EECS 4414 Final Project Report
Reliability of Road Network

Harsh Patel
York University
harsh710@my.yorku.ca

Alan Huang
York University
alan.rj.huang@gmail.com

Yixuan Ge
York University
yge95@my.yorku.ca

Xi Hu
York University
huxi@my.yorku.ca

Abstract

With increasing population, traffic congestion has become a common problem in modern society, especially in developing countries. Traffic congestion negatively impacts people's work efficiency and social development. According to a report, nine out of the twelve Canadian cities studied, experienced more traffic congestion in 2019 than they did in 2018 [2]. Conditions worsen at rush hour, drivers with a 30-minute commute each way will spend an extra 17 minutes in the morning and an extra 21 minutes in the afternoon on the road [2]. Thus, road construction and design is highly comprehensive and complex work.

Our goal is to propose a smart traffic application for road construction based on the city map networks. It can evaluate the city roads to determine if an alternate route exists during the traffic jam for emergency vehicles or if a better road infrastructure is needed for some existing road. Our application can help improve the quality of city roads. The application can be used by city planners or emergency vehicle drivers.

This paper includes discussion regarding details of the models, algorithms and actual results through our experiments, as well as analysis of our experiment and further work directions.

Keywords

reliability of road network, traffic congestion, road re-routing

1 Introduction

People rely on roads for almost everything, however, the most important use for them is for emergency situations. To save time, emergency vehicles need to arrive at their destination in the shortest time. A traffic congestion at such a time may result in people's deaths. Ambulance driver and paramedic Joseph Laylo says one patient died inside his ambulance after heavy traffic on a narrow road added 10 minutes to the journey from the patient's home to the hospital [4].

To evaluate the quality of a city's roads, people should not only focus on its role in traffic and city development, but also focus on the need for residents in its planning and design. We would like to analyze the factors that may affect the efficiency of the road. Test the road from multiple aspects such as reliability, robustness, and convenience to meet the requirements of safety, speed, and efficiency.

2 Problem Definition

We focus on analyzing the factors that may affect the road efficiency in this project. We mainly address the following problems through our project:

- Problem 1: Given a detailed map of a city, construct a road network G that can be used to analyze the road quality effectively.
- Problem 2: Given a road network, simulate some factors that may affect the road efficiency such as traffic congestion, the distance, and the number of roadblocks.
- Problem 3: Given a road network, simulate and compare the performance of each possible path between any two selected points, we will get the most efficient path as a result.

3 Related work

3.1 CHIMERA

CHIMERA (congestion avoidance through a traffic classification mechanism and a re-routing algorithm) is an intelligent traffic application based on vehicular networks proposed by de Souza et al[1] which is able to detect when traffic congestion occur on a certain section of the road, and uses a re-routing algorithm which ranks streets according to different congestion levels to effectively divert vehicles from the current congested road to other roads, so that it will not cause a larger and more serious congestion on the road. CHIMERA computes alternative routes by using the K-Shortest Path algorithm based on road weights, and uses the Boltzmann probability distribution to assign different routes to different vehicles, therefore, it is able to perform a load balance across alternative paths.

The advantage of this application is that it cannot only detect the current traffic congestion and then re-route vehicles to avoid the current congested area, but also can make more effective use of road resources and provide a long-term solution to rationally divert vehicles in congested areas such as metropolitan cities, so that it won't transfer the congestion problem from current road to another road closer to the congested one. This application is more efficient in forecasting congestion and performing a proper load balance of vehicular traffic. In our work, we employed this application to remove congested roads and to get some alternative routes after traffic congestion occurred to test our algorithm's effectiveness.

3.2 Critical Nodes

We also used information from a similar paper, titled "Reliability of San Francisco-Bay Area Road Network" [3]. The objective of this paper was to determine the general congestion of a road network, by observing how traffic changed when critical nodes were removed from a graph representation of the road network. The methodology of this paper was to model the road network as an undirected weighted graph, then to simulate traffic without critical nodes removed, and finally to then remove the most critical nodes, measure traffic with said critical nodes removed, and to observe the traffic increase of the new graph. Here, traffic was modelled via a game theoretic approach. Each car was treated as a rational actor with a goal of travelling from a source node to a destination node, where the cost of taking a specific route was modelled with various objective functions measuring different values such as least time, shortest distance, and least traffic. These objective functions were each used individually in separate tests, to compare different metrics of traffic. Finally, iterative simulations were done, where each car would take the most rational path based on the objective function, and this process would continue, taking note of the actions of other cars, until convergence occurred.

