Title
A Study on Identifying Road Network Vulnerability

Author
Lim, Jonghee

Publication Date
2021

Peer reviewed|Thesis/dissertation

A Study on Identifying Road Network Vulnerability

By

JONGHEE LIM

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____

Prof. Dipak Ghosal, Chair

____

Prof. Chen-Nee Chuah

____

Prof. Raissa M. D’Souza

Committee in Charge

2021

TABLE OF CONTENTS

ABSTRACT

The use of GPS navigation apps has been surging with increasing the use of smartphones. People used to use a map to find a way to their destination, but now the navigation apps provide us with several options that can be taken as the best path for drivers. Also Connected and Autonomous Vehicles (CAVs) have been attracting attention and have improved a lot in tandem with the popularization of electric vehicles. Also, it is expected that autonomous driving which means that the car is self-driven instead of being driven by a person will emerge in the near future. This autonomous driving will rely on algorithms that will determine the best path once the destination is set and will perform dynamic routing to adapt to road network congestion during the trip. These algorithms that suggests the best paths will affect road traffic flow. Furthermore, when more CAVs rely on the same best path algorithm in the future, it will lead to a significant impact on a city's road network traffic if an adversary can intentionally cause real or virtual disruptions in an few strategic locations.

This thesis presents the results of simulation experiments to determine vulnerability in road network in the city of Davis in California. Specifically, by developing a simulation model, this study models how disruption in one part of the network can spread and how this is related to or not related to some of the topological properties of the network. Lastly, this thesis discusses a way to minimize the effect of the vulnerability.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my committee chair, Professor Dipak Ghosal, for invaluable guidance and support through the course of study. I am so fortunate to be advised by him. I would not have successfully completed the study without his help.

I would also like to thank committee member Professor Chen-Nee Chuah who introduced me to CP3O and provided valuable suggestions during the meeting. I would not have met such valuable people without her help.

Thanks especially also to Professor Michael Zhang, who was willing to engage with my research, providing wonderful advice for me to strengthen my knowledge. I would like to express my special appreciation to Professor Raissa M. D'Souza, for her guidance with network science as applied to road networks.

Lastly, I would like to thank my family and friends who always give me infinite support and emotional relief during tough times.

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The use of GPS navigation has been soaring since smartphones have become an essential tool for people. Today people do not need a printed maps like before because it is more convenient for them to make use of mobile apps like Google Maps[1], Waze[2], and MapQuest[3]. People tend to use GPS apps even though they already know the way to their destination. This is because these apps not only offer the best paths for their destination, but also provide information about traffic congestion and accidents that occurred on the way to their destination. Moreover, these apps present spots where speed cameras are located so that drivers can be aware of them. The more people utilize these GPS apps, the higher influence they have on road traffic. In fact, one can envision that rerouting recommendation provided by the navigation programs following congestion in one part of the road network can lead to congestion in other parts of the network.

Not only that, Tesla, an automotive company, has made huge progress in electrical vehicle and self-driving technology. There are six Levels of automated driving from zero to five as defined by the Society of Automotive Engineers [4]. In Level Zero there is no automation and the driver is totally in charge of managing the vehicle, performing tasks such as accelerating, steering, and braking. On the other hand, Level Five capable vehicles are fully autonomous and driver is not needed behind the wheel. One of the projects to achieve Level Four took place in the U.K. last year. The cars operated on a nine-mile circuit through a city's main train stations [5]. Furthermore, Tesla's Full Self-Driving technology is expected to reach Level Five autonomy by the end of 2021. Whether they could reach Level Five autonomous cars by the of 2021 or not, it seems this goal will be reached in the near future. All the functions of vehicle in Level Four or Five will be fully automated by relying on road-side sensors and traffic control management system. In the future, these connected autonomous vehicles (CAVs) and the traffic control system will be fully integrated with a ultra-reliable low-latency wireless communication channel defined in the 5G wireless network standard [6]. The traffic control management system consists of traffic signal controllers

(TSC) which schedule traffic at the intersections and provide speed guidance to vehicles will be significantly enhanced by this integrated system.

This self-driving environment can provide drivers with many benefits such as stress-free parking, fewer traffic jams, and the decreased probability of car accidents. However, new safety, security, and hacking concerns will also arise. The more CAVs are introduced, there are new vulnerabilities that can be exploited by adversaries.

There are several ways by which an adversary can impact the traffic flow in the road network. This includes attacking vehicles directly, attacking the specified traffic signal controllers, or attacking the traffic control management system. The purpose of the adversary is to create significant disruption in the road traffic network that can lead to congestion and significantly increase the travel time. In this study we consider an adversary that attacks the TSC or traffic control management system instead of targeting individual vehicles to affect the traffic flow in the road network.

1.2 Methods to Cause Road Network Disruptions

This study starts with the assumption that an adversary can launch an attack on a TSC and cause it to take wrong scheduling decision thereby disrupting the flow of traffic through the intersection. Given this capability, the key question is to determine the adversary attack strategy that can create network wide disturbance and significantly increase the average travel time. This study was motivated by a previous study that showed that in the power grid network disruption in a few specific link can cause cascading failures resulting in large network failure [7].

Blocking roads sections which are critical in the network are likely to lead to an impact on the whole network. Hence the disrupting a small number of road sections can result in a disturbance for whole areas. One of the goals of this thesis is to demonstrate the result of attacking some traffic signal controls, how the disruptions spread through the network, and what is the impact of the spread on the travel time of vehicles. Another goal is to show how various traffic patterns can be predicted, and what is a way to minimize the effect from the disruption.

1.3 Objective and Contributions

This thesis seeks to contribute to network science of road networks by studying flow patterns that resulted after blocking some specific edges. Given the map of the road network, the study sought to determine the location of the critical roads where injecting disruption could impact the whole network. This study is based on a simple network topology metric in which the edges that are used the most are the important edges.

In addition, the study provides a detailed simulation analysis of the traffic flow variation. Specifically, the simulation is used to determine the alternative roads that will be exploited when a heavily used edge is blocked and determine the new locations of congestion in the road network.

The ideas explored in this study could apply to other networks such as communication networks, power gird networks, and other transportation networks when they injected with accidental or intentional failures.

1.4 Approach

First, we determine which roads are most used by simulating (traffic)- with a certain number of sources and destinations to find the best path for them. Next, we block the most used road, and simulate to determine new road sections that are the most used. We then block the most used road from the second simulation, and perform the next simulation. By reiterating this several times, we determine how the traffic patterns for a given road network changed and which roads will be used for the main routes.

1.5 Thesis Overview

Chapter 2 covers the background knowledge to understand what algorithms and methods such as centrality of network, Monte Carlo integration, and Dijkstra algorithm were used in the experiments. The simulation programs and the fundamental settings of the studies are covered in Chapter 3. This chapter also discusses the many procedures involved in conducting experiments to attain the objectives. Chapter 4 discusses the results of experiments and Chapter 5 finalizes the thesis with the conclusion.

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

In this chapter we discuss centrality measures that are used to characterize complex networks. We also review the shortest path routing algorithm that is used in the simulation analysis. Finally, we discuss the Monte Carlo method that was used to determine the number of source and destination pairs for the simulation experiments.

2.2 CENTRALITY MEASURES

Centrality measures [8,9,10,11] are an essential property for understanding complex network structure. These measures based on graph theory are used to determine the importance of nodes and edges in a network. The centrality of nodes identifies which nodes are more "central" than others. Freeman (1978) [12] suggested that central nodes were focal points. He presented his idea with an example that showed a simple network consisting of 5 nodes (see Fig 2.1). In the graph, the center node (node A) has three distinguishing features compared to the other nodes. First, it has the maximum connection to other nodes. Second, it has the closest distance (in number of hop counts) to all the other nodes. Third, flows between any pair of nodes pass through it. Freeman (1978) defined these three conceptual foundations for network centrality: degree, closeness, and betweenness.

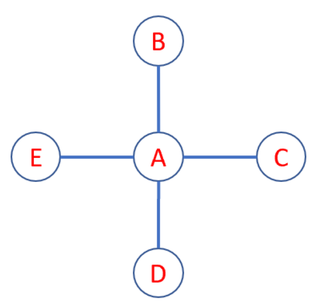


Figure 2.1 A simple network consisting of 5 nodes and 4 edges. Node A is the central node with respect to degree, closeness and betweenness centrality of the network.

Degree of a node is the number of nodes that the node is directly connected to. Therefore, degree is easy to calculate once the adjacent nodes around a node are provided. The central node with respect to degree centrality is the node with the highest degree. However, degree centrality has limitations as it does not characterize the global structure of the network. Closeness was introduced to enhance the feature and is defined as the inverse sum of shortest distance to all other nodes from a node. The node with the largest closeness value is the central node with respect to closeness centrality of the network. A major drawback of closeness centrality is that closeness cannot apply to a network with disconnected network. Betweenness evaluates the extent to which a node lies on a geodesic path between two other nodes. This measure can be applied to networks with disconnected component with some limitations.

2.2.1 Degree Centrality

Degree centrality is the simplest measure, which is defined by the number of links held by each node. This means how many direct, 'one hop' connections each node has to other nodes in the network. Degree centrality assigns an importance score based simply on the number of connections attached to each node. A node that has more ties to other nodes is regarded to be in an advantaged position with a higher score. Because the node has many ties, that node has easy access to other nodes, also the node can be a deal maker in exchanges among the others. This method is useful for finding popular nodes, very associated nodes, nodes that can quickly connect with the broad network, or nodes that are likely to maintain the most information. For directed graphs it is important to differentiate in-degree and out-degree, i.e., consider the number of inbound links and the number of outbound links as distinct measures. However, degree centrality has some limitations in directed graphs as well.

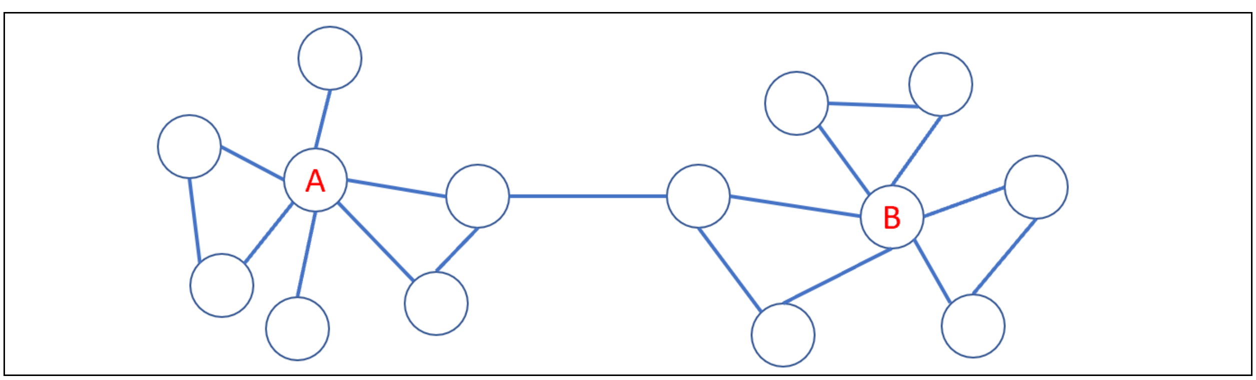


Figure 2.2 A network illustrating two local center nodes A and B. Both A and B have high degrees but their importance is local.

For instance, in Figure 2.2, node A and node B both have node degree of six but they are not central as they are at the periphery of the network. Thus, the degree centrality may not determine the importance of the node in the network. The degree centrality reflects importance of the node in the local surrounding.

2.2.2 Closeness Centrality

Closeness centrality is defined in terms of the inverse of farness, which is the sum of distance of a node to all others. Closeness centrality scores each node based on its 'closeness' to all other nodes in the network. This measure assesses the shortest paths between all nodes, then allocates each node a score which is the inverse of the sum of the shortest paths. Closeness considers all the nodes in the network even as some of the nodes are quite distant. On the other hand, degree centrality just examines adjacent nodes which are directly connected to the focal node. The closeness centrality measure is useful for finding the individual nodes which are best placed to affect the whole network most rapidly. However, the closeness centrality also has some limitations. First, the range of variation is too restricted since closeness centrality is only based on the shortest distance. Results show that all nodes have a similar score [13]. Moreover, it cannot be guaranteed that nodes with the same closeness have the same role in the network. Second, it does not work well in networks that have disconnected sub-networks. When it comes to calculating two nodes that belong to different networks, the closeness centrality scores for those nodes will be infinite. Therefore, closeness centrality is normally limited to connected networks which means that from a given node all the other nodes in the network are reachable.

