current,R} = 0.00632\ mi$$
$$\sigma_{optimized,R} = 0.00334\ mi$$

The standard deviations yield a p value of less than $10^{-}5$ and therefore, at a 95% confidence level, we can say the robustness metrics are significantly different. Therefore, at the cost of connecting fragmented sections of the network together and increasing average shortest flight path distance, we significantly increase the robustness of the network.

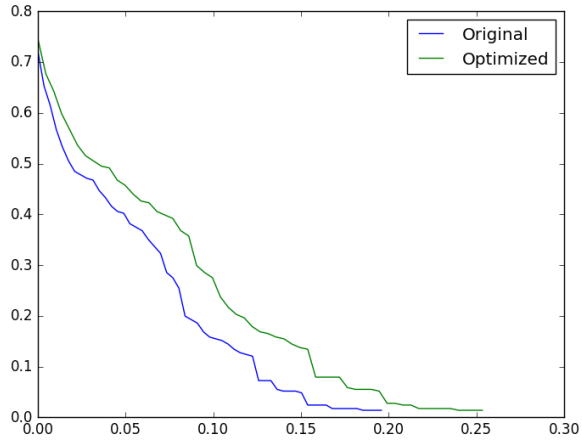

Fig. 13. Robustness of the targeted disaster network versus the robustness of the optimized targeted disaster network

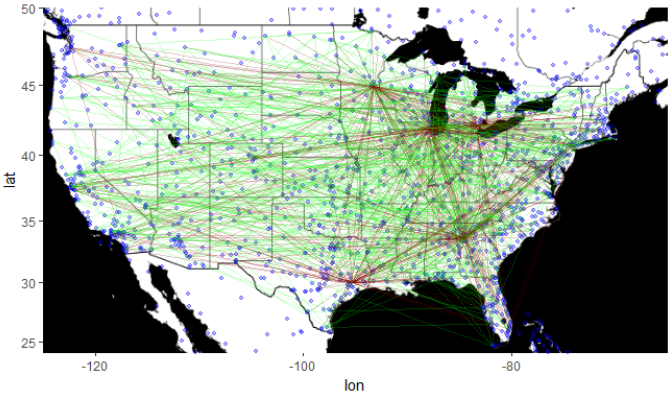

Fig. 14. Added and removed flights in green and red respectively between the original network and the targeted disaster optimized network.

7 FURTHER WORK AND CONCLUSION

The initial genetic breeding algorithm demonstrates relatively successful results considering it is able to optimize our network with respect to a metric we feed into the model. However, it is evident that there are some underlying tuning issues with the genetic algorithm as seen with the inability to reduce both metrics at once. The stochastic search nature of the algorithm renders it particularly susceptible to high variance and makes it inherently difficult to correctly optimize. With that said, although many techniques from various papers were included, there are many areas that could be improved with respect to optimizing the air transportation network around spatial hazards.

For example, metrics other simply using the shortest flight path distance and robustness. As we saw in the previous section, there are some inherent issues with simply using SFPD as a metric. Therefore, it is worth to note to exercise caution when looking at different metrics. Other metrics under consideration are Bonacich Centrality, betweenness, and eccentricity.

Although a full spatial model is realized, there are many assumptions to our model. This includes that the disaster is perfectly circular which in fact will most certainly not be the case. Another critical assumption that is made that may completely change the model is that there are exactly the same number of flights that are available for us during the event of a spatial hazard. In our case, this is 1884 flights and we assume for all of the models that this is the exact number of flights that are available to us. other assumptions also include the fact that all flights are undirected and always have bidirectional paths. Perhaps a motif analysis (a three-way flight) would yield a better analysis.

Adding some of idea of an inherent community structure would assist in the modeling effort as well. For now, the genetic algorithm assumes that any flight can be rerouted. However, this most certainly is not the case and therefore, understanding the underlying community structure may yield additional information.

In addition, there is no related work in this paper including any sort of temporal analysis. Since spatial processes are highly intertwined with time series, it would be negligent to not realize the impact of a temporal aspect. For future iterations of the project, a sensitivity analysis with respect to time may yield additional results as well as a more accurate spatiotemporal model.

Finally, using different methods of optimizing the network during a spatial hazard should be considered. Since the search space of an optimized graph is so large, something similar to reinforcement learning might be able to tackle the optimization problem well. Other techniques to optimize this network might be to use a neural network with a cost function that has the metrics as a parameter. Regardless of the technique, exploring other algorithms to navigate the search space is key for future iterations to optimize the traffic network.

REFERENCES

P. Simo D. Gonzalez-Prieto O. Lordan, J. M. Sallan. 2014. Robustness of the air transport network. *Transportation Research Part E.* 68 (2014), 155–163.

A. Turtschi R. Guimer, S. Mossa and L. A. N. Amaral. 2005. The Worldwide Air Transportation Network: Anomalous Centrality, community structure, and cities' global roles. *PNAS* 10, 22 (2005), 7794–7799.

S. Ma. S. M. Wilkinson, S. Dunn. 2012. The vulnerability of the European air traffic network to spatial hazards. *Natural Hazards* 60, 3 (2012), 1027–1036.

E. Paolo X. Hu. 2007. A Genetic Algorithm Based on Complex Networks Theory for the Management of Airline Route Networks. *Studies in Computational Intelligence* 129 (2007), 495–505.