Our methodology has two main differences from this paper. Firstly, this paper evaluates traffic as a measure of all cars operating at once, but in the case of emergency vehicles, all vehicles must stop to let them pass. Secondly, this paper evaluates congestion by removing critical nodes, then measuring traffic as described above. In our paper, we assume that critical nodes are where the emergency buildings are located, as we reason that critical nodes are generally where major intersections occur, and emergency buildings tend to be located close to these major intersections.

4 Methodology

4.1 Dataset

For this study, we are investigating the reliability and robustness of a road network. Our team was able to find the city of San Joaquin County road network dataset created by the University Of Utah [5]. This dataset will be able to create a weighted graph. This road network was chosen as it showcases a good mix of complexity and availability of data. If time permits, later in the project some smaller road networks such as City of Oldenburg road network and larger road networks such as San Francisco-Bay Area road network may also be considered to test the algorithm's effectiveness on different road networks.

Size of San Joaquin Road Network		
Road Network	Nodes	Edges
San Joaquin	15263	21797

4.2 Network Construction

In order to test for robustness of the road network, the dataset was constructed into a graph using Networkx. In the constructed model network, nodes will represent the endpoints and intersections of roads. While edges represent the roads between intersections and between intersection and endpoint. The data set also contains the length of the edge between each pair of nodes. This means that we can construct an undirected weighted graph of the road network where each edge will be weighted according to the dataset.

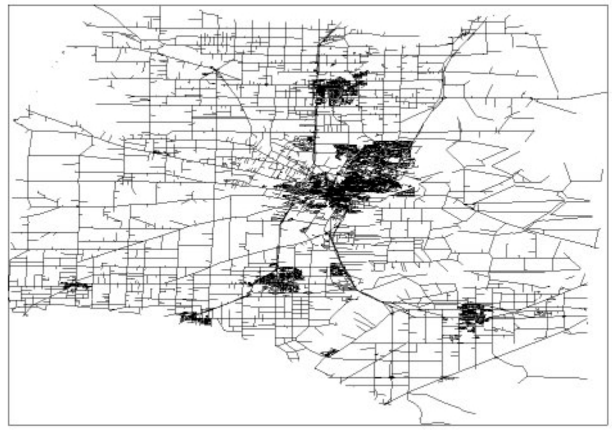

Figure 1: City of San Joaquin County Road Network

The Giant Connected Component of the road networks was calculated to eliminate small networks not connected to the main network. However, this turned out to be unnecessary as the dataset itself is strongly connected.

4.3 Data Collection

(1) Gathering of road dataset, which is discussed briefly in Section 4.1.
(2) Utilizing the dataset, the road network graph will be built. This process will be thoroughly covered in Section 4.2.
(3) After the network has been constructed, we can perform some simulations and run some tests. In section 4.3, three different types of tests will be computed on the network and their results will be evaluated in section 6.

5 Testing Strategies

As our project studies the reliability of road networks, we will examine the model's robustness by conducting three different types of tests. The methodology for each test is very similar to each other. The main difference among the tests is the parameters being used to conduct the experiment.

5.1 General Methodology

We will employ a general methodology which we will use to determine the overall road network reliability. The general methodology is as follows. Firstly, we decide on a start and end node which we will calculate the shortest path for. The shortest possible path between the start node and end node will be computed using Dijkstra's algorithm and the total weight of the path will be stored. The strategy used to choose the start and end node will be discussed later.

Once the shortest possible path has been computed, traffic congestion will be simulated into the graph. A number of roads on the computed path will be blocked to simulate a rush hour situation, and the choice of the roads to be blocked will be discussed later.