Closeness centrality is applicable to find good influencers, but within a single cluster, it will often get several nodes with the same level of closeness centrality scores.

2.2.3 Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest path between other nodes. This measure also indicates which nodes work as bridges between nodes in a network. The nodes with high betweenness centrality are more likely to have significant influence on other nodes within a network since the nodes are capable of controlling information flow between other nodes. They are also regarded as the ones whose removal from the network will lead to considerable disruption to the whole network since they lie on the largest number of shortest paths taken by other nodes.

Formally, the betweenness centrality (BC) of a vertex (node) v is defined as:

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

where σ_{st} indicates the number of shortest paths from s to t, and can be viewed as the number of information pathways and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through v.

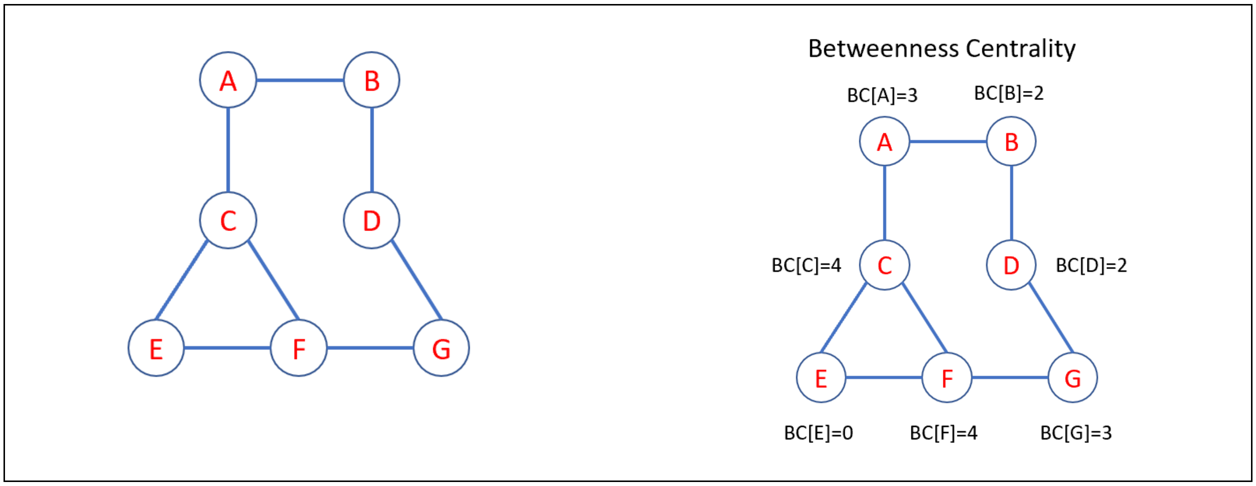


Figure 2.3 Example illustrating the betweenness centrality of each node. The shortest path is in terms of the number of hops.

For instance, to calculate Betweenness centrality for node A, it is necessary to find the number of shortest paths between vertices. First, it is necessary to find the total number of pairs of vertices not including node A. There are fifteen such pairs such as BC and FG. For a given pair there maybe more than one shortest path. For example, in Figure 2.3 between nodes B and F there are two different shortest paths - one is B-A-C-F and the other is B-D-G-F. On the other hand, between nodes A and E there is only one shortest path which is A-C-E. Next we need to determine the number of shortest paths that include (pass through) node A. For example, node A is on the shortest paths for the pairs BC, BE, BF, and CD. The number of shortest paths for BC and BE is only one while that of shortest paths for BF and CD is two. Therefore, betweenness centrality for node A is three.

For A	BC	BD	BE	BF	BG	CD	CE	CF	CG	DE	DF	DG	EF	EG	FG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	1/1	0/1	1/1	1/2	0/1	1/2	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	
BC(A)	1	0	1	0.5	0	0.5	0	0	0	0	0	0	0	0	0	3
For B	AC	AD	AE	AF	AG	CD	CE	CF	CG	DE	DF	DG	EF	EG	FG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	1/1	0/1	0/1	1/2	1/2	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	
BC(B)	0	1	0	0	0.5	0.5	0	0	0	0	0	0	0	0	0	2
For C	AB	AD	AE	AF	AG	BD	BE	BF	BG	DE	DF	DG	EF	EG	FG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	0/1	1/1	1/1	1/2	0/1	1/1	1/2	0/1	0/1	0/1	0/1	0/1	0/1	0/1	
BC(C)	0	0	1	1	0.5	0	1	0.5	0	0	0	0	0	0	0	4
For D	AB	AC	AE	AF	AG	BC	BE	BF	BG	CE	CF	CG	EF	EG	FG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	0/1	0/1	0/1	1/2	0/1	0/1	1/2	1/1	0/1	0/1	0/1	0/1	0/1	0/1	
BC(D)	0	0	0	0	0.5	0	0	0.5	1	0	0	0	0	0	0	2
For E	AB	AC	AD	AF	AG	BC	BD	BF	BG	CD	CF	CG	DF	DG	FG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	0/1	0/1	0/1	0/2	0/1	0/1	0/2	0/1	0/2	0/1	0/1	0/1	0/1	0/1	
BC(E)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
For F	AB	AC	AD	AE	AG	BC	BD	BE	BG	CD	CE	CG	DE	DG	EG	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	0/1	0/1	0/1	1/2	0/1	0/1	0/1	0/1	1/2	0/1	1/1	1/1	0/1	1/1	
BC(F)	0	0	0	0	0.5	0	0	0	0	0.5	0	1	1	0	1	4
For G	AB	AC	AD	AE	AF	BC	BD	BE	BF	CD	CE	CF	DE	DF	EF	TOTAL
$\frac{\sigma_{st}(v)}{\sigma_{st}}$	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	1/2	1/2	0/1	0/1	1/1	1/1	0/1	
BC(G)	0	0	0	0	0	0	0	0	0.5	0.5	0	0	1	1	0	3

Table 2.1 Calculation for the betweenness centrality measure for all the nodes in Figure 2.3.

Table 2.1 shows the betweenness centrality for all nodes in Figure 2.3. Node C and F have a high BC score since they lie on important paths for other pairs, while Node E has a zero score since node E is not on the shortest path between any of the pairs of vertices. Nodes which most frequently lie on these shortest paths will have a higher betweenness centrality score.

It is important to note that the betweenness centrality values depend on the size of network. To compare the Betweenness centrality values of two different networks, a normalizing method is usually adopted in which the BC scores are normalized by their largest possible value. It is useful for comparing networks that changes over time.

Betweenness centrality is useful for analyzing social networks, transportation, and scientific cooperation by finding individuals who influence network flow. A high betweenness value could indicate a critical position with control over disparate clusters in a network, or just that they are on the periphery of both clusters. This measure acts as a significant indicator for analyzing networks in this thesis, since it will be used to compare experiment simulations.

2.3 Routing Algorithm

2.3.1 Dijkstra Algorithm

Dijkstra's Algorithm [14,15,16] determines the shortest path between one node and all other nodes in a graph by determining the shortest pass tree which has a set of nodes that have the shortest distance from the first node. The concept of the algorithm is to repeatedly calculate the shortest distance from a source node to the end nodes while it excludes longer distances when making an update.

Algorithm 1: Dijkstra's algorithm

Input: Graph $G = (V, E)$

$(\forall x \neq s) dist[x] = +\infty$ //Initialize dist[]

$dist[s] = 0$

$S = \emptyset$

$Q = V$ // Keyed by $dist[]$.

while $Q \neq \emptyset$ **do**

    $u = extract\_min(Q)$

    $S = S \cup \{u\}$

    foreach *vertex* $v \in Adj(u)$ **do**

        $dist[v] = min(dist[v], dist[u] + w(u, v))$

        //"*Relax*" operation.

Figure 2.4 Dijkstra's algorithm.

The Dijkstra algorithm consists of the following steps:

1. Select the start node with distance value zero, and then initialize of all other nodes with distance 'infinite'.
2. Mark the starting node as visited and the distance of the starting node is permanent which is 0, and all other distances can be changeable except for visited nodes.
3. Calculate the distances of all neighbor (adjacent) nodes of the just visited node by summing up its distance with the weights of the edges.
4. If a new calculated distance of a node is smaller than the current value, update the distance and set the current node as previous node. Now these neighbor nodes which have a new calculated distance and previous node information are added to a priority queue. The key for the priority queue is distance.
5. Next, pop a node from Q with the shortest distance and mark the node as visited.
6. Repeat 3 to 5 until no nodes are left from Q.

The main idea of the algorithm is to keep track of the shortest distance from each node to the start node and to update these distance values in case a shorter path is found. The Dijkstra algorithm is efficient enough to find the best paths. However, it has some limitations like doing a blind search, wasting unnecessary time and resources. Also, it cannot work with graphs that have negative weights, so sometimes it reaches the wrong paths. Nevertheless, Dijkstra provides the shortest path between one node of the graph and all other nodes, so it is widely used in various fields. In this study, we have used Dijkstra's to find the shortest path between source and destination in the road network simulation.

2.4 MONTE CARLO INTEGRATION

Monte Carlo integration [17,18] is a numerical method to approximate the integral of a function using random sampling. This method tends to be straightforward, expandable, and flexible. In general, Monte Carlo techniques allow complex models to be encoded through a set of rules that can be performed on a computer. Monte Carlo method is a practical way for the multi-dimensional integration problems that widely occurs in computer graphics.

The principle of a basic Monte Carlo estimation is to use random sampling of a function to approximate the value of an arbitrary integral. Suppose we want to integrate a one-dimensional function f(x) from a to b such as given by

$$F = \int_a^b f(x)dx$$

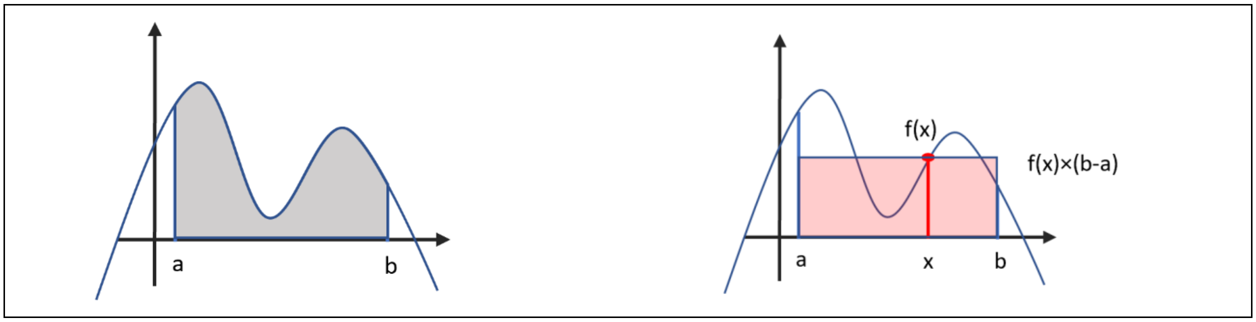


Figure 2.5 Figure illustrating the basic idea of Monte Carlo integration to find the area under the curve f(x) in the range a to b.

Figure 2.5 shows an arbitrary function f(x) for which we need to calculate the area in the range from a to b, i.e., the shaded area in graph on the left. If any random point x is selected in the range between a and b, the area of the rectangle of width (b-a) and height f(x) can be multiplied by the value of (b-a), the length between a and b. The figure is the same as the rectangular area where the function value is height and length (b-a) is width.

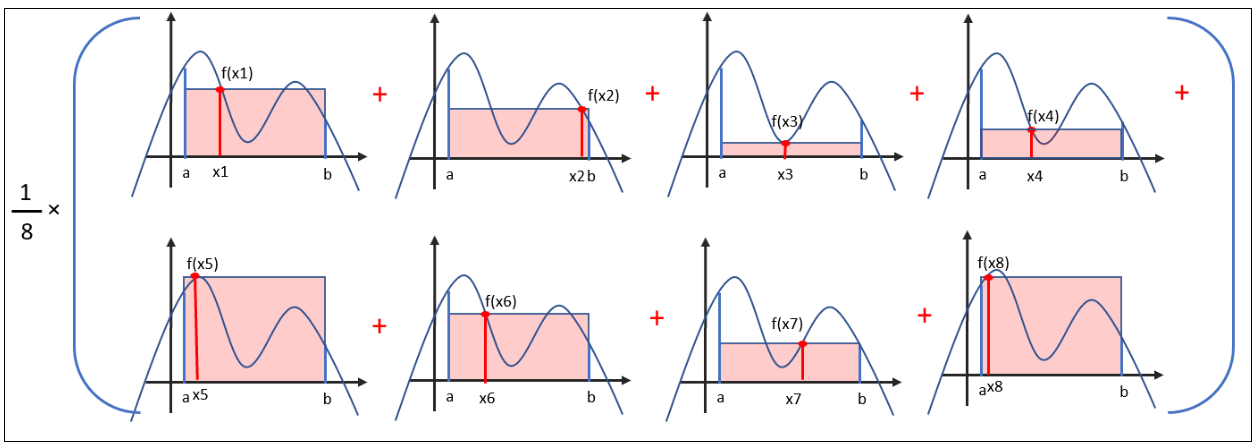


Figure 2.6: An illustration of basic Monte Carlo estimation using eight samples.

Figure 2.6 illustrates the basic Monte Carlo estimator using eight samples. This integral can be approximated by adding a rectangular region and averaging the sum. The more samples that are applied, the more accurate the acquired results will be. Given a set of N uniform random variables Xi ∈ [a,b) with a corresponding PDF of 1/(b − a), the Monte Carlo estimator for computing F is

$$\langle F^N \rangle = (b-a)\frac{1}{N-1}\sum_{i=0}^{N} f(x_i)$$

By the law of large numbers, the integrals described by the expected values of some random variables can be approximated by taking the empirical mean (i.e., sample mean) of independent samples of variables. Therefore, as N approaches infinity, the Monte Carlo estimator converges in probability to F, the true value of the integral: $\Pr\{\lim_{N\to\infty}\langle F^N \rangle = F\} = 1.$ Furthermore, as the size of sample N increases, the estimator $\langle F^N \rangle$ becomes a closer and closer approximation of F. Due to the Strong Law of Large Numbers, it can be guaranteed to have a solution with limited accuracy. By checking the convergence rate of the estimator variance, it can be seen how quickly the estimator actually converges to a sufficiently accurate solution.

Monte Carlo method is a computational algorithm that utilize the process of repeated random sampling to make numerical estimations of unknown parameters. They allow for modeling of complex situations associated with many random variables and assessing the impact of risk. Monte Carlo methods have a wide spectrum, but they all share the commonality that they rely on random number generation to solve deterministic problems. In this study Monte Carlo metho is used to determine the number of samples for experiments.

2.5 CONCLUSION

Centrality measures, a routing algorithm, and Monte Carlo concept were reviewed in this section. In next section, how these concepts are used for identifying the vulnerable roads with the experimental setups and conditions is described.

CHAPTER 3

EXPERIMENTAL SETUP

3.1 INTRODUCTION

In this chapter we first describe the simulation tool that we developed to study the vulnerability of the road network. In Section 3.3, we outline the simulation methodology and the metrics. In Section 3.4, we discuss the simulation method with random O/D pairs and in Section 3.5 we consider non-uniform O/D pairs. Finally, in Section 3.6 we determine the number of O/D pairs to simulate.

3.2 SIMULATION TOOL

Python is a high-level programming language and widely used nowadays. Python has many advantages. It is easy to read, learn and write because it is based on simplicity and efficiency. This programming language is free to use and is available under Open-Source license. It leads many programmers to make and develop application packages. These packages allow other people to save their time for making the same function programs. In this thesis, NetworkX [19] and OSMnx [20] are main packages to implement the simulation.

NetworkX is a Python package for exploration and analysis of networks. The package provides data structures for representing various networks including directed, undirected, and multigraphs. This is extremely flexible for researching graph algorithms. Nodes can be any hashable object in Python, and edges can contain arbitrary data. With connection to other packages such as SciPy, NumPy, and matplotlib, NetworkX provides strong scientific computation and makes graphs visible for helping us to understand such a complex computation.

OSMnx is a Python package for downloading OpenStreetMap street network data and then building it into NetworkX graphs. OSMnx simplifies and corrects the network's topology to ensure that nodes exclusively represent intersections and dead-ends. After the network is built and amended, OSMnx provides the shortest path from one node to another node. It can also calculate various network measures relevant to transportation networks including average street segment length, average intersection degree, intersection density, circuity, edge density, clustering coefficients, betweenness centrality and closeness centrality. By using the matplotlib package, OSMnx can easily plot various network's features like routes, high/low connectivity intersections, dead-ends, one-way streets, and figure-ground diagrams of street networks.

3.3 Simulation Method

The experiments were performed with the map of the city of Davis in California. There are several ways to download the street map. One is to load the graph from the place name which will be Davis. Another way is getting it from address and distance. However, neither way is suitable for using highways which cross the city of Davis. The best way to fully utilize the highway is to get from the b-box which is set from boundaries. The points of boundaries are 38.5754, 38.5156, -121.6759, and -121.7941 from the north, south, east, and west respectively.

Figure 3.1 shows the best paths for randomly chosen 10 O/D pairs in the Davis map.

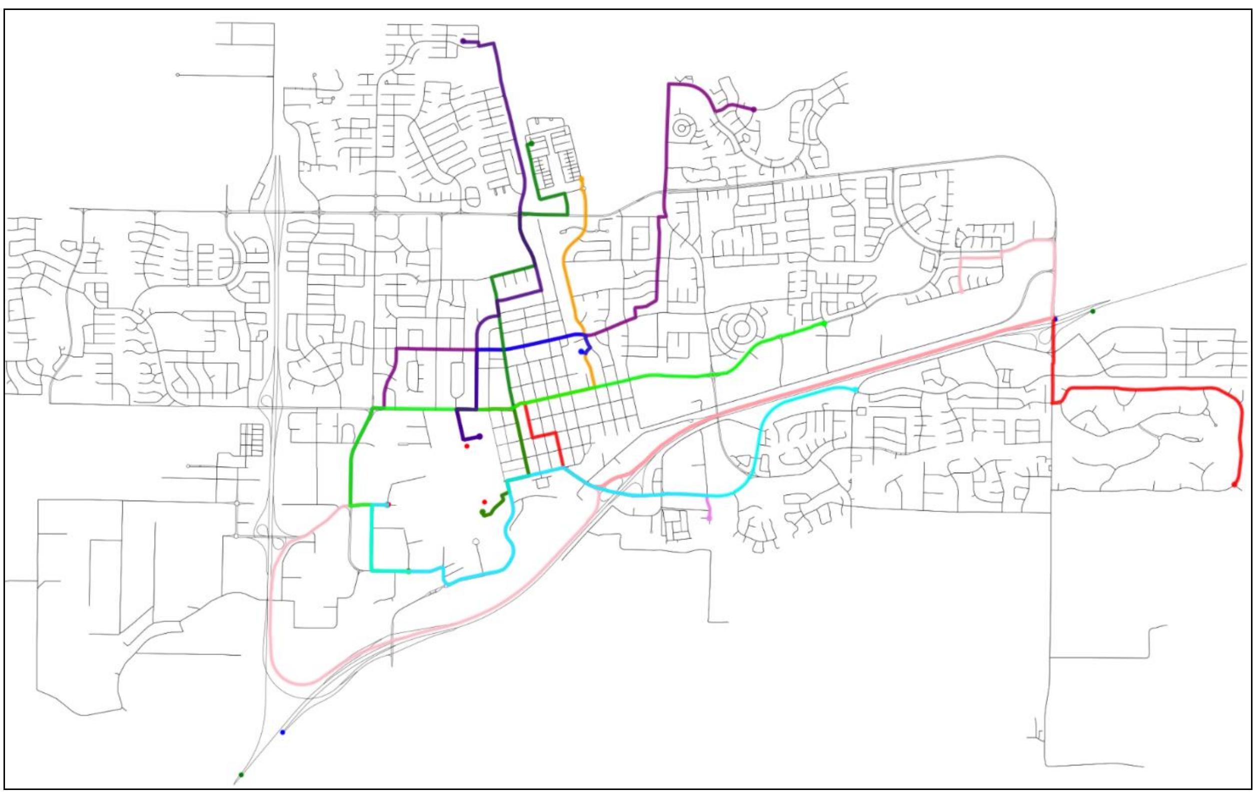


Figure 3.1 Best paths for 10 randomly chosen O/D pairs in the Davis map.

The next step is allocating the speed limit for the roads. OSMnx has basic attributes for edges like the road name, road type, road length, and whether the road is one way or not. To make use of travel time to find the best path, it is necessary to add speed limit and travel time which is derived from the speed limit and length. The base travel time is of an edge is determined as

$$Travel\ time\ T_t = \frac{Length(meter)}{Speed\ limit(kph)}$$

Due to just a small portion of edges having a speed limit, it is necessary to set the max speed for the rest of edges to calculate the travel time. OSMnx employs Kilometers Per Hour (kph) as a unit of speed instead of Miles Per Hour (mph). The set speeds are 110, 60, 50, 40, 40 (kph) respectively for the road types highway, primary, secondary, tertiary, and residential. This represents the speed limit corresponding to the speed in mph in Davis. This speed limit is used for calculating travel time for the edges. NetworkX provides the best path based on travel time by using the Dijkstra algorithm.

One of the methods to identify important links is figuring out which edges are most used over a certain period of time. To implement this method, all edges in the map have a count value which is incremented when the edge belong to the route which is the best path for a given origin and destination (O/D) pair. As more routes are created, the count for the edges increases. The larger numbers of O/D pairs give us more accurate data about which edges (roads) are most used in a given map. To determine the number of O/D pairs, we use the Monte Carlo method described in Chapter II. From experiments, we found that the top ten most used edges are constant and invariable when the number of O/D pairs is larger than 10,000. In other words, regardless of whether the number is 10,000 or 100,000 pairs, both have the same top ten dominant edges. Accordingly, in this study we have considered 10,000 O/D pairs to determine the most used edges which are denoted as critical edges.

3.4 Simulation with Random O/D Pairs

A natural heuristic to identify important edges is just to use the top five most used edges from a simulation with randomly chosen 10,000 O/D pairs. However, this approach does not incorporate the user behavior as drivers are likely to select alternative routes when faced with traffic congestion. A more practical approach is to choose only the top edge from the first iteration with 10,000 O/D pairs and then apply traffic congestion model to the top edge. The traffic congestion will alter the speed of the edge which will cause traffic delays. This information is provided to the drivers in the next phase of the simulation providing them with an opportunity to change their paths. After applying the speed change, the second iteration is executed with the same 10,000 O/D pairs that were used during the first iteration. The most used edge from the first iteration was blocked (the speed is set to 1 kph) while the second simulation was running. The second iteration yields a new most heavily used edge. The congestion model is then applied to this new edge to determine the new speed of the link. With the new Afterward, the speed of the top edge was changed again from the second iteration. By repeating those simulations five times, five edges were obtained which were more likely to have important roles for the map.

3.4.1 Simulation of Random O/D Pairs with Traffic Light Delay.

Even though the travel time is calculated using the length and the speed limit of the edges, this is not sufficient to estimate the travel time realistically. There are other factors that can make the travel time longer. One of the factors is traffic light delay. We need to consider the elapsed time between red and green lights at the intersection during which nom cars are forwarded. Also, cars must stop when they reach at stop sign and potentially wait for others who arrived at the stop sign earlier. These factors affect the travel time.

Most of the primary or secondary roads employ signal traffic lights, while most residential roads employ stop signs. Typically, it takes longer to be scheduled at a traffic light than at a stop sign. In this study, the travel time was modeled as follows:

$$Travel\ time\ T_t = B_t + L_t,\ \ Base\ time\ B_t = \frac{Length(meter)}{Speed\ limit(kph)},\ L_t = Traffic\ light\ delay$$

The set traffic light delays are 10, 10, 6, and 4 (seconds), for the road types primary, secondary, tertiary, and residential respectively.