The edges on the blocked roads will be temporarily removed. After removing each edge, a new shortest path will be computed and stored every time an edge is removed. The new shortest paths will be compared to the initial shortest path and the percentage increase between them will be calculated. These percentage increases will show the reliability of the road network. The goal is to repeat the test on several different node pairs on the graph. We decided on an arbitrary sample count of 900 different node pairs. After all these nodes have been tested, nodes that have ratios lower than a certain ratio will be deemed to have unreliable roads.

5.2 Random to Random Node Testing Strategy :

The first and most basic strategy to determine the start and end node will be to randomly select these nodes from the road network graph, while also ensuring that we don't pick the same node for start node and end node. This strategy, while basic, is effective, as it gives us an unbiased view of the shortest paths in the graph.

5.3 Critical to Random Node Testing Strategy :

In a road network graph, critical nodes are usually major intersections, where a majority of traffic passes through making them prone to traffic jams in rush hours. Cities often have hospitals, fire stations and police stations located close to these intersections. These critical nodes need to be very well connected in the graph. By identifying critical nodes, we can better simulate an emergency vehicle's path in a real life emergency situation. Meanwhile, we would expect an emergency to start anywhere in a road network. Thus, the start node will be selected randomly, while the end node will be a critical node.

In order to conduct this testing strategy, critical nodes in the graph need to be identified. We used 2 different measures to identify critical nodes, betweeness centrality and maximum closeness centrality, this approach was inspired by "Reliability of San Francisco-Bay Area Road Network" [3] paper where they use a similar approach. More information on the paper can be found in Section 3.2.

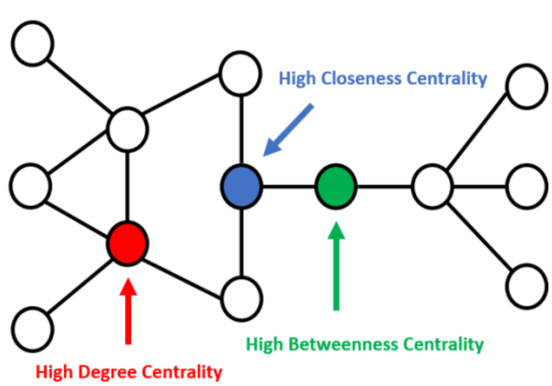

Figure 2: Betweenness and Closeness Centrality of Network

5.3.1 Betweenness Centrality :

Betweenness centrality measures to what extent a node stands between other nodes within the network. It can be measured as the percentage or number of shortest paths that pass through a node. In our case to get the most critical nodes, we calculated the betweeness centrality of all the nodes on the graph. In order to calculate centrality three steps need to be done.

(1) For each pair of vertices (s,t), compute the shortest paths between them. (Path_st)
(2) For each pair of vertices (s,t), determine how many shortest paths pass through the vertex v. (Path_st(v))
(3) Sum this over all pairs of vertices in the graph.

$$BC(v) = \sum \frac{Path\_st(v)}{Path\_st}$$

Algorithm 1: Betweenness Centrality Algorithms

```
Map (V,count) : Map with node id  count (default count = 0)
V : Array of all possible pairs of vertices in G (Graph)
Paths : 2D array with smallest possible paths (default empty)
foreach i in V do
    Paths.append(Shortest_Path_Possible(i.v1, i.v2))
end
foreach i in Paths do
    Paths.append(Shortest_Path_Possible(i.v1, i.v2))
    foreach v in i do
        if Map.has(v) then
            Map.put( v, map.get(v) + 1 )
        end
    end
end
Map.sort()
```

However, as with the paper which described the critical node measures, it was infeasible to compute the shortest path lengths for every single node pair possible. Thus, we instead opted to use only the shortest paths obtained by collecting all of the single source shortest paths on a sample of 1500 nodes. This functionality is provided by the NetworkX graph library.

5.3.2 Closeness Centrality :

Closeness centrality measures the shortest path from an individual towards all separate other individuals within the network. A high closeness centrality suggests that the node is very well connected in the network, and is easily reachable. Calculating the max value of closeness centrality for each node enables us to find how centralised it is in the graph. In order to calculate centrality two steps need to be done.