Five edges were obtained from the five iterations after applying the traffic light delay. The five edges are much more similar to the five edges with betweenness centrality than those not applied with the traffic light delay. This time one of the critical edges from the simulation was on Interstate 80. Incorporating the traffic light delay made the simulation of the road network more realistic by incorporating vehicular traffic flow features.

3.4.2 Simulation of Random O/D Pairs with Traffic Congestion

One way to introduce traffic congestion in the simulation is to apply a traffic jam based on the count for each edge. However, for a more realistic model, congestion should include features such as the capacity of the edge, the number of lanes in the edge and the type of road. In this simulation, the congestion delay time is modeled as follows.

$$Congestion\ delay\ time\ C_t = \frac{\beta \times flow\ count\ of\ edges}{capacity\ of\ edges \times the\ number\ of\ lanes\ of\ edges}(second)$$

The β values are 40, 50, 50, 60, 60 for the road type motorway, primary, secondary, tertiary, and residential, respectively. Flow count of edges means that the number of counts (i.e. flows) use that edge. The capacity of an edge depends on the type of road: 2400, 2000, 2000, 1800, 1800 corresponding to motorway, primary, secondary, tertiary, and residential, respectively. The number of lanes for each edges depends on the edge attribute. Some edges have multiple lanes, while most lanes have only 1. Specifically, the number of lanes for Russell Boulevard and East or West Covell Boulevard was set to 2; all other streets within the city had 1 lane.

The congestion delay time in an edge C_t was set to be inversely proportional to the number of lanes in the edge as shown below. For example, the flow count of an edge is 1200, the type of roads is primary, and the number of lanes is two, the C_t is sixty-six seconds.

$$Travel\ time\ T_t = B_t + L_t + C_t,\ \ B_t = \frac{Length(meter)}{Speed\ limit(kph)},$$

$$L_t = Traffic\ light\ delay, \qquad C_t = \frac{\beta \times flow\ count\ of\ edges}{capacity\ of\ edges \times number\ of\ lanes\ in\ the\ edge}$$

Each iteration of the simulation was run with total of 20,000 O/D pairs consisting of 10 steps each with 2000 O/D pairs. Traffic congestion was applied at each step by using the accumulated flow count for each edge to determine the C_t. For the initial first step, only B_t and L_t was used to determine the number of counted edges. For the second step, the travel time was calculated using B_t, L_t, and C_t where C_t was calculated by using the flow count determined from the previous step. The flow count was accumulated at every step. For example, in step seven, the flow count for an edge is the the sum of the flow count from each of the previous steps one through step six.

Consequently, the traffic congestion time worsened with number of steps as the flow count increases. This allowed modeling how the traffic adapts to traffic congestion variation. For the tenth step which was the last step for first iteration, the most used edge, i.e., one with the most flow count was obtained. The next iteration of simulation was performed after removing the most used edge (setting the speed of the edge to 1 kph). This was repeated for a total of 5 iterations to study how location of the critical link (defined to be the one that is most used) changed as the critical links were sequentially removed.

3.5 Simulation with Non-Uniform O/D Pairs

Traffic pattern is influenced such as the time of day, day of week, public events, weather conditions, and holidays. Different traffic patterns impose different demands and may lead to congestion at different locations. Given the traffic pattern, it is possible to estimate the traffic congestion for the city. For instance, on any weekday morning, a large fraction of the traffic from the highway and elsewhere in the city Davis goes to the UC Davis campus. To study the impact of this traffic pattern, we considered a non-uniform distribution of O/D pairs. Specifically, with regards to the destination we assumed that 50% of the traffic went to the UC Davis campus, another 10% went to the to five grocery stores in Davis, and another 10% went to downtown of Davis, and another 30% went to random destinations in Davis. With regards to the source, we assumed that 15% of sources came from eastbound I-80, 10% of sources came from westbound I-80, and other 75% of sources from somewhere within Davis.

The simulation steps and iterations with non-uniform O/D pairs was conducted in the same way as with the uniform O/D pairs.

3.6 Determining the Number of O/D Pair to Simulate

One of the goals for the simulation with the uniformed samples was to determine how many O/D pairs would be needed to model appropriate level of traffic congestion. If more O/D pairs are tested, more accurate results will be obtained from the background with Monte Carlo Integration, which can be useful to exploit for a probabilistic interpretation.

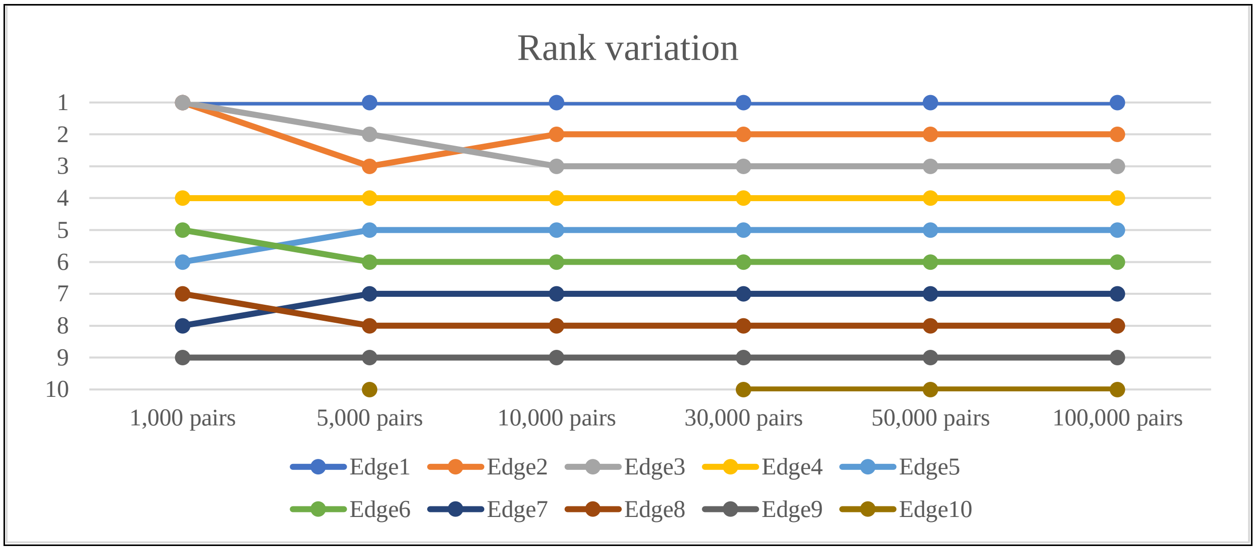


Figure 3.2 Variation of edge rank a function of the number of O/D pairs.

Figure 3.2 shows the rank variation of the top 10 most used edges based on the number of given O/D pairs. The edge's ranks are constant if more than 10,000 O/D pairs are given. The chart informs us the marginal number of O/D pairs is 10,000 pairs. Therefore, 10,000 O/D pairs were mainly used in the simulations of experiments.

3.7 Conclusion

Simulation tools, methods, and different conditions to identify the vulnerable roads were described in this section. In next section, the results of the experiments with these contents are discussed.

CHAPTER 4

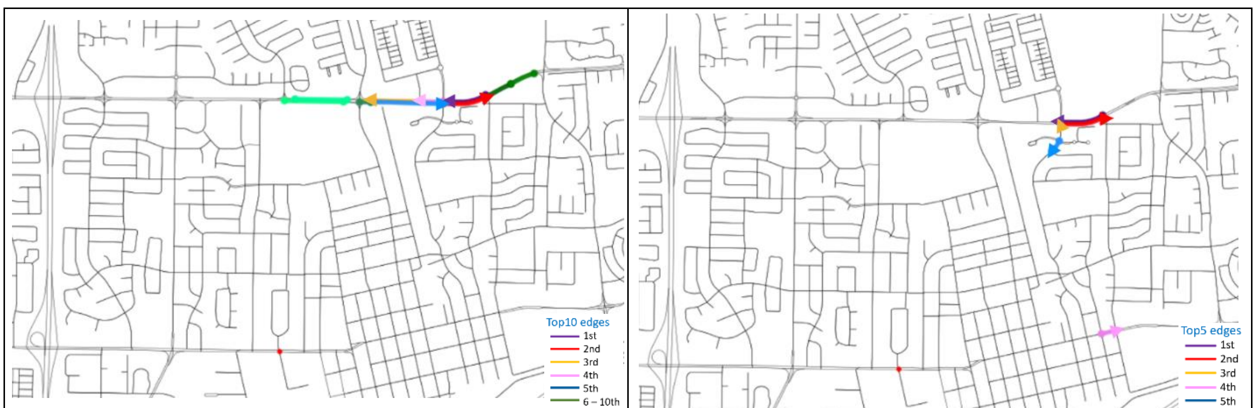
RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

In this chapter we discuss the results from the various simulation experiments. The results are presented as five key observations. Additionally, the impact of different traffic pattern on the network are also presented.

4.2 OBSERVATION 1: NON-LOCAL MIGRATION OF CRITICAL EDGE

Recall that the edge with the largest flow count is the critical edge. An important question is if the critical edge is removed which edge becomes the next critical edge. We studied this with uniform O/D pairs. For a benchmark we first determined the top five critical edges from one iteration of the simulation. Next we determined the top five critical edges following the five iterations of the simulation where in each iteration we remove the critical edge. Figure 4.1(a) shows the top five edges from one iteration of the simulation. The edges are close to each other and lie along the same primary road (E.Covell Blvd). This was because the edges that were connected to the edge with the most flow count were likely to have high flow count so that the top five counted edges were concentrated at specific roads.



Top five edges from first simulation.	Top five edges from five step simulation.

Figure 4.1 Location of the critical edge with accumulated sequential disruptions under uniform O/D pairs.

Figure 4.1(b) shows the top five critical edges where at the end of each iteration the most critical edge is removed. To be specific, at the end of the first iteration, the purple color edge removed. In second iteration, the red edge was the critical edge which was removed for the third iteration and so on. In this case we observe that the location of the critical edge changes different part of the network still remaining on the primary roads such as the Fifth Street and the J Street. This was because alternate routes were used when the critical edge was blocked. The result show that under uniform O/D pairs there is non-local effect in how the critical edge migrates when disruptions are injected in the network. Figure 4.2 This phenomenon was remarkable in the simulation with traffic light delay.

(a) Top five critical edges from one iteration of the simulation.

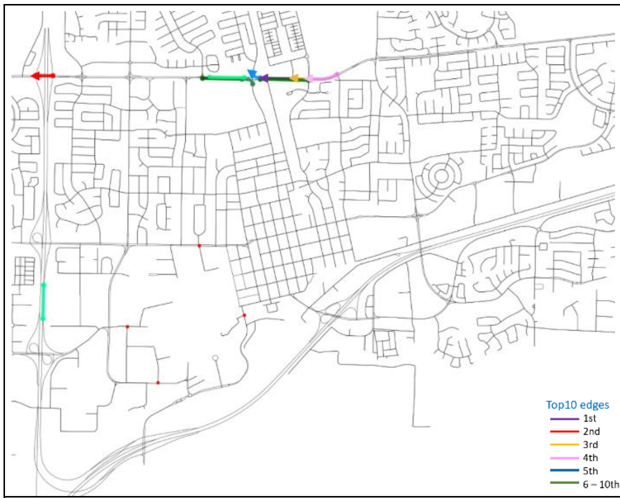
(b) Top five critical edges from five iteration of the simulation with disruptions.

Figure 4.2 Location of the critical edge with accumulated sequential disruptions under uniform O/D pairs and with traffic light delay.

With traffic light delay migration of the critical edge is even more widespread and demonstrate the non-local effect that influences how the critical edge migrates when disruptions are injected in network.

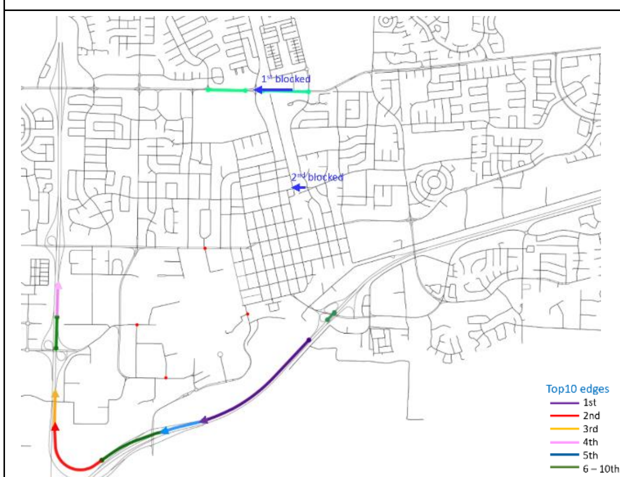
4.3 Observation 2: Impact of Traffic Light Delay

All the five critical edges from the simulation with uniformed O/D pairs were mostly located in the middle of the map. This result meant that the Interstate 80 was not much used for the best paths of O/D pairs even though the highway was used by drivers a lot in the real world. This posed a problem in that the first result did not reflect the reality adequately. All the edges (roads) have different speed limit based on the road's type. Thus, a highway with a higher speed limit is more likely to be used than other roads such as primary, secondary, and residential with lower speed limits. However, the simulation with uniform O/D pairs did not present the preference for the highway in the real world. This was due to the fact that there were no traffic lights applied to the map even though all roads had travel time based on the length and speed limit. Most intersections have a traffic light or a stop sign which results in additional delay in the travel time of the edge. However, the highway does not have the traffic light except at intersections at the highway ramp. By adding some traffic light delays to primary, secondary, tertiary, and residential roads far more realistic results were obtained.

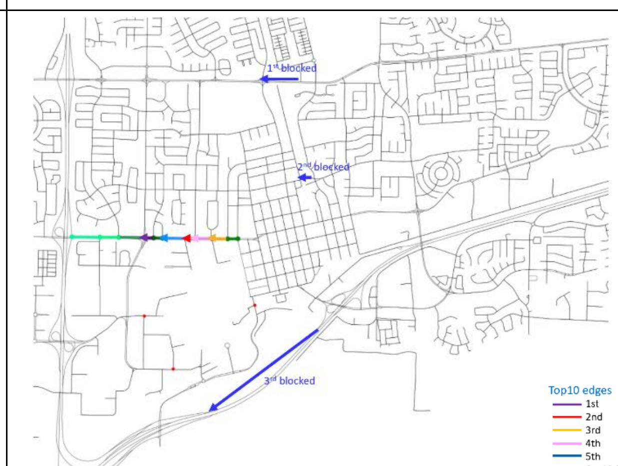


The top 10 edges from 1st iteration

The top 10 edges from 2nd iteration



The top 10 edges from 3rd iteration



The top 10 edges from 4th iteration

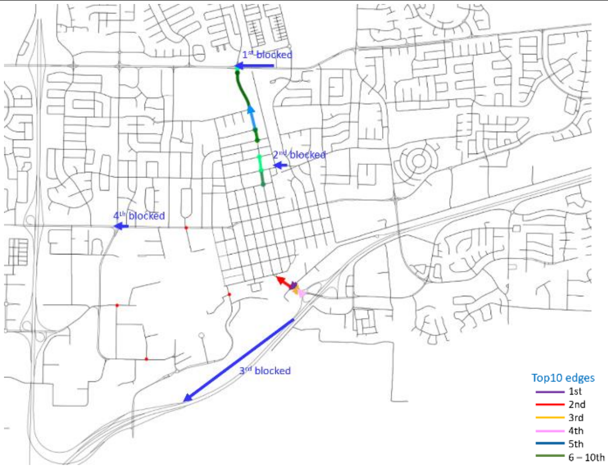
	
The top 10 edges from 5th iteration	

Figure 4.3 Changes in traffic flow with traffic light delay with sequential removal of critical edge. At the end of each iteration the critical edge is removed.

Figure 4.3 shows changes in traffic flow of the simulation with traffic light delays. In first iteration, major traffic flow focused on Covell Boulevard. In second iteration, the critical edge was on Eighth Street, but lot of traffic moved to Interstate 80. Afterward, when Interstate 80 and highway 113 were crowded, then traffic flow moved to Russel Boulevard. In the fifth iteration, the major traffic direction was on between north and south and especially traffic flow moved to F Street and Richards Boulevard, which is one of the major roads to connect between north and south of Davis.

The first remarkable point was that the five critical edges from the simulation were more in the east and west direction than north and south direction. The reason for this is that the map of Davis is much wider from east to west than from north to south. This phenomenon made the traffic flow between east and west dominant.

Another intriguing result from this simulation was that if the first road was blocked, the next critical road was not adjacent to the first road but far from the road. This is opposite to common sense of just using the alternative roads which are closest to the first road, but, in reality, traffic finds a new path which maybe remote from the road. This result presents non-local effect, which demonstrates that the impact of traffic disturbances did impact locally but widely in the network.

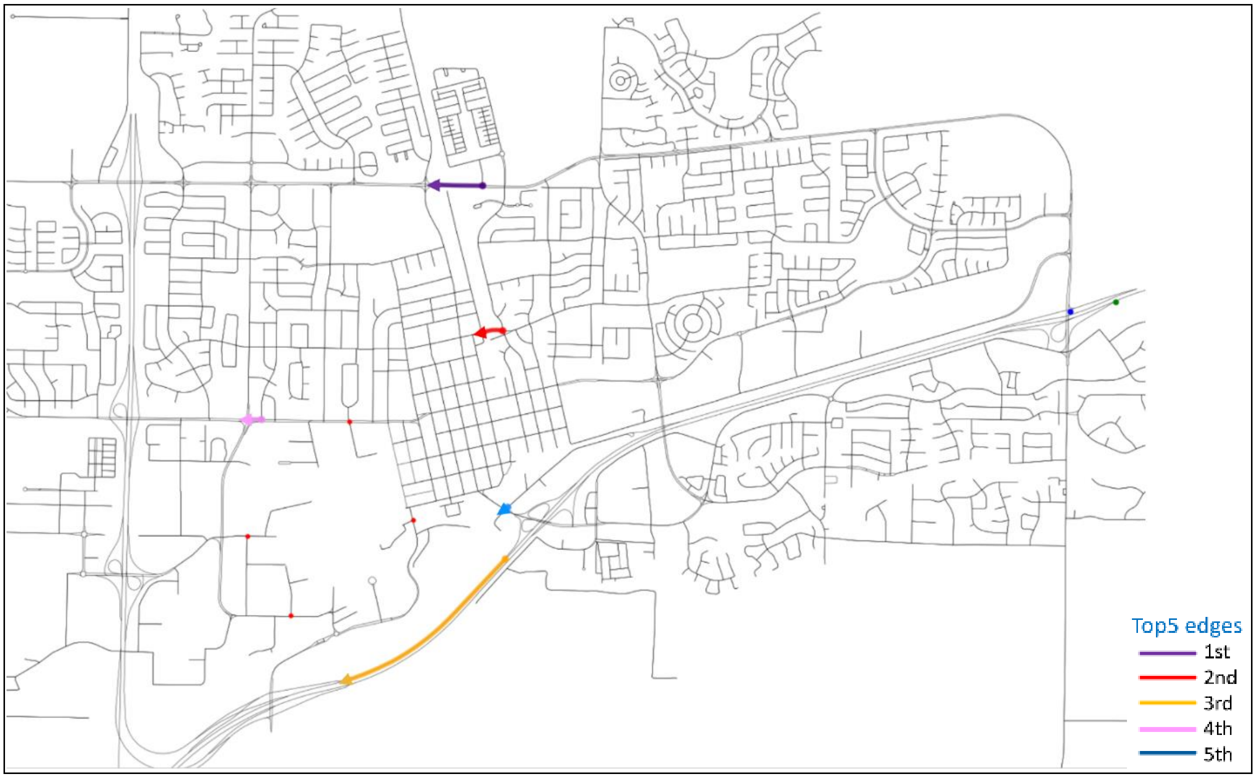


Figure 4.4 The top 5 edges from simulation of uniformed O/D pairs with traffic light delay.

Figure 4.4 shows the considerable difference from the previous result after applying the traffic light delay. The result was obtained from five-step simulation after applying traffic light delays.

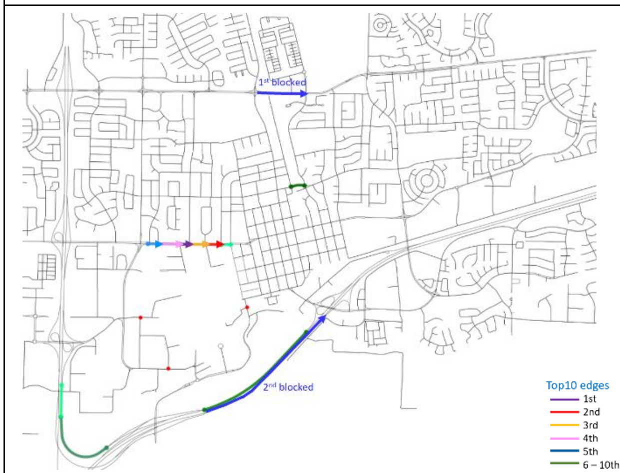
The result presented shows that the five critical edges were widespread and incorporating traffic light delays resulted in the Interstate 80 started to be used. It was intriguing to note that the 5 edges had intersection with those determined by the betweenness centrality measure. We discuss this in detail in Section 4.5. By adding traffic light delays, the usage of Interstate 80 increased, and in contrast, local roads' usage decreased.

4.4 Observation 3: Impact of Congestion

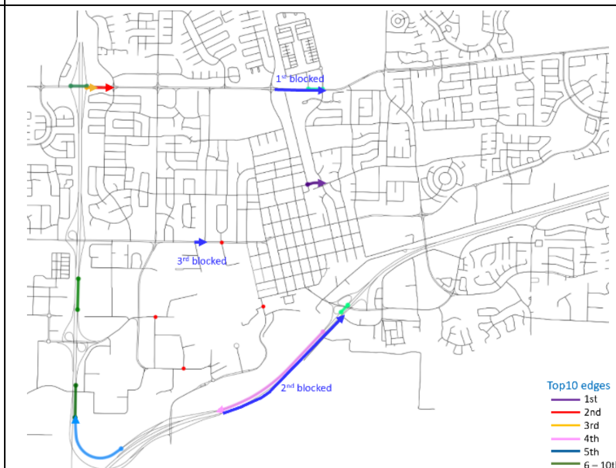
The previous study did not consider delays due to traffic congestion. In real life, traffic congestion conditions vary continuously. To introduce this feature, a new congestion model was applied. This was described in Chapter 3. One of the key factors of the congestion model is the capacity of the edges which depended on the road type and the number of lanes and played a significant role for the major roads that were utilized frequently.

The top 10 edges from 1st iteration

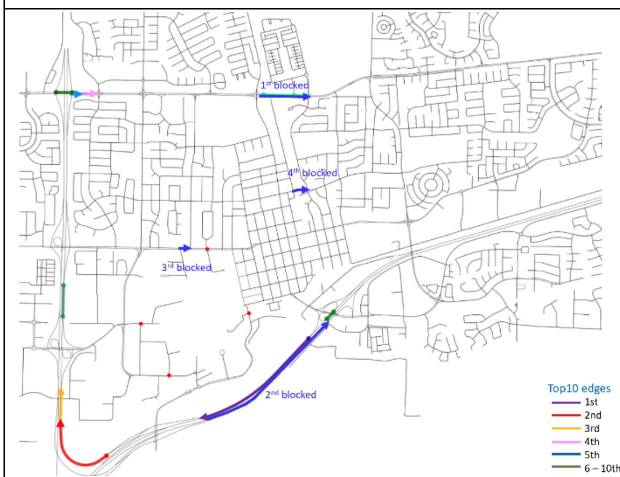
The top 10 edges from 2nd iteration



The top 10 edges from 3rd iteration



The top 10 edges from 4th iteration



The top 10 edges from 5th iteration

Figure 4.5 Changes in traffic flow of simulation with traffic congestion

Figure 4.5 shows the result of simulation with traffic congestion. In first iteration, major traffic flow was concentrated on Covell Boulevard and Interstate 80. In second iteration, not only the most used road was on Interstate 80, but also most traffic was on Interstate 80. Afterward, when Interstate 80 and Covell Boulevard were crowded, then traffic flow moved to Russel Boulevard. In the fourth iteration, the top crowded road was on Eight Street that was one of the major alternative roads to connect east and west of Davis. In the fifth iteration, the most crowed road was on Interstate 80. However, the road's direction was east to west which was opposite to the one in the second iteration. Also, the major traffic flow was on Interstate 80 west direction.