(1) Compute all the shortest possible paths from Node v to all other nodes in the graph.
(2) Next, compute the inverse of the maximum shortest path possible in step 1.

$$CC_{max}(v) = \frac{1}{max(min(distance(v,u))} \forall u \in G$$

Algorithm 2: Max Closeness Centrality Algorithm

```
CC_max : Array of max closeness centrality values (default:
  Empty Array)
all_nodes : list of all nodes in G
foreach v in all_nodes do
    foreach i in all_nodes do
        if v != i then
            distance.append(Shortest_Path_Possible ( v, i ))
        end
    end
    CC_max[v] = 1 / max(distance.values)
    distance = { }
end
```

An improvement over the paper which described the critical node measures was that we calculated the full maximum closeness centrality, as opposed to only choosing a sample of 20 nodes as the destination nodes when calculating all the shortest possible paths from node v. Not only is this a more accurate measure, it turns out that this proclaimed optimization does not work in practice, as computing the shortest path between two nodes is no faster than computing the shortest path between a node and every other node in Networkx at least.

Once critical nodes have been ranked and sorted, critical nodes with highest evaluations will be taken as parameters to conduct the tests. Next tests similar to (Random Node to Random Node Testing Strategy) will be conducted with one of the nodes being a critical node and other being a random node. At the end of this test, the shortest path possible before edge removals and shortest path possible after edge removals will be used to compute the robustness of the graph network where critical nodes are involved.

5.4 Edge Removal Strategies :

We have discussed above the general node selection strategy for the start and end nodes of the shortest paths to be calculated. Now, we will discuss the different strategies we will employ in choosing the edge to treat as a roadblock. Firstly, one condition on the edge removed is that it may not be the first or last edge of the shortest path, to alleviate the issue of the graph becoming disconnected. With that said, we have decided on three strategies, which we will discuss below.

5.4.1 Random Edge Removal Strategy :

The first strategy is to remove an edge at random in the shortest path. This strategy is effective, as we can never know for sure where a roadblock is most likely to occur. Therefore, picking an edge at random is a safe bet to simulate a real life circumstance.

5.4.2 Remove Edge by Weight Strategy :

The second strategy is to remove the edge with the greatest distance in the shortest path. Often times in real life, the most congested roads are the long, major roads. So it makes sense to consider the case where the longest roads become roadblocks, giving us more insight into the road stability of a network.

5.4.3 CHIMERA Testing Strategy :

Online map services such as Google Maps, Apple Maps etc, are able to collect real time traffic information from their users and use that information to reroute users to other less traffic congested routes. This is where CHIMERA rerouting system comes in. By using the traffic rerouting application of the CHIMERA system, we will be able to determine routes where traffic would likely be routed when a path is blocked due to traffic congestion.

The final strategy is to remove the most critical edge in the shortest path. We have conceived of a new, experimental metric called edge centrality, which is based on the degree centrality of nodes. Put simply, edge centrality is defined as the sum of the degree centrality of the edge's two nodes. Then, we define a critical edge out of a set of edges E to be the edge with the greatest edge centrality. Critical nodes in a real road networks are major intersections which connect many roads. The CHIMERA [1] system is used to develop an algorithm which will effectively determine edges where traffic will be potentially rerouted in the event of an traffic congestion.

Algorithm 3: CHIMERA algorithm

```
vehicles : Set of the vehicles in the network
all_edges : List of all edges in G
foreach v in vehicles do
    highest_centrality = 0
    final_edge = null
    foreach edge in all_edges do
        edge.centrality = sum of degree_centrality of two
         nodes
        if edge.centrality > highest_centrality then
            highest_centrality = edge.centrality
            final_edge = edge
        end
    end
    edges.remove(final_edge)
    reroute(v)
end
```

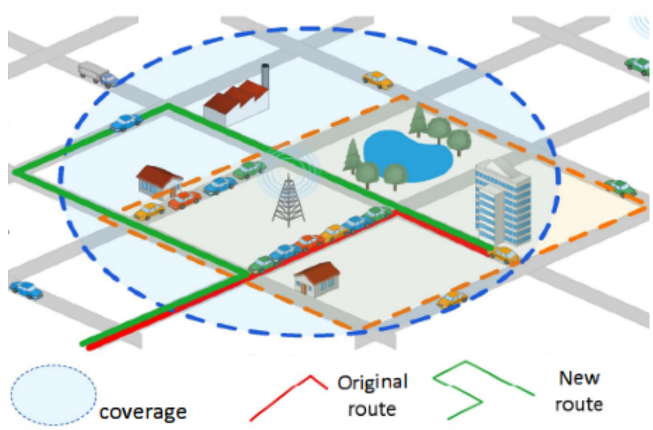

Figure 3: CHIMERA Rerouting System

6 Setbacks

Some additional setbacks were encountered in the implementation of the methodology, due to unforeseen implementation details.

The first issue was that by sheer chance, it was possible that the generated start and end nodes created a shortest path which was too short in node length to remove any edges from. This occurs if the shortest path has three or less nodes in it. The solution formulated for this problem was to ensure the initial generated shortest path was greater than a certain node length (in our case, we chose 7).

The second issue was that it was possible that forming a path between the start and end node would be impossible, due to the creation of a disconnect in the graph while adding in roadblocks. This issue was completely unexpected during the initial methodology. The initial solution was to fill the remaining shortest path lengths with a punishment value, decided to be thrice the length of the initial shortest path length, the idea being that the possibility of a graph being completely disconnected is a severe indication, yet we don't want to consider undue weight to one single disconnection out of a sample of 900 trials. However, we decided in the end to exclude instances when forming a path is impossible in the resulting score calculation, but keep a count of when such an event occurs.

7 Evaluation

7.1 Generating initial statistics

Firstly, before actually performing the methodology and evaluating the results, we decided to run some basic experiments in order to generate some initial statistics on our dataset, giving us insight into our methodology. As stated earlier, our current plan for our methodology is to collect the shortest path lengths for multiple start and end node pairs, then to calculate the shortest path lengths again except by removing edges from the shortest paths. We decided on two strategies to determine the start and end nodes used. The first strategy was to pick random start and end nodes, and the second strategy was to pick a critical node as the start node, and a random node as the end node. Thus, the initial statistics we decided to generate were based around observing the paths generated with our two node picking strategies. The experiment for random to random node picking is described below:

Algorithm 4: Preliminary evaluation statistics pseudocode

```
for 900 iterations do
    generate the start and end node, based on the node
      picking strategy.
    compute the path and path length of the resulting
      shortest path.
    append the current path length to a list containing the
      path lengths for all current shortest paths.
    append the current path node length to a list containing
      the path node lengths for all current shortest paths.
    keep track of two current longest paths, measured in
      terms of path distance and number of nodes in the path
      respectively
end
Calculate the average path length and average path node
  length out of all shortest paths generated, using the two
  lists mentioned above
```

Note: When generating the start and end nodes for critical to random nodes, because critical nodes are not a binary measure, the plan was to select critical nodes by alternating between each metric for critical nodes, and choosing the node with the greatest value.

The results were as follows (rounded to 4 decimal places):

	Maximum Path Length	Average Path Length	Maximum Path Node Length	Average Path Node Length
Random to Random	11294.1221	3678.1247	301	106.4222
Critical to Random	9258.1778	2519.6887	263	85.0878
% increase from C2R to R2R	21.99%	45.98%	14.45%	25.07%

From these results, we can conclude that random to random node selection results in longer paths. This makes sense, as critical nodes are meant to be a general measure of nodes which are most connected to other nodes. In regards to the problem at hand, these statistics suggest that we have to be cautious when comparing percentage increases in path lengths between random to random and critical to random paths. For example, if we assume that invoking a detour for random to random and critical to random shortest paths results in the same constant increase in distance, the percentage increase between the original and new random to random path would be less than the percentage increase for critical to random! However, this assumption might not hold, and it depends on the average makeup of the shortest path, e.g. the average edge length in the shortest path, whether the choice of critical vs random influences the structure of the shortest path. Thus, further investigation may be necessary.