One of notable results from this simulation was that major traffic flow was on Interstate 80 in both directions of east and west. This result was much closer to the reality that highway was preferred.

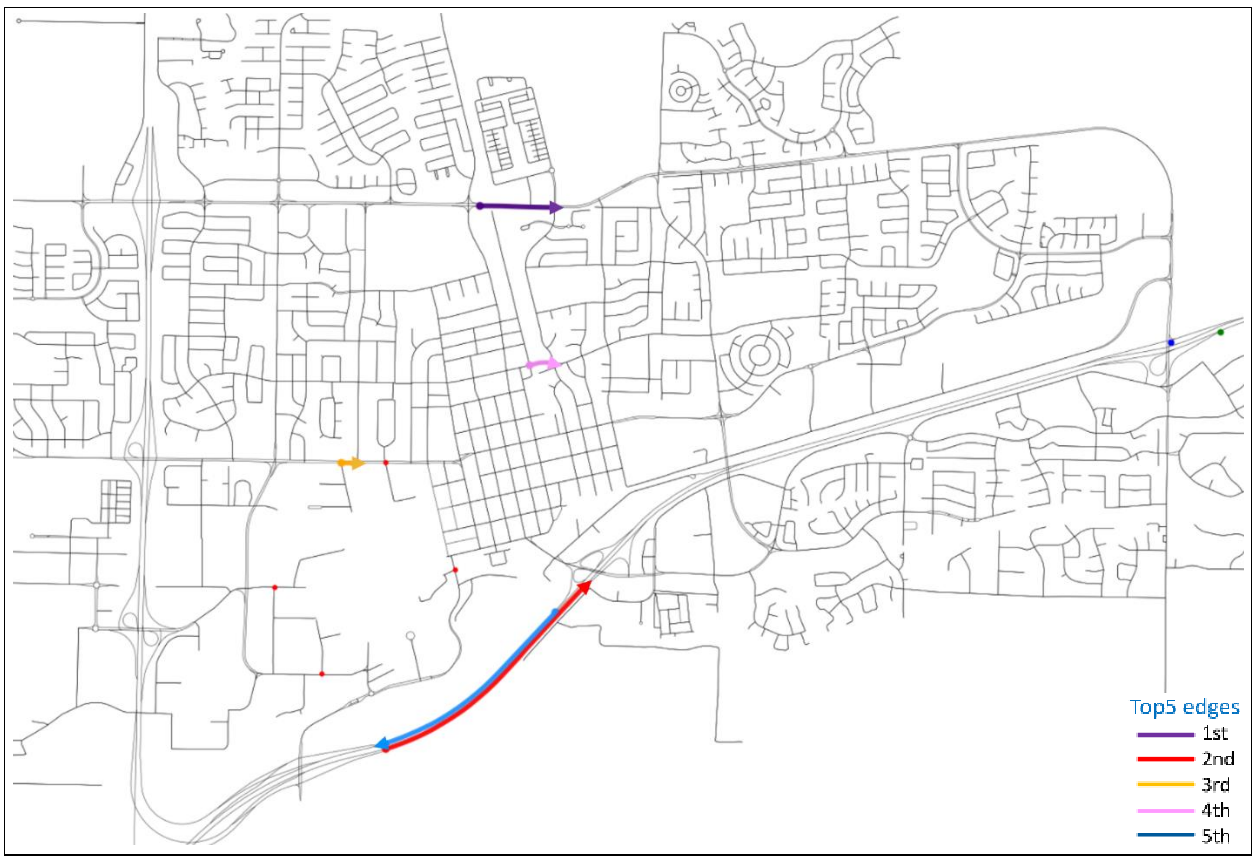


Figure 4.6 The top 5 edges from simulation with congestion model.

Figure 4.6 shows the top five vulnerable edges from the simulation with applying congestion model.

The five critical roads from this simulation were also widespread, and Interstate 80 was dominant. It was interesting that the most vulnerable edges also had some relationship with those from betweenness centrality measure.

4.5 Observation 4: Relationship with Betweenness Centrality

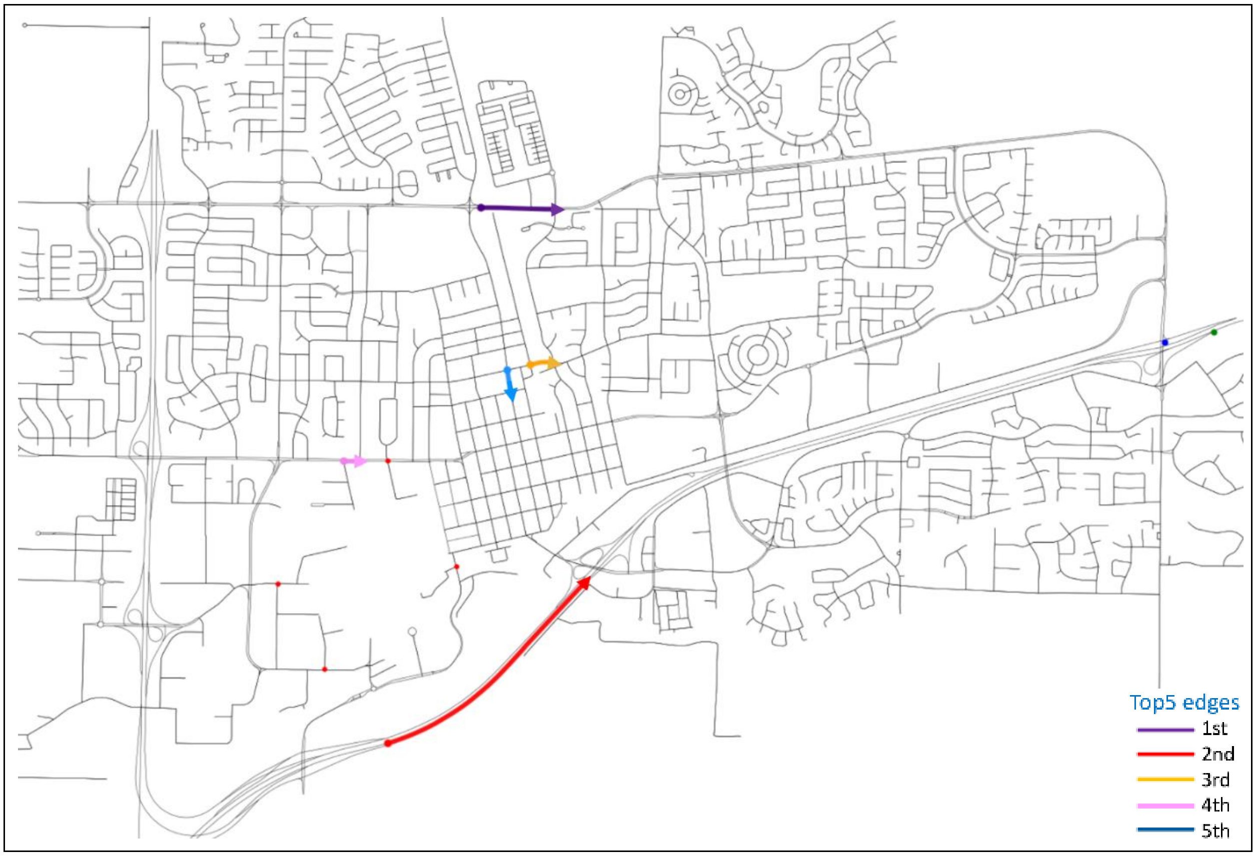
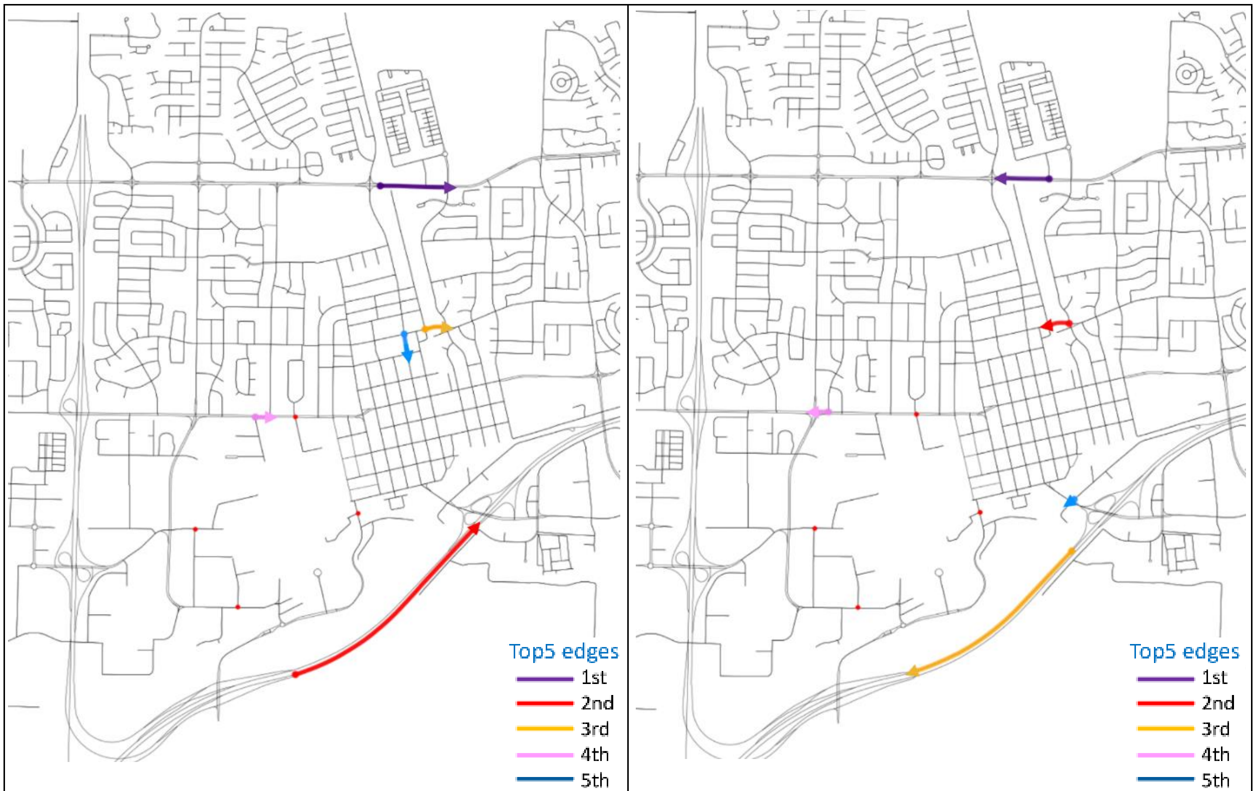


Figure 4.7 The top five edges from betweenness centrality by five steps.