7.2 Road Reliability Scoring Strategies

In our original proposal, we had proposed a single scoring strategy to determine road reliability. This scoring method was as follows. Firstly, we would perform the described methodology for a number of chosen start and end node pairs in order to collect the path lengths for a number of edge removals on the shortest path, an edge removal representing an induced roadblock. For each start and end node pair, we will generate a score for each shortest path with edge removal. This score will be based on the percentage increase between the original shortest path distance and the path with edges removed, obtained by dividing the former distance with the latter distance, for each number of roadblocks. Then, we take a weighted average of these scores, weighted by the likelihood of increasing roadblocks happening at once. All these weighted averages will then be combined into one final score, which represents the road unreliability.

However, in light of discovered issues with the initial scoring strategy, we have opted to use two different scoring strategy. The first scoring strategy is the original strategy described above, taking the average percentage increase between the original shortest path length and the shortest path length after roadblocks are applied, except with a slight modification. Namely, we decided against combining the average percentage increases for each roadblock count into one single score, as deciding how much to weigh each average based on the roadblock count seemed too arbitrary. In other words, we couldn't really say how much more important the average distance increase for paths with one roadblock were compared to paths with multiple roadblocks. While this scoring system has intuitive results, it also has the disadvantage that longer paths will inherently score as less congested, as generally speaking, for a road network, increasing the shortest path length does not result in a proportional increase of individual edges.

During our testing we realised that when some edges are removed they can cause the strongly connected road network to become disjoint, leaving no way another shortest path to reach the target node. The third scoring strategy was devised to highlight this. This metric counts the number of shortest paths that could not reach target node after an edge has been removed. This metric shows how much the road network relies on few critical roads with no alternative.

Throughout the implementation, numerous additional scoring strategies were thought of in light of discovered issues with the current strategies. One of these suggested strategies were to measure statistics based on shortest path distance ranges, that is, we would calculate scores for shortest paths in the range of 0-500, 500-1000, 1000-1500 etc. Another suggested strategy was to continue adding roadblocks until the distance of the roadblocks added exceeded a percentage of the original shortest path length. However, we did not pursue implementing these in the interest of time, and also because based on past experience we would be stuck in a loop of finding flaws for these proposed strategies, and subsequently devising new strategies which resolve the flaws of the old proposed strategies.

7.2.1 Shortest Path length Increase

For our analysis we stimulated road blocks by removing the edges as per the edge removal strategy. We decided to create 4 roadblocks and collected data after each simulated roadblock. We wanted to analyze the increase in shortest path length when a roadblock event occurs. This increase can give an insight into how reliable and robust the road network is. Note: All the metrics are an average of 900 trials.

Random Node to Random Node

Shortest Path Length Increase				
Edge Removal Strategy	1st Road Block	2nd Road Block	3rd Road Block	4th Road Block
Random Edge Removal	2.66%	4.43%	6.71%	8.19%
Weight Edge Removal	3.86%	7.76%	11.87%	18.07%
CHIMERA System	1.14%	2.34%	3.51%	4.9%

Critical Node to Random Node

Shortest Path Length Increase				
Edge Removal Strategy	1st Road Block	2nd Road Block	3rd Road Block	4th Road Block
Random Edge Removal	2.11%	3.98%	6.04%	7.97%
Weight Edge Removal	4.06%	7.95%	12.4%	18.5%
CHIMERA System	0.65%	1.73%	2.69%	3.48%

As per the tables it seems like on average the shortest possible path length increase is greater in random to random node strategy than in critical to random node strategy.

Critical nodes to random node strategy has a lower shortest path increase because the selected critical nodes have a high betweenness centrality and closeness centrality making them very well connected. On the other hand random node to random node strategy selects nodes randomly, ensuring that nodes other than well connected ones are also selected.

The highest increase is seen when longest edges (edges with max weight on path) are removed, this is because many cities have few long and important roads that are used to quickly traverse across the city. When these roads are blocked, traffic is forced to traverse through city roads causing a major increase in travel.

The CHIMERA traffic rerouting strategy shows that it is effective when rerouting traffic. As CHIMERA reroutes traffic based on edge centrality, CHIMERA is able to route traffic efficiently, and in a way that causes the least distance increase.

A thing to note is that, on average the path length of random to random node is 45% higher than critical to random node (Section 7.1). And as mentioned in Section 7.2, longer paths inherently score as less congested.

7.2.2 Disjoint Caused Due To Edge Removal

Initially the whole road network is one giant connected component, where each node can reach any other node on the network. We noticed that for some shortest possible paths when roadblocks are introduced, they lose the ability to reach the target node. This means that if a certain edge was removed from on the graph, it caused the graph to become disjoint. This is important because it shows how vulnerable the road network is when traffic congestion occurs.

For this test we calculated how many instances (out of 900 trials) of the road network became disjoint when 4 and 10 roadblocks were introduced.

Random Node to Random Node

Disjoint Rate at 4 & 10 Roadblocks		
Edge Removal Strategy	4 Roadblocks	10 Roadblocks
Random Edge Removal	2.67%	5.33%
Weight Edge Removal	4.22%	12.1%
CHIMERA System	1.22%	3.89%

Critical Node to Random Node

Disjoint Rate at 4 & 10 Roadblocks		
Edge Removal Strategy	4 Roadblocks	10 Roadblocks
Random Edge Removal	1.44%	5.22%
Weight Edge Removal	3.00%	7.89%
CHIMERA System	0.55%	2.78%

It can be observed that random node to random node has a higher disjoint rate than critical to random node. This is because critical nodes are more well connected than random nodes, we see a lower disjoint rate.

7.2.3 San Joaquin Road Network Reliability

To conclude, overall the road network of City of San Joaquin road network seems robust, however it relies a lot on long roads. The city needs to look into developing new road infrastructure around long roads in the likely event of traffic congestion on these long roads.

7.3 Comparing Different Road Networks

Another thing we plan to do is to identify certain weak points of the network, near critical nodes where removal of one or more edges results in a much greater shortest path distance compared to the shortest path distance without any removed edges.

7.3.1 Oldenburg vs San Joaquin Road Network:

In order to test the effectiveness of our proposal we decided to test it on a second database. We decided to test it on City of Oldenburg roads dataset [5]. Unlike City of San Joaquin dataset, this dataset does not contain a downtown area, i.e. it is devoid of dense area where many small roads intersect. And does not seem to be planned city (city with straight roads and planned blocks) like San Joaquin . This can give us an insight into how our proposed method evaluates such an area.

Size of Road Networks		
Road Network	Nodes	Edges
San Joaquin	15263	21797
Oldenburg	11305	13029

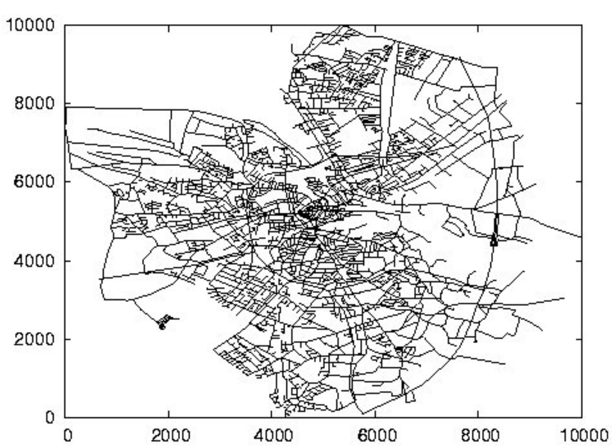

Figure 4: City of Oldenburg Road Network

7.3.2 Shortest Path length Increase

We decided to compare Shortest Path length Increase data when 4 roadblocks are introduced to each city.

Random Node to Random Node

Shortest Path Length Increase		
Edge Removal Strategy	San Joaquin - 4th Road Block	Oldenburg - 4th Road Block
Random Edge Removal	8.19%	19.35%
Weight Edge Removal	18.07%	25.98%
CHIMERA System	4.9%	17.23%

Critical Node to Random Node

Shortest Path Length Increase		
Edge Removal Strategy	San Joaquin - 4th Road Block	Oldenburg - 4th Road Block
Random Edge Removal	7.97%	20.3%
Weight Edge Removal	18.5%	23.1%
CHIMERA System	3.48%	13.9%

It can be observed that random node to random node has a higher shortest path increase than critical to random node but not by much in Oldenburg. This is because Oldenburg has many long roads and roads have not been carefully planned. Making critical nodes have a lesser score than critical nodes in San Joaquin Network.