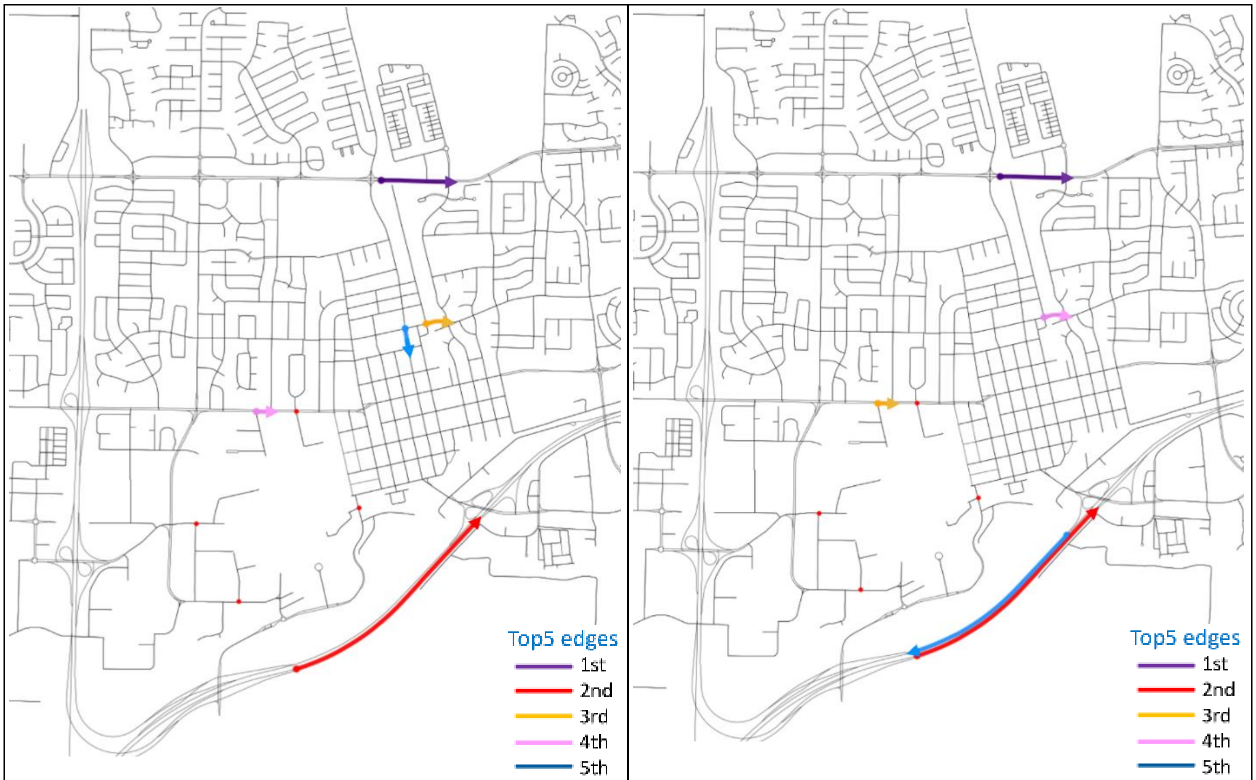
Five edges with the highest betweenness centrality were acquired by five-step simulation. First, the betweenness centrality values were calculated for the whole map and the first edge with the highest value among all the edges was chosen. Next, the first edge was removed and betweenness centrality values were recalculated. Afterward, the second edge with the highest value was obtained. By doing so, five edges were obtained from the betweenness centrality measure.



The top five edges from betweenness centrality	The top five edges with traffic light delay

Figure 4.8 comparing top five edges based on different measures

Figure 4.8 compares two results. One is betweenness centrality and the other is from the simulation with traffic light delay. Two results showed similarity of vulnerable roads, i.e., three of five edges were close to each other.



The top five edges from betweenness centrality	The top five edges with the congestion model

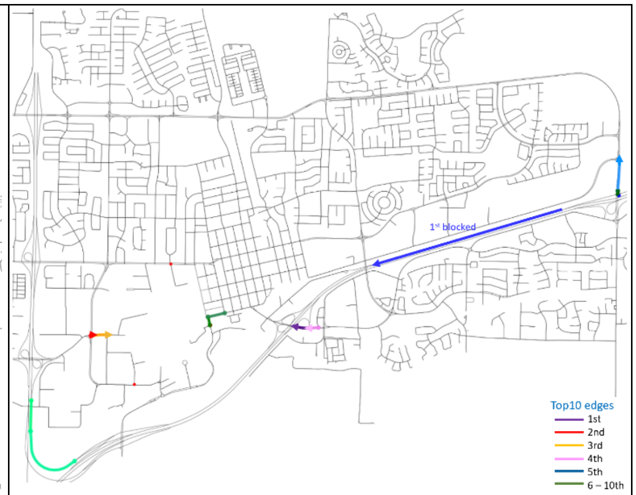
Figure 4.9 compares the top five edges based on different measures

Figure 4.9 showed similarity of vulnerable roads from two different measures that were obtained from betweenness centrality and the simulation with the congestion model. This result indicated that betweenness centrality measure had a significant role in identifying vulnerable roads in ordinary traffic networks. Only one of five edges was different while four roads were exactly same. Betweenness centrality can be a useful method to ascertain critical edges under normal conditions in networks.

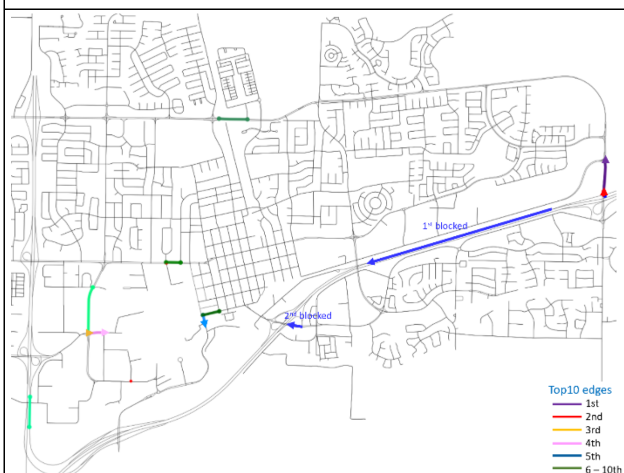
4.6 Observation 5: Impact of Non-uniform Traffic Pattern

Some proportion of source and destination was changed to identify critical roads from traffic pattern changes since traffic congestion can be influenced by different conditions such as the time of day, day of week, holdays, events, and weather. The data was set under conditions in Davis on weekday morning.

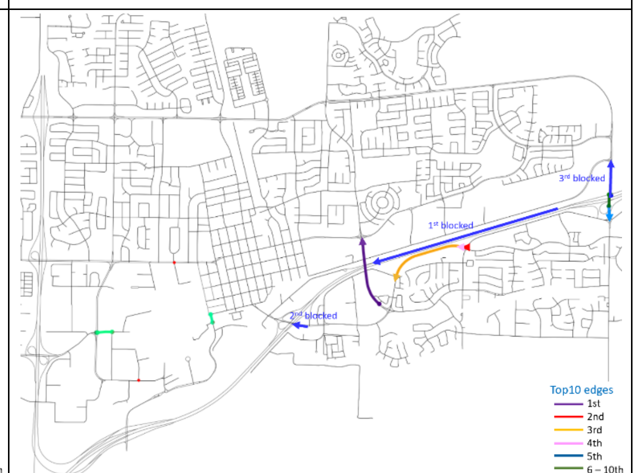
The top 10 edges from 1st iteration



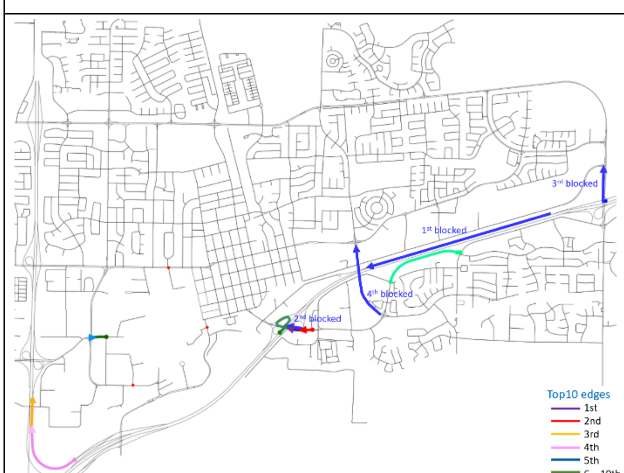
The top 10 edges from 2nd iteration



The top 10 edges from 3rd iteration



The top 10 edges from 4th iteration



The top 10 edges from 5th iteration

Figure 4.10 Changes in traffic flow of simulation of specifically targeted O/D pairs with congestion model.

Figure 4.7 shows changes in traffic flow of the simulation of specifically designated O/D pairs with the congestion model. In first iteration, the major traffic flow was on Interstate 80, specifically east to west direction. This was because 15% of source nodes were on east of I-80 and the half of destination nodes were in UC Davis. Such a particular changed rate of origin and destination nodes created most traffic flows from east to west. In the second iteration, the most critical road was on Richards Boulevard, which connected with the ramp of Interstate 80 east and west-bound. There were two reasons for this. The road on Richards Boulevard was on the path that was the alternative way to UC Davis from east of Interstate 80. Also, Richards Boulevard is one of the major roads to connect North and South Davis. Then the most critical road moved to Mace Boulevard, which was on another alternative route to go to UC Davis campus via Second Street. In the fourth iteration, the busiest road was on Pole Line Road, which is another major road to connect North and South Davis. Actually, Pole Line Road was the only possible option to go to North of Davis from South after blocking Richards Boulevard and Mace Boulevard. In the fifth iteration, the critical road was the same road on Richards Boulevard from the second iteration. After blocking the tree major roads that connect North and South Davis, the road on Richards Boulevard was the best choice to reach the destination.

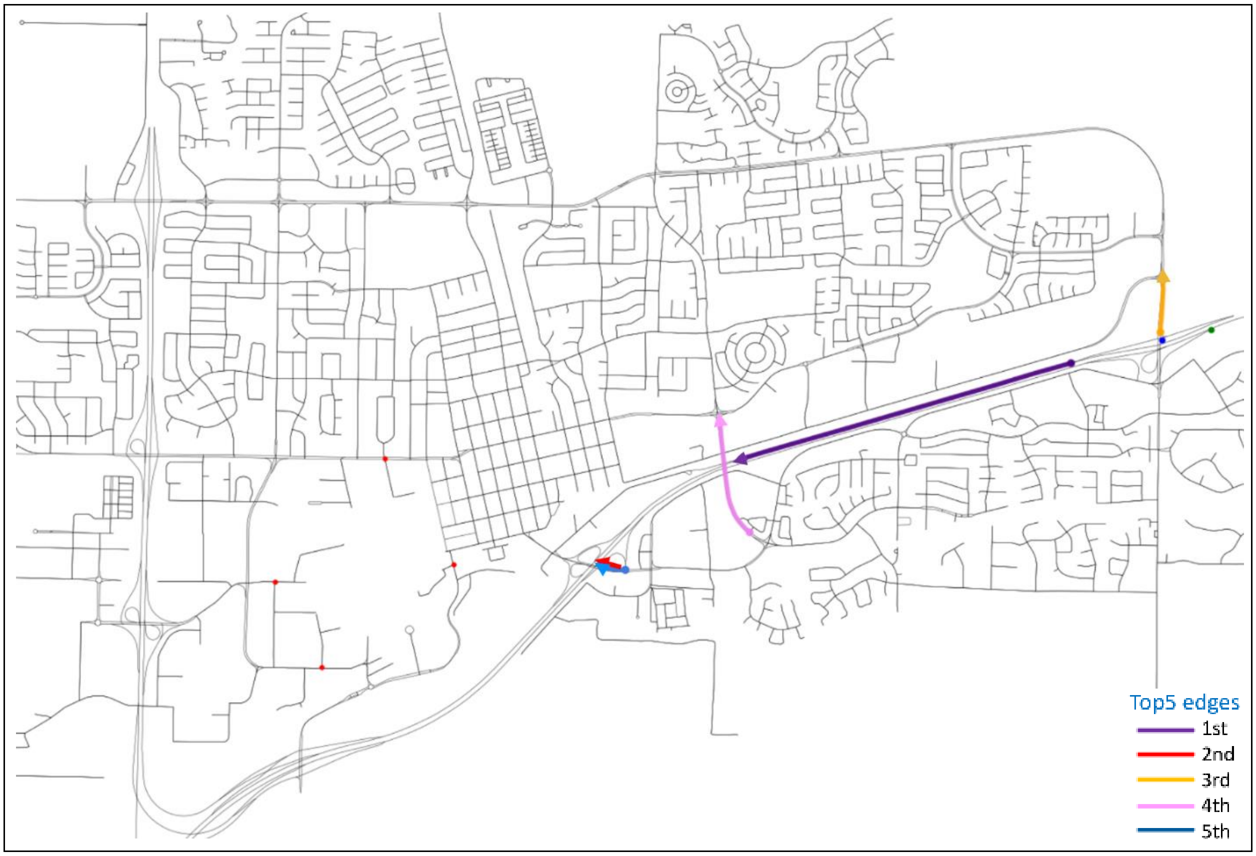


Figure 4.11 The top 5 edges from Simulation of targeted O/D pairs with applying congestion model.

It was demonstrated that the traffic pattern obviously changed by specifying origin and destination nodes which was non-uniform so that the critical top five edges were different with those from uniform O/D pairs. This result indicated that the traffic statistics allowed us to predict the traffic pattern and congestion. The stats helped drivers the view of what’s happening on the roads. If the data resources can be secured officially, the estimated traffic pattern will be more precise and accurate.

4.7 IMPACT OF VULNERABLE ROADS' ON THE NETWORKS IN DIFFERENT CONDITIONS

4.7.1 MEAN TRAVEL TIME DELAY

Travel time (second)	Uniformed O/D pairs					Uniformed O/D with traffic light delay				
	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th
Mean	336.3	343.5	350.4	354.2	355.9	502.5	516.6	525.7	548.4	554.5
Difference		7.2	6.9	3.8	0.8		14.1	9.1	22.7	6.1
75th %	435.4	445.5	458.1	465.0	467.5	654.3	675.7	688.2	712.2	717.1
Median	326.6	333.0	339.7	343.7	344.7	502.1	514.6	521.1	528.9	531.1
25th %	225.7	229.3	233.1	236.0	236.4	349.9	355.2	358.1	359.7	360.1
Travel time (second)	Applied traffic congestion model					Targeted O/D pairs with congestion model				
	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th
Mean	628.5	654.9	701.3	719.5	738.6	625.8	724.4	779.5	875.5	968.3
Difference		26.4	46.4	18.2	19.1		98.6	55.1	96	93.3
75th %	821.4	862.2	915.1	936.5	963.5	790.5	919.9	996.4	1110.5	1344.6
Median	614.2	635.3	650.9	659.2	669.7	607.8	663.8	695.2	732.8	744.9
25th %	419.3	429.6	433.6	437.1	440.9	441.8	466.5	480.6	494.9	492.5

Table 4.1 The variance of traffic time for simulations of four different conditions

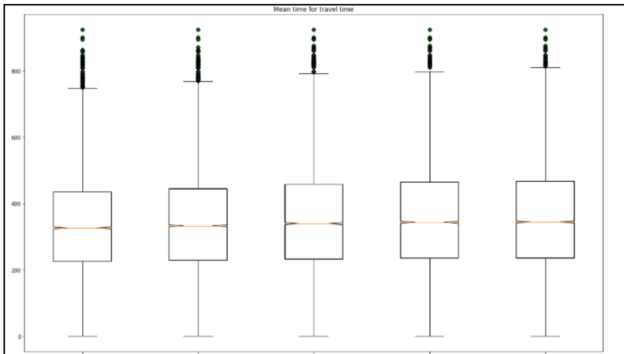
Table 4.1 shows average travel times for four different simulations. The simulations consisted of five iterations, and the mean, 75th, median, and 25th percentile travel times were calculated respectively in each iteration. Difference indicates mean time increased between the previous iteration and the current iteration. Difference is one of the significant features since it means increased travel time caused by blocking the important edge between iterations. Particularly it indicates the extent of the impact on the whole network from the critical edges.

Mean time of the applied congestion model is far higher than the one with traffic light delays. This is because the congestion model already includes traffic light delays as well as congestion delays.

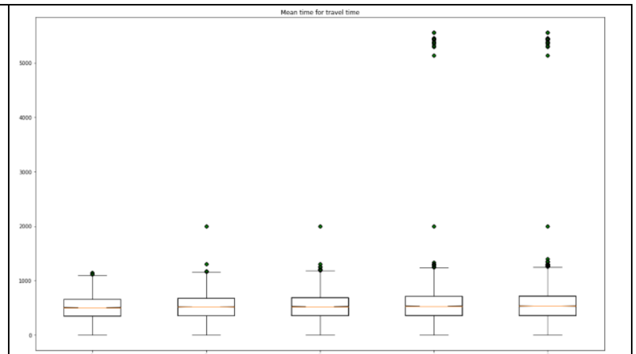
In addition, travel time that congestion model applied was accumulated for ten steps so the travel time also gradually escalated over later steps.

4.7.2 Difference of Increased Travel Time.

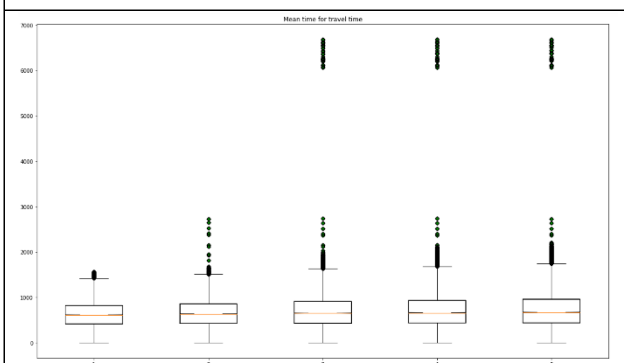
Box plot [21] is a useful method of data display, showing not only some indication of the symmetry and skewness of the data but also the distribution/spread of data. In addition, box plot indicates the minimum and maximum values, and quartiles of data.



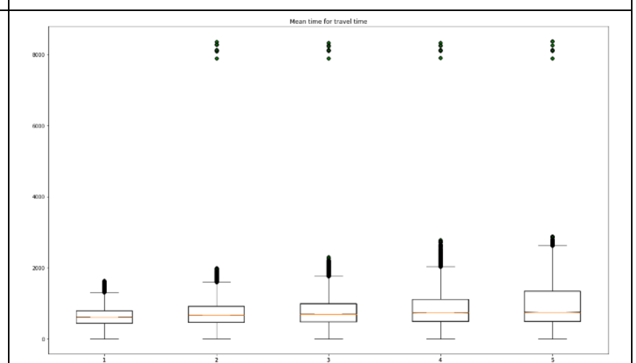
a) Uniformed O/D pairs samples



b) Traffic light delay applied



c) Traffic congestion model applied



d) Targeted O/D pairs

Figure 4.12: box plot for four different simulations. a) Simulation with random O/D pairs. b) Simulation of random O/D pairs with traffic light delay. c) Simulation of random O/D pairs with traffic congestion. d) Simulation with non-uniform O/D pairs.

The graphs in Figure 4.12 display the extent of the increased travel time for five iterations of different four simulations. From the box plot, it is clear that mean travel time increases as iterations repeat. One of the remarkable points was that travel times significantly increased at the certain point of iterations. This was because some of O/D pairs did not have any alternate routes so that the O/D pairs had to use the vulnerable roads which was blocked from previous iterations.

4.8 SUMMARY OF RESULTS

A way to identify important edges which is just to use the top five most used edges from the first simulation with 10,000 O/D pairs is not practical. Since, it does not apply the user's behavior that drivers are likely to select alternative routes when faced with traffic congestion. A more realistic approach is to choose only the top edge from the first iteration with 10,000 O/D pairs and then apply traffic congestion by using an appropriate method to the top edge. After applying traffic congestion, the next top edge is obtained from the next iteration. Doing so for five times, which is called a five-step way, better results can be secured.

The simulation with uniformed samples did not present the preference for the highway in the real world. Far more realistic results were secured by applying traffic light delays or traffic congestion model. Moreover, the results from the simulation with traffic light delays or traffic congestion showed non-local effect which was that if the first road was blocked, the next critical road was not adjacent to the first road but far from the road. As a result, the traffic disturbances happened not just locally, but widely.

Similarity of vulnerable roads from two different measures was confirmed. One was obtained from the betweenness centrality, and the other was from the simulation of the applied congestion model. This result indicated that betweenness centrality was a useful tool in identifying vulnerable roads in ordinary traffic networks.

It was shown that specifying non-uniform origin and destination nodes caused the traffic pattern to alter completely, with the vulnerable five edges differing from those of uniform O/D pairs. As a result, traffic patterns and congestion were able to be predicted by using traffic information.

CONCLUSIONS

The top five observations such as non-local effect, impact of applying traffic light delay and congestion, relationship with betweenness centrality, and the impact of non-uniform traffic pattern were described in this chapter. Also, mean travel time delay and the difference of travel time based on simulations were explained.

CHAPTER 5

CONCLUSION

In this thesis, it was simulated how traffic pattern change, what features affect the diversity of traffic patterns, and what reasons contributed to the results.

It was not realistic to use the top five most used edges from the first simulation with 10,000 O/D pairs to select critical edges. It did not take into account the user's behavior because drivers were more likely to choose alternate routes when confronted with traffic congestion. A more realistic technique was to choose only the top edge from the first 10,000 O/D pairs iteration and then apply traffic congestion to the top edge using an acceptable manner. The next top edge was obtained from the next iteration after traffic congestion had been applied. Better results were obtained by repeating the process five times.

When a road was blocked, the next traffic congestion happened to the road that was far from the blocked road. The traffic congestion was widespread, not just local. This result described non-local effect of vulnerable edges adequately and also implied that traffic congestion in small areas could impact large areas and if some roads were blocked, it could lead to a whole network disturbance.

Betweenness centrality is an applicable measure to analyze traffic networks quickly under ordinary conditions. The results presented from simulations showed that there were strong relationships between the simulation and betweenness centrality.

It will be interesting to explore traffic delay estimates by introducing machine learning with several features such as betweenness centrality, degree measure, and the rate of use per the number of O/D pairs. Also, the study with machine learning to predict the delay time after learning those labeled data will be helpful for future traffic control systems.

References

[1] Google Maps, 2021, https://maps.app.goo.gl/googlemaps

[2] Waze. 2021. Waze Mobile Ltd.(version 4.75.0.3). [Mobile app]

[3] MapQuest, 2021. System1 LLC.(version 3.28.1). [Mobile app]

[4] The Society of Automotive Engineeers (SAE).

[5] By Jay Ramey "Level 4 Autonomos Cars Begin Real – World Testing in Oxford", Oct 2020. Autoweek.com

[6] David Elliott, Walter Keen, Lei Miao, Recent advances in connected and automated vehicles, Journal of Traffic and Transportation Engineering (English Edition), Volume 6, Issue 2, 2019, Pages 109-131, ISSN 2095-7564, https://doi.org/10.1016/j.jtte.2018.09.005.

[7] Yang Yang, Takashi Nishikawa and Adilson E. Motter, "Small vulnerable sets determine large network cascades in power grids**",** *Science* **358** (6365), eaan3184. DOI: 10.1126/science.aan3184

[8] Noah E. Freiedkin, "Theoretical Foundations for Centrality Measures".

[9] Opsahl, T., Agneessens, F., Skvoretz, J., 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks 32 (3), 245-251.

[10] Francisco Aparecido Rodrigues (2019), Network centrality: an introduction.

[11] Newman, M. E. J.**Social Networks** Vol. 27, Iss. 1, (Jan 2005): 39-54. DOI:10.1016/j.socnet. 2004.11.009

[12] Freeman, L. C., 1978. Centrality in social networks: Conceptual clarification. Social Networks 1, 215-239.

[13] Jennifer Golbeck, Analyzing the Social Web, Morgan Kaufmann, 2013, Pages 25-44, ISBN 9780124055315, https://doi.org/10.1016/B978-0-12-405531-5.00003-1.

[14] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs" . *Numerische Mathematik*. **1**: 269–271. doi:10.1007/BF01386390. S2CID 123284777.

[15] Sniedovich, M. 2006. Dijkstras algorithm revisited: the dynamic programming connexion. Control and Cybernetics 35(3):599–620.

[16] Ariel Felner. 2011. Position Paper: Dijkstra's Algorithm versus Uniform Cost Search or a Case Against Dijkstra's Algorithm Proceedings, The Fourth International Symposium on Combinatorial Search (SoCS-2011).

[17] Sachs, I., Sen, S., & Sexton, J. (2006). Monte Carlo integration. In *Elements of Statistical Mechanics: With an Introduction to Quantum Field Theory and Numerical Simulation* (pp. 82-111). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511755620.006.

[18] Wojciech Jarosz. *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. Ph.D. dissertation, UC San Diego, September 2008. (pp. 149-154).

[19] NetworkX -Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008.

[20] OSMNX : Boeing, G. 2017. "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks." Computers, Environment and Urban Systems 65, 126-139. doi:10.1016/j.compenvurbsys.2017.05.004.

[21] Wickham, Hadley; Stryjewski, Lisa. "40 years of boxplots". Retrieved December 24, 2020.