From the data in the tables it can be seen that traffic congestion in the city of Oldenburg can cause a more serious increase in distance to travel than the city of San Joaquin . This illustrating how important city planning is.

7.3.3 Disjoint Caused Due To Edge Removal

We decided to compare Disjoint Rate data when 4 roadblocks are introduced to each city.

Random Node to Random Node

Disjoint Rate at 4 Roadblocks		
Edge Removal Strategy	San Joaquin %	Oldenburg %
Random Edge Removal	2.67%	8.56%
Weight Edge Removal	4.22%	8.89%
CHIMERA System	1.22%	5.00%

Critical Node to Random Node

Disjoint Rate at 4 Roadblocks		
Edge Removal Strategy	San Joaquin %	Oldenburg %
Random Edge Removal	1.44%	5.11%
Weight Edge Removal	3.00%	5.78%
CHIMERA System	0.55%	3.33%

It can be observed that random node to random node has a higher disjoint rate than critical to random node. This is because critical nodes are more well connected than random nodes, we see a lower disjoint rate. However, compared to San Joaquin , the disjoint rate of Oldenburg is at least 3 times higher. Due to Oldenburg's poor city planning a minor traffic congestion can stop people from being able to get across the city, halting all the traffic.

In conclusion, cities created with appropriate city planning considerations tend to be more robust than cities which create roads based on need. Poorly planned cities are also prone to having greater traffic delays.

7.4 Future Work and Challenges

Now that we have identified the weak links in the road network, a future plan is to create an algorithm which creates new edges around the weak node, to a target node. These new edges can be used to direct traffic over the weak link (a flyover). So that traffic that needs to get across the city can take the new path. This is a common technique seen in many developing countries.

8 Conclusion

In our work, we have designed a program to find the shortest route between two positions and use it to investigate the reliability and robustness of a road network. In our simulation of city transportation networks, we use edges and nodes to simulate roads and intersections, and we find the shortest possible path between two nodes as per our methodology testing strategies. After finding the shortest possible path, we temporarily remove the road with the highest congestion level on the selected path, and recompute the shortest possible path to get a new route. We use three different types of method to achieve edge removal: removing a random edge, removing the longest edge, and removing the most critical edge using CHIMERA[1], then we simulate the scenario of avoiding traffic congestion by temporarily removing this edge from the dataset and recomputing the shortest possible paths to get new optimal routes. In the end, shortest possible paths before removal and after removal are compared and robustness of the road network is calculated. Therefore, the goal of calculating the robustness and reliability of the road network in case of an emergency situation can be obtained.

We test our algorithm by using two different road reliability scoring strategies, and we not only evaluate our algorithm using the city of San Joaquin County road network dataset, but also use the city of Oldenburg road network which is a smaller road network dataset. By comparing road networks of two cities, we can see that cities with appropriate city planning considerations are more robust.

In the future, we may test our algorithm's effectiveness using some road networks in Canada closer to us such as the Toronto road network, or some larger and more complex road networks such as the China road network, so that our work can be applied to real life and has more practical significance.

GITHUB REPOSITORY

https://github.com/Harsh-B-Patel/Road_networks

References

[1] G. Maia A. Loureiro A. M. de Souza, R. S. Yokoyama and L. Villas. 2016. Real-time path planning to prevent traffic jam through an intelligent transportation system. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. 726–731. https://doi.org/10.1109/ISCC.2016.7543822

[2] Ryan Flanagan. 2020. These are the Canadian cities where drivers spend the most time stuck in traffic. (2020), 1–2. https://www.ctvnews.ca/canada/these-are-the-canadian-cities-where-drivers-spend-the-most-time-stuck-in-traffic-1.4790142

[3] Chung Yu Wang Hsin-Fang Wu and Alice Yeh. 2014. Reliability of San Francisco-Bay Area Road Network.

[4] Agency Report. 2019. Patients die as ambulances get stuck in traffic jam. (2019), 1–2. https://punchng.com/patients-die-as-ambulances-get-stuck-in-traffic-jam/

[5] @University Of Utah. 2005. *Real Datasets for Spatial Databases: Road Networks and Points of Interest*. Retrieved August 4, 2021 from https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm