THE EFFECTS OF REDUNDANCY AND INFORMATION MANIPULATION ON TRAFFIC NETWORKS

by

BERK ÖZEL

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Master of Science

Sabancı University

Fall 2013

THE EFFECTS OF REDUNDANCY AND INFORMATION MANIPULATION

ON TRAFFIC NETWORKS

APPROVED BY:

Prof. Dr. Ali Rana Atılgan ....

(Thesis Supervisor)

Asst. Prof. Dr. Ahmet Kutsi Nircan ....

Assoc. Prof. Dr. Güvenç Şahin ....

Assoc. Prof. Dr. Nilay Noyan ....

Prof. Dr. Canan Atılgan ....

DATE OF APPROVAL: ....JANUARY 13, 2014..........

ACKNOWLEDGEMENTS

First of all, I would like to thank to my family, for showing their endless love, care, and tolerance all the time.

I also thank to Merve, the most understanding person I have ever known, not for only helping me to handle the excessive amount of workload, but for making the life more meaningful to me.

I express my sincere gratitude to my thesis supervisor, Ali Rana Atılgan for his endless guidance in completion of this project. I believe that I have learned a lot from him.

I'm also thankful for Güvenç Şahin and Ahmet Kutsi Nircan's help and suggestions throughout the evolution of this work.

I also thank to my friends who always support and assist me for years.

THE EFFECTS OF REDUNDANCY AND INFORMATION MANIPULATION ON TRAFFIC NETWORKS

Berk Özel

Industrial Engineering, Master's Thesis, 2014

Thesis Supervisor: Prof. Dr. Ali Rana Atılgan

Keywords: Urban Traffic Networks, Congestion, MinMax Driven Path, Agent Based, Adaptivity

Abstract

Traffic congestion is one of the most frequently encountered problems in real life. It is not only a scientific concern of scholars, but also an inevitable issue for most of the individuals living in urban areas. Since every driver in traffic networks tries to minimize own journey length, and volume of the traffic prevents coordination between individuals, a cooperative behavior will not be provided spontaneously in order to decrease the total cost of the network and the time spent on traffic jams. In order to perceive the effects of cooperative behavior, we develop an agent based traffic application, in which adaptive agents are able to receive traffic information and have different path selection strategies, in order to decrease own journey lengths. We lead them to a cooperative behavior by manipulating the traffic information they receive. Also, by constructing a redundant road to the network, we conceive the importance of the adaptivity to varying information. Moreover, we analyze network topologies of Scale Free, Random, and Small World networks to evaluate the compatibility as traffic networks. Then we try to create fair traffic networks from the network topologies above, in which the selfish behaviors of non adaptive drivers causes less congestion and total journey lengths, by road closures. By doing these experiments an analyses, we obtain a deeper perception about the importance of adaptivity, information retrieval, topology, and redundancy for traffic networks.

ARTIK YOLLARIN VE BİLGİ MANİPÜLASYONUNUN

TRAFİK AĞLARI ÜZERİNDEKİ ETKİLERİ

Berk Özel

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2014

Tez Danışmanı: Prof. Dr. Ali Rana Atılgan

Anahtar Kelimeler: Trafik Ağları, Tıkanıklık, MinMax Güdümlü Yol, Vekil Tabanlı, Uyumluluk

Özet

Trafik tıkanıklığı, gerçek hayatta en sık karşılaşılan problemlerden biridir. Bu problem, akademik çevreleri yakından ilgilendirmesinin yanında, kentlerde yaşayan birçok insan için de kaçınılmaz bir sorundur. Trafikteki her sürücü kendi yol uzunluğunu enküçüklemek isterken trafiğin hacmi, insanların, tıkanıklığı azaltmak amacıyla birbiri ile koordine olmasına engel teşkil eder. Bu nedenle, toplam maliyeti ve trafikte geçirilen zamanı azaltmak için kendiliğinden gelişen koordineli bir hareket mümkün olmamaktadır. Bu çalışmada, koordinasyonun etkilerini anlamak amacıyla, vekil tabanlı bir trafik uygulaması geliştirilmiştir. Bu uygulamadaki uyarlanabilir vekiller trafik bilgisini alabilmekte ve yol uzunluklarını azaltmak amacıyla farklı yol stratejilerinin arasından seçim yapabilmektedirler. Koordineli bir hareketin sağlanması için bu vekillerin aldıkları trafik bilgisi manipüle edilmektedir. Ayrıca, ağa fazladan bir yol eklenerek, değişen bilgiye olan uyumluluğun önemi ortaya çıkarılmaktadır. Bununla beraber, Ölçeksiz, Rastgele ve Küçük Dünya ağları ve bu ağların trafik ağları olarak uygunluğu analiz edilmiştir. Bu ağ topolojilerinden faydalanılarak, bazı yolları kapatmak suretiyle, uyumlu olmayan sürücülerin bencil davranışlarının daha az tıkanıklığa ve daha kısa yol uzunluğuna sebep olduğu adil trafik ağları üretilmiştir. Bu analizlerin sonucunda, trafik ağlarındaki uyumluluk, bilgi edinme, topoloji ve artık yolların önemi daha derin bir şekilde anlaşılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Being a resident of the second most congested city of Europe [6], we believe that Istanbul provided us the motivation to progress on this academic study. Figures exist in *Appendix A* show Istanbul's traffic ranking compared to the other European cities and a sample path generation experiment between two points in the city for a morning commuter. We urge you to take a look at those figures to realize the reality of the problem before reading this thesis.

The volume of the vehicles on the roads are increasing continuously especially in the urban areas since the population of the cities is growing continuously. However, owing to GPS devices and smart phones, individuals are able to see the traffic densities on the possible paths to their destination point. Continuous information retrieval via electronic devices provides drivers the adaptivity to avoid congestion.

The traffic assignment problem has evolved due to the technological changes [15]. Different solution approaches have been introduced ranging from mathematical programming [12] [10] to simulation based models where the route choice behavior of the drivers is a significant factor [17] [9]. In order to reveal the adaption utility of the drivers, also some important agent based traffic applications are simulated recently [1] [21].

Furthermore, the topology of the road networks is also effective on the congestion behavior [31]. Road networks of some cities in real life have very similar features with the prominent networks such as Random and Scale Free networks [11]. Road constructions and closures may have unpredictable consequences since they change the topology of the network [8].

The evolution of road networks is not generally under the control of the policy makers. Since the road networks alter with an evolutionary way [18]. The resulting networks may not be efficient for congestion. The inefficiency results with the congestion on the road pieces. Heterogeneity of the road pieces in terms of delay, or congestion reveals another cost measure for the drivers. [14]

In the light of these studies, we develop a discrete time agent based application in order to reveal the characteristics of the traffic flow, in which the agents are responding to congestion by changing their path selection methods. The agents receive instant traffic information and can switch between different path selection methods along the way to the destination.

Since we come up with a result that the non cooperative behavior of the agent causes traffic congestion, we deceive some of the agents by changing the traffic information they receive. Then, we test the application on Braess Network, which is famous for having counterintuitive results.

We also analyzed the characteristics of Random, Scale Free, and Small World Networks to understand which one of them is more suitable to be a traffic network that is resistant to congestion

Last but not the least; we suggested the ratio of fair*n*ess for the networks by using average path lengths. This ratio indicates the robustness of the network against the selfish behavior of the drivers which results with congestion.

Chapter 2

Problem Description

The traffic congestion for road networks is a well known problem which we encounter in real life frequently. The main purpose of this thesis is to make suggestions for decreasing the traffic congestion in real life networks. However, replicating the traffic congestion problems in a generic way in order to suggest solutions is difficult, since it is dependent on the characteristics of the networks, the features of the vehicle flows, and countless measures which may be particular to the network. We try to deal with the real life traffic congestion problem by dividing it into two sub problems as the congestion problem on toy networks, and the congestion problem on Large Networks.

2.1 Toy Networks

The toy networks that we work on are the subnetworks of real life networks. The first network topology is called the Bridge Network, which can also be encountered in the Istanbul road network depicted in Figure 4.1. The other network topology is called the Braess Network which was analyzed by Dietrich Braess in 1968. We do the analyses on toy networks assuming that all of the residents of the networks are congestion adaptive agents, who can determine new paths according to varying congestion.

2.2 Large Networks

We analyzed the traffic congestion issue on Scale Free and Random Networks, which are known as having similar characteristics of real life road networks. The road networks of Venice is one of those which resembles to Scale Free Networks [16], also have some of the Small World properties. Moreover, Dresden's road network is more likely to be a Random Network [11]. These examples can be expanded with referring Sardinia Region which also has features of Random Network [7]. We analyze large networks and perceive that existence of some redundant roads may cause congestion on the network. Then we suggest a road closure method, which may be helpful to decrease the congestion caused by the selfish drivers.

Chapter 3

Congestion Adaptive Agent Based Application

We develop an application for simulating traffic networks. The agents represent drivers of the vehicles on the network. Agents are able to retrieve instant traffic information and decide the path to their destinations considering the information received.

3.1. Parameters and Assumptions of the Application

Graph: We use directed graphs which are composed of nodes and arcs. Each arc has a weight of 1 initially. Arc weights represent road congestion, and the initial congestion at each arc also implies that there is no congestion initially and all nodes connected with an arc have the same physical distances between each other. The weight 1 is the minimum possible congestion; the maximum is limited with the road capacity.

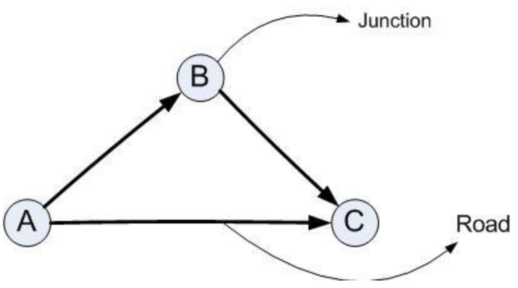
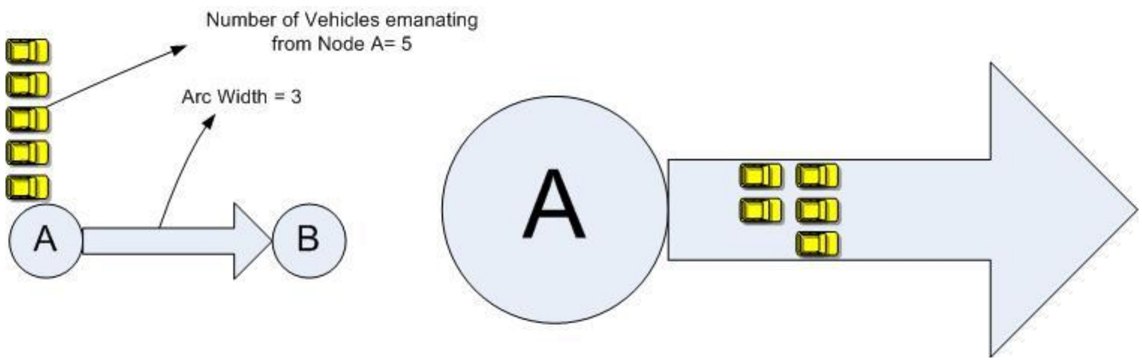


Figure 3.1. A simple network illustrating the roads and junctions is shown above. Vehicles can only travel through the direction of the arrows

Nodes represent junctions and arcs represent one-way roads between junctions.

Roads: Road pieces are represented by arcs. Initially every arc has a weight of one, which means passing a road piece will only take one unit of time for a vehicle leaving the node from which the arc emanates. Except for the head and tail nodes, an arc has two more parameters. Arc width represents number of lanes the road consists of, and Arc Capacity represents maximum length or maximum congestion of the road.

Arc Width: Arc width represents the maximum number of vehicles on a row. There may be more vehicles than the arc width emanating from the node. In this case remaining vehicles will constitute another row.



a) At Time t, 5 vehicles leaving Node A   b) At time t+1, 5 vehicles on the arc

Figure 3.2. Five Vehicles are trying to leave Node A through Node B. Since the number of lanes is less than the number of vehicles, two of them constitute the second row.

The first row of an arc is static. Even if the first row of an arc is full, partially filled, or empty; it will take one unit time for a vehicle to pass the arc considering there are no more rows of vehicles after the first row.

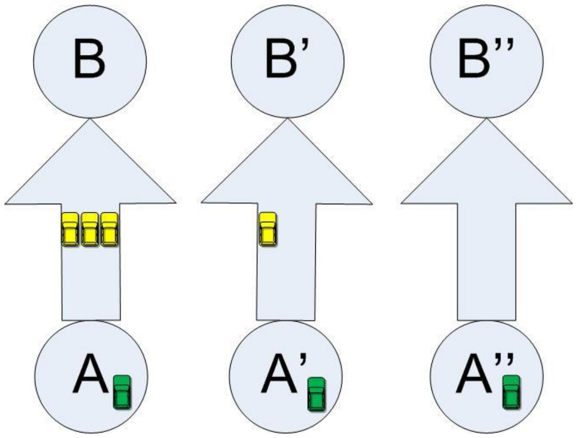


Figure 3.3. When the green vehicles enter the arcs, the vehicles on the first row of the arc will also leave. The time needed to pass the arc between Node A and Node B, Node A' and Node B', Node A'' and Node B'' is the same for the green vehicles leaving the node A, A', and A'' because of this reason.

Arc Capacity: The number of rows on a road which represents the congestion should be limited by a number. Otherwise the congestion on a road piece doesn't affect the road behind. According to our assumption, an arc doesn't accept more vehicles, if it reaches its capacity.

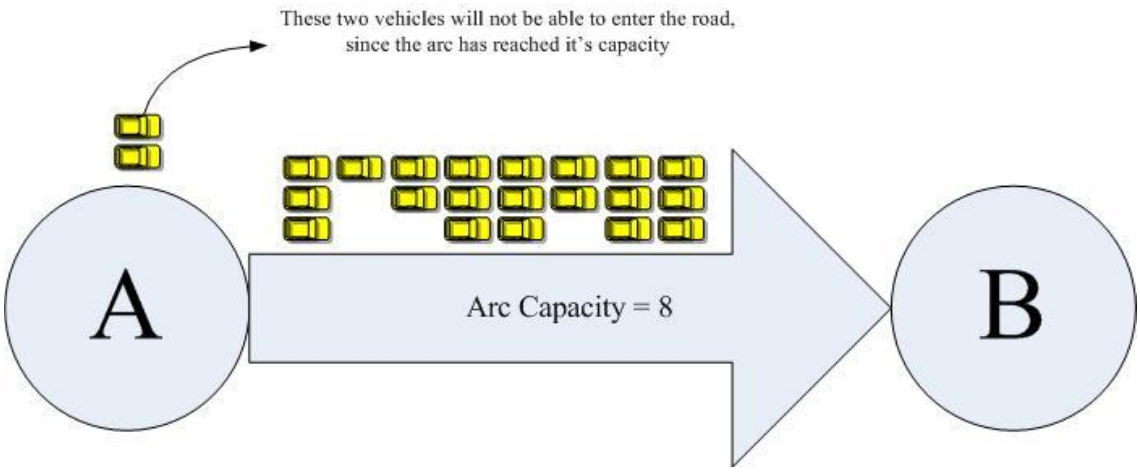
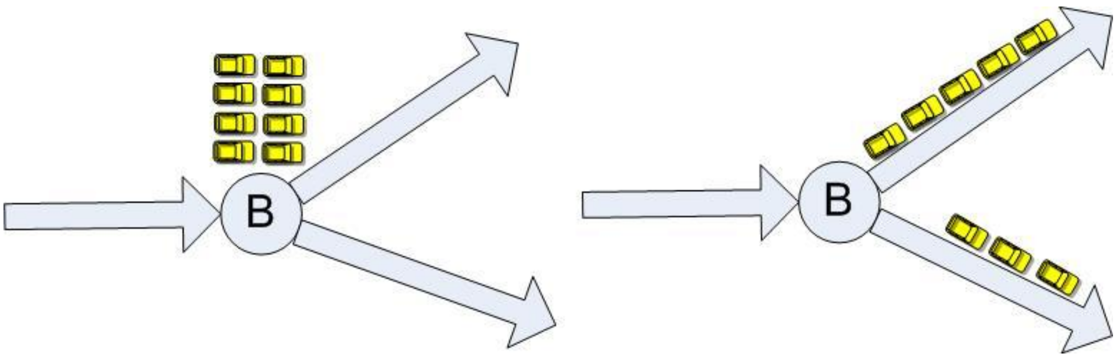


Figure 3.4. This figure illustrates congestion on the arc from Node A to Node B. Two vehicles at Node A are unable to move forward, because the road has reached its maximum capacity.

It may be realized that there are some empty spots on the road, which may be filled with the vehicles behind. This might also provide free space for the vehicles waiting to leave from Node A. However this adjustment will cause some problems which make the flow illogical. This issue will be covered in more detail at “Time Assumption” section. Also, road lengths are excessively more than the number of road lanes in real life. Because of this huge difference, empty spots on the lanes become insignificant.

Junctions: Junctions are represented by nodes. There may be many incoming and outgoing roads from a junction. Junctions are essential for the application, because vehicles can change their directions, or make a new path selection decision only on the junctions. Briefly, if a vehicle has entered to a road, it is obliged to proceed until to the next junction. Departure process of the vehicles at a junction is different than at a road. All of the vehicles can leave the junction, if the road which is selected to enter has capacity. A vehicle waits at the junction until the last row of the queue in recently selected road becomes empty.

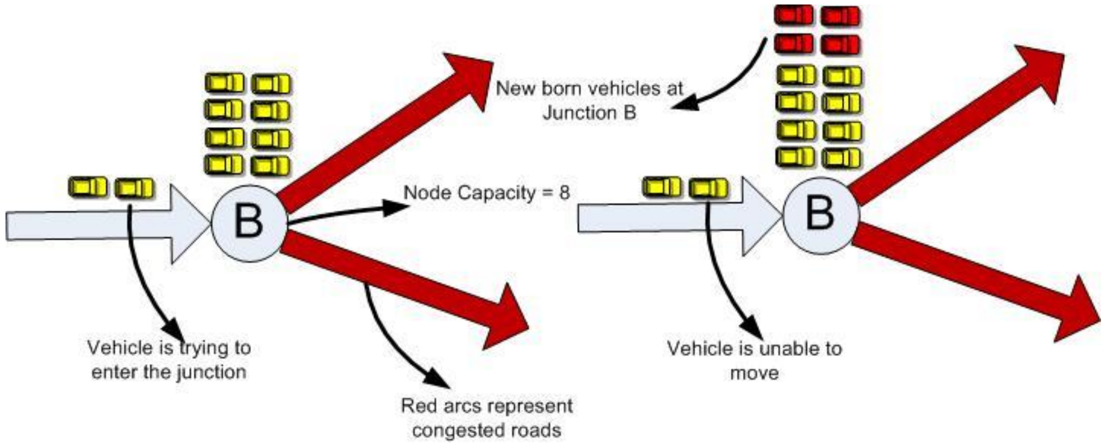


a) At time t, Undecided vehicles at the Junction

b) At time t+1, vehicles decided their path to the destination and started to move

Figure3.5. Vehicles at Junction B can decide their next move only at the junction. They can continue to their former path to the destination, or make a new path decision according to the criteria they own. None of the vehicles have any priority to wait at the junction unless the road they decided to proceed is full of its capacity.

Node Capacity*:* Junctions also have capacities, which are called as node capacity. Node capacity restricts any vehicle to enter the junction, if the junction has reached its capacity. Node capacity and arc capacity have similarities, but they are not totally same. If a node reaches to its capacity, vehicles coming from incoming roads are unable to enter the node. However, new vehicles can be born at the junction and make the junction over capacitated. For the networks with multi-lane roads, even if there is one empty spot at the junction, the junction accepts the whole row of vehicles. This feature provides elasticity to the junctions.

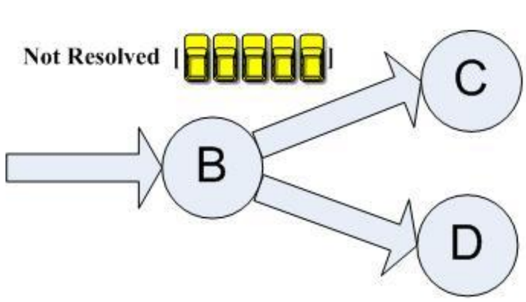


a) Vehicle wants to enter a congested junction

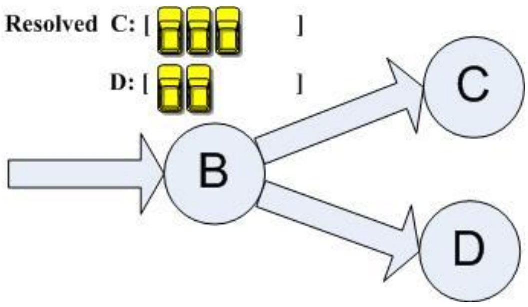
b) The vehicle coming from an arc is refused, yet vehicles are born at the congested junction

Figure 3.6. Junction B has a capacity of eight which is filled by vehicles. The outgoing arcs of Junction B are also congested, which means none of the cars at Junction B can leave and make space for incoming vehicles. For this situation vehicle coming to Junction B is refused. However, four vehicles are born and make the junction over capacitated

Junctions also have two kinds of lists. The first one is called “Not Resolved Vehicles” list. This list includes the vehicles which have just entered to the junction from a road, or the vehicles that are unable to leave the node because the first road of its selected path is congested. The rest of the members of the list are the vehicles that are ready to leave the junction. Those vehicles are temporary members of the list in a time period. They will be transferred to another list when they decide from which road they will leave the junction. The other list is “Resolved Vehicles” list. The vehicles that have not entered to the junction are assigned to that list according to their departure road selection. This list is actually a map of lists. Each list of the map has a key value indicating the name of the emanating road.



a) The vehicles not decided their departure roads



b) The vehicles exist at the resolved list e after deciding the departure road

Figure 3.7. There are five vehicles on Junction B's Not Resolved Vehicle List. Junction B's Resolved list is empty before the vehicles are processed. Since the vehicles are processed and decide their path to their destinations, they are assigned to the related sub lists according to the selected path's second junction. The selected path's first junction is the node that they are leaving.

The congestion on the network is directly related with the arc and node capacities. When a directed arc or a node reaches its capacity, it prevents the incoming flow to move forward. If the vehicles on the arc or node continue waiting for the following discrete time steps, the congestion will expand towards the back.

Vehicles: Vehicles can be born at any junction and any time, but they cannot be born at the roads. Vehicles only have source, destination and path selection method initially. They have constant speed, if the path they follow does not get congested at any time. If a vehicle comes up with a congested road or a congested junction, it stops. It is unable to move to the next row on the road until the vehicle or the vehicles in front of it move. Vehicles are unable to pass any other vehicle ahead, or be passed by another vehicle neither on a road nor at a junction. Once a vehicle departs through a road, it cannot change the direction, or set a new path to its destination until it reaches the head node of the road. Even in congested traffic, vehicles cannot fill the empty spots in front of them. Because of these features, the movement of a vehicle is similar with "First-in First out Principle".

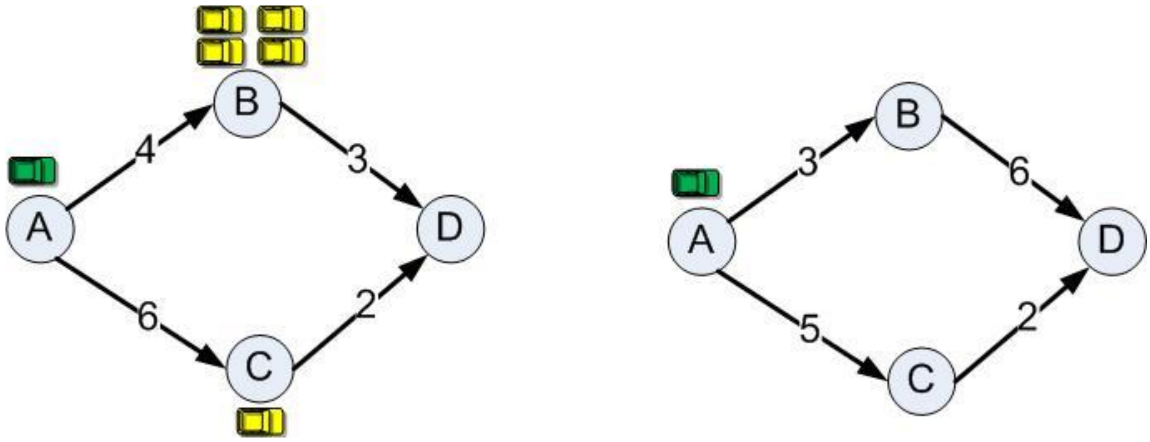
Time: Time is discretely increasing in the application.

At each time step, a vehicle can

- Be born at any junction, regardless of the capacity restriction.
- Move to the next row on the road.

- Make a path decision through its destination and move.
- Just make a path decision and stop, which means that its first road piece of selected path is congested.
- Just stop, because the vehicles ahead cannot move due to a congested junction.
- Leave a road and enter a junction.
- Not enter a junction, due to full capacity of the junction
- Stuck at a junction due to a fully loaded road.

We assume that each vehicle can see the traffic information at any time and decide its strategy to arrive the destination point. However, because of the information delay and for not providing an advantage to the early processed vehicles, we display traffic information of 1 preceding time unit to the vehicles.



a) The state of vehicles at time t-1

b) All of the vehicles are processed except for the green vehicle at time t

Figure 3.8. Green vehicle at time t will decide by its path selection strategy regarding to the arc weights at time t-1. By applying time assumption we prevent an unequal situation between vehicles which cannot be processed at the same moment in a time step.

Path Selection: Each vehicle can decide, keep or change its path to the destination point at any time, unless it is on a road or has just arrived to a junction. There are three possible path selection algorithms a vehicle can use.

Shortest Driven Path: Shortest Driven path algorithm finds the shortest unweighted path from the source node to the destination node. This algorithm ignores the congestion and decides the shortest path according to the physical distances. In other words it selects the path, which has the least number of hops. If there are more than one Shortest Driven paths, which have equal number of hops, the algorithm compares the sum of arc weights of each path, and selects the path which has the lowest weight.

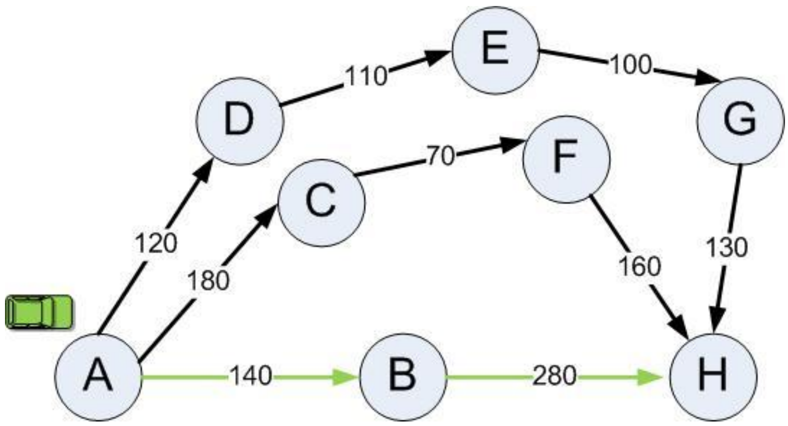


Figure 3.9. If the green vehicle, which is departing from junction A through junction H, has a path selection method of “Shortest Driven Path”, it will follow the path A-B-H. The number of hops of the selected path is two. The most congested arc is the arc between junction B and junction H.. Also the sum of arc weights of the selected path is 420 units.

Regardless of the effect of traffic density, “Shortest Driven Path” method calculates physical distances from one node to another, and then selects the minimum distanced path.

MinMax Driven Path: The main purpose of MinMax Driven path is to avoid most congested roads. This algorithm compares the highest weighted arcs of all possible paths between the source node and the destination node. It picks the path of which the highest weighted arc is the lowest of all. For the tie breaking rule, it takes the lowest number of hopped path, if there are many.

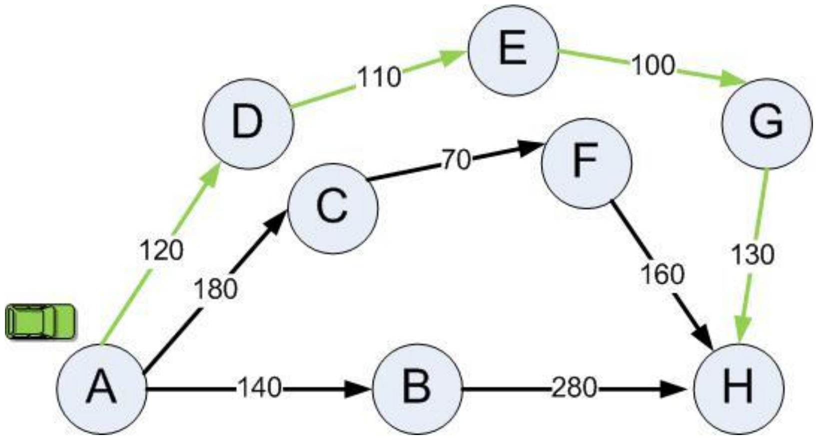


Figure 3.10. The vehicle having "MinMax Driven" path selection decision will select the path through junctions A-D-E-G-H. This method compares the most congested roads of the possible paths. G-H =130, A-C =180, B-H = 280. According to the algorithm, the green car will select the path includes the road G-H, since it has the lowest traffic density among the most congested road pieces of all paths.

Combined Path: This path selection method is the combination of two algorithms above. It implements the Shortest Driven Path first, and then stores the path. If the path is congested according to predetermined criteria, it runs the MinMax algorithm which tries to minimize the highest congestion. The important point of this method is that; the congestion becomes significant for the vehicle in order to change its path selection method. Defining a congestion threshold value will be a simple measure to decide.

Adaptiveness: The path selection method of a vehicle is adaptive to the changes on the network. At the beginning of each time step, vehicles receive the traffic information and make their decision regarding to that information. At each junction through the destination, vehicles can change their selected paths. Moreover, if a vehicle is unable to leave the junction due to congestion at the emanating road, it can make another path selection decision at the next time step. This property provides vehicles to be adaptive to changes on the traffic density, which is also acceptable in reality.

3.2 The Algorithm

According to the parameters, assumptions, and path selection methods described in Section 1.2., the algorithm of the application works as follows:

Initially, no vehicles exist on the network. Thus the arc weight of each road is one, which is the minimum possible value. Arc width, arc capacity and node capacity are specified at the beginning. Also, the life time of the application is specified.

At each time “t”, vehicles are generated on the specified junctions, which are decided by the user. Then, new born vehicles generated at time “t”, are added to vehicle list, which also includes other vehicles generated before time “t”. Before any movement happens on the network, all vehicles receive the traffic information. The flow at time “t” starts with draining of the roads. Every vehicle at the first row of a road leaves the road as long as the junction to which they are moving has at least one empty spot. The vehicles, which arrive at a junction, are placed into the “not resolved vehicles” list. This list consists of vehicles that have not decided which road they are going to leave the junction from. Recently placed vehicles are also labeled indicating that they have just arrived at the junction in order to prevent zipping. Because of draining of the roads, the free capacity of the road increases by one. Vehicles are not informed about this change on the density of the road until the next time step. If there are not any empty spots at the junction which the road is reaching, neither of the vehicles on that road moves.

After all of the roads are processed, vehicles at the junctions start to leave. For a vehicle to leave the junction, it has to decide the path to follow first. We process the vehicles, which are not labeled as “just arrived”, at the junction’s “not resolved list” sequentially. Path decision of a vehicle depends on the path selection algorithm it has.

If its method is “Shortest Driven Path”, the vehicle calculates the number of hops it needs to reach its destination. There may be alternate roads with the same number of hops. To make its decision wiser, the vehicle makes a modified Depth First Search on the map. The depth of the search is limited with the number of hops calculated earlier. When the vehicle finds a path to the destination with a specified depth, it stores the path and the sum of weights in order to compare with other equal hopped paths. At the end of the Shortest Driven path algorithm, the vehicle finds the less total congested path among the shortest physical distanced alternatives.

If the path selection method is “MinMax Driven Path”, all of the paths to the destination are revealed with a modified Depth First Search. There is no depth limitation on this modification. When it finds a path to the destination, it stores the path and the most congested arc weight of the path. If the most congested arc of recently discovered path is less than the one that we stored earlier, we replace the stored path with the recent one. If the most congested arcs of two paths have the same value, the vehicle selects the one which has less number of hops.

There may be some vehicles which have “Combined Path” method. These vehicles select their path according to the “Shortest Driven Path” algorithm initially. Once they find their Shortest Driven path, they examine the arc weights. If the weight of any arc on the Shortest Driven path is more than the congestion threshold which is determined earlier, “MinMax” algorithm is applied.

Once a vehicle decides its path, it is removed from the “not resolved vehicles” list and added to “resolved vehicles” list, in which each vehicle is placed into a sub list specifying the road from which the vehicle leaves the junction. The vehicles with “just arrived” labels are also processed by eradicating their labels while the vehicles which are not labeled are transferred to the “resolved vehicles” list.

Vehicles at the “resolved list” leave the junction if there is enough capacity on the arc they will use to exit. The vehicles, which are unable to leave the junction because of their selected exit is reached to the capacity, are sent back to the “not resolved vehicles” list for a new path selection at the next time step “t”. The flow spills out of a junction does not affect decision mechanism of other vehicles on the other junctions, since this information will be received at the start of next time period “t+1”. If a vehicle arrives its destination point, it provides its journey information such as the travel time, and the time it was stuck at a junction or a road to the other vehicles. After providing the information, the vehicle disappears.

This procedure is repeated until the time counter reaches to the end of life. When the program terminates we get the distributions of travel times of the vehicles and the distributions of the times where the vehicles are blocked. We will analyze this information and see the effects of information manipulation later.

Pseudo codes of the algorithm can be found in Appendix C.

Chapter 4

Computational Experiments

Some computational experiments are performed on two different network topologies in this chapter.

4.1. Bridge Network

The first experiment is performed on the topology which we frequently come up with. They are the networks which have a link behaves like a bridge. This link connects two major parts of the network and mostly there exist more than one incoming links to the bridge. To illustrate the bridge on a network, we can consider the traffic network of Istanbul.

Figure 4.1. A part of Istanbul Network. This directed graph has an alternate path to the bridge. We will run our simulations on a small network similar to this topology.

The small network we run the simulations is similar to the network above. This structure is common on the networks which have bridge like structures. We will try to see the effects of evolutionary construction of shortcuts or alternative roads which are longer than the present road.

The exact network we use on the simulations is below.

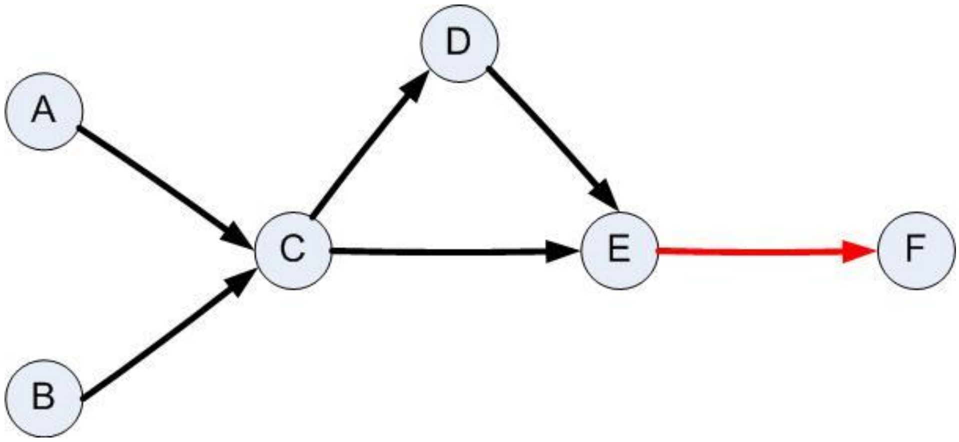


Figure 4.2. The sample network has six nodes. The red link represents the bridge, Node A and Node B are the source nodes where the vehicles are bo*rn* and depart from. Node F is the destination node. The link widths are 1 for all links. C-D-E path is the alternate path constructed to decrease the traffic congestion.

4.1.1 Empty Bridge Network

The link widths are 'one' for all links. This is done to simplify the process and also prevent misleading outcomes caused by the assumption "A vehicle cannot fill the gap ahead even if the traffic is congested." The life of the application in the analysis is 20 time units. If a vehicle can't arrive the destination point for the next 20 time units after the first vehicle was born, it is labeled as "Unable to Arrive", and a predetermined penalty score is added to the total travel time.

All of the agents are adaptive to the changes on the network. However, the topology of the network lets the agents change the path selection at 'Node C' for the last time. Maximum number of vehicles that can reach to 'Node F', which is the destination point, is 15. This value is stable since we send the agents to an empty network for now. Even if we start sending the agents from the start of the lifetime, it takes five steps for the first born vehicle to reach the destination. Although the maximum number of vehicles that can reach to the destination is fixed, we are trying to avoid congestion on the critical link (the link between ‘Node E’ and ‘Node F’), and also to decrease the total journey length of vehicles by changing the frequency of arising vehicles.

We are using three type of frequencies for generating agents, which we call this process ‘Sending a Pulse’ from now on. In all of the experiments we have made on the Empty Bridge Network, we are going to send one pulse from ‘Node A’, and one pulse from ‘Node B’. We will postpone the pulse generated by ‘Node B’ step by step and observe the results.

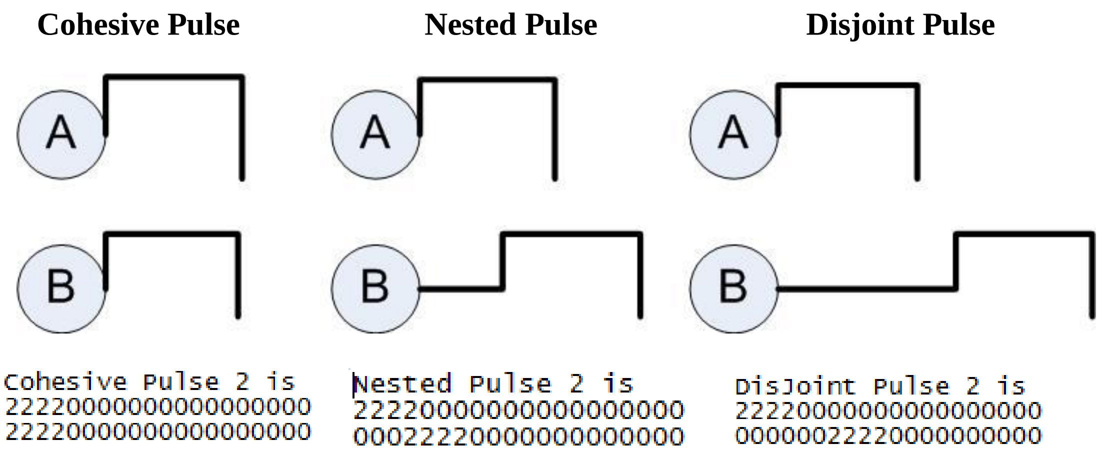


Figure 4.3. For each type of pulse, the magnitude and the length of the pulse is the same for Node A. However, the departure of the vehicles at Node B is postponed from Cohesive to Disjoint Pulse. For each figure representing a pulse type, 16 vehicles has been departed in total.

All of the vehicles are sent from the source nodes can have either ‘Shortest Driven Path’ or ‘MinMax Driven path’ path selection method. They are free to change the path they will follow according to the path selection method they have.

The table below shows the statistics of vehicles sent with different pulses.

Shortest Driven Decision	Disjoint	Nested	Cohesive
Vehicles Unable to Arrive	1	1	1
Decision Direct	1	1	1
Decision Alternate	0	0	0
Vehicles Able to Arrive	15	15	15
Decision Direct	15	15	15
Decision Alternate	0	0	0
Total Journey Length	132 + 20	150 + 20	173 + 20
Blocked Time at Critical	0	0	0
Blocked Vehicles at Critical	0	0	0
Blocked Vehicles at Previous Nodes	0	0	16

Table 4.1. Vehicles which have Shortest Driven path method will follow A-C-E-F or B-C-E-F paths. Twenty penalty values are applied for the vehicles which are unable to arrive to the destination point. “Blocked vehicles at critical” and “blocked time at critical” are the sum of vehicles blocked at the blocked times and total number of blocked times. For example if two vehicles has been blocked at time t and three vehicles are blocked at time t+1 and no more vehicles are blocked at the other times, blocked time at critical is two, and blocked vehicles at critical is five.

For the disjoint and nested pulses, the Shortest Driven path is able to handle the flow. Congestion does not occur at any time. Total journey length of the nested pulse is longer than the disjoint pulse, which is an expected circumstance. However, when we send cohesive pulse, not only the total journey length increases, but also congestion occurs on the links before the critical link. The reason for this situation is illustrated in the figure below.

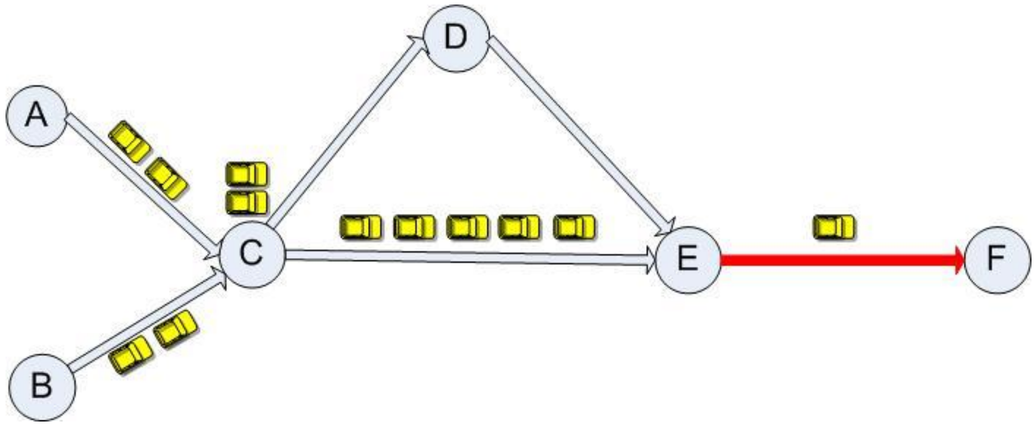


Figure 4.4. All of the arcs and nodes have capacity of 5. 'Total blocked vehicles at previous nodes' does not imply distinct number of vehicles. If the same vehicle is congested at the same node for more than one time step, this variable increases more than once.

There is only one link used to reach 'Node E', since any path including C-D-E is never used by the vehicles which have Shortest Driven Path method. Both the input and the output of 'Node E' is one, which makes the link between Node E and Node F never becomes congested. The congestion occurs at link between Node C and Node E. If there is not enough idle time between the sources, an extra vehicle appears between the incoming and outgoing vehicles of Node C. This extra vehicle increases the density of the link between Node C and Node E, until this link reaches its capacity. Vehicles will be congested at Node C, after this link has reached its capacity. Congestion also increases the total journey lengths.

All of the vehicles are sent again with the same pulse frequencies, but this time we impose them to apply MinMax Driven method. We named X-C-E-F path, the Direct Path, and X-C-D-E-F path the alternate path, where X stands for 'Node A' or 'Node B'.

Min-Max Decision	Disjoint	Nested	Cohesive
Vehicles Unable to Arrive	1	1	1
Decision Direct	1	1	1
Decision Alternate	0	0	0
Vehicles Able to Arrive	15	15	15
Decision Direct	13	10	9
Decision Alternate	2	5	6
Total Journey Length	132+20	153+20	174+20
Blocked Time at Critical	0	1	4
Blocked Vehicles at Critical	0	1	6

Table 4.2. There is a small increase in Total Journey Length of Nested and Cohesive Pulses comparing with the state where each vehicle uses Shortest Driven path. MinMax Driven method vanish the congestion before the critical link, but on the frequent different pulses, critical link gets congested with the vehicles which use MinMax Driven method.

The figure below clarifies why the critical link gets congested.

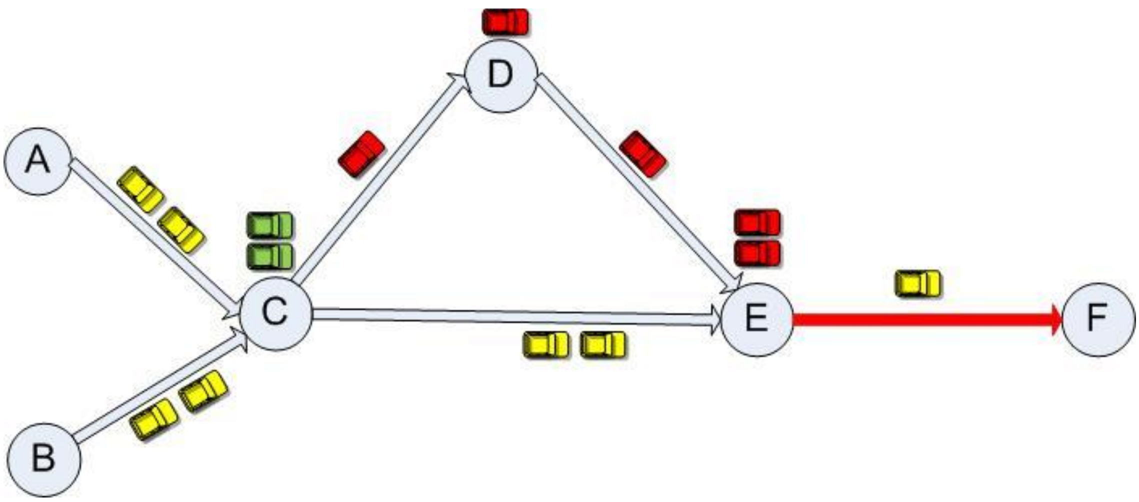


Figure 4.5. The vehicles at Node C will pick the alternate path according to MinMax algorithm. There are going to be more than one vehicle at Node E at least for the next five time steps. These additional vehicles will cause congestion at the link between Node E and Node F.

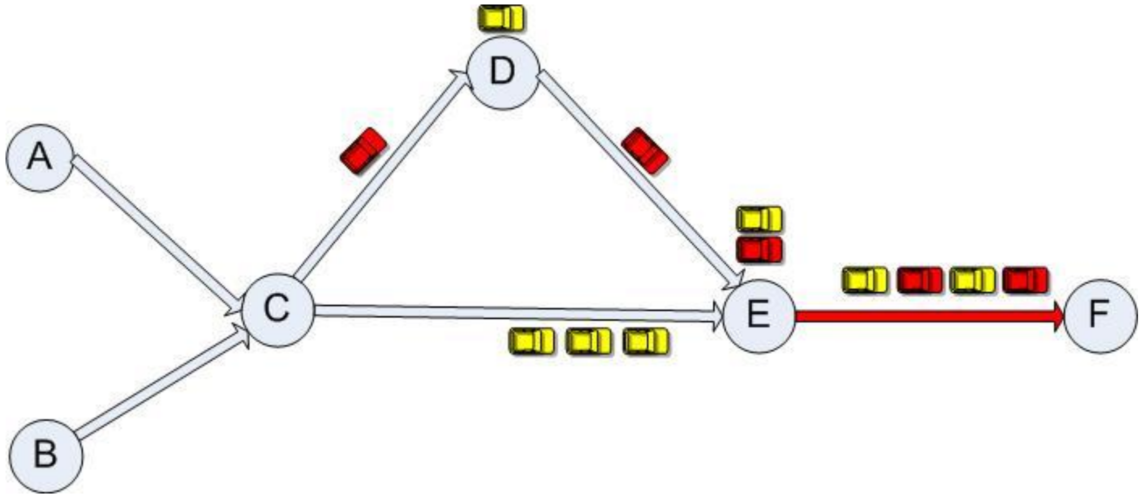


Figure 4.6. The Critical Link is congested after four iterations. The vehicles that are using the alter*rn*ate path cause an overload at Node E. Since the remaining capacity of the link between Node E and Node F is one, the link becomes congested.

The solution will be to manipulate the link congestion information for the vehicles which will make the critical link become congested later. The number of vehicles on the link between ‘Node D’ and ‘Node E’ will always be one, since no vehicles are born at ‘Node D’, according to the inflow and outflow equilibrium. This link may be the link whose density information will be manipulated. Similarly we could choose the link between ‘Node C’ and ‘Node E’. The algorithm for manipulation differs for different topologies. The basic idea behind the manipulation algorithm is: If the number of vehicles on the alternate path is reached to the critical arcs idle capacity, manipulate the pre chosen arc’s density information for the vehicles at the junction, where they will give the decision for selecting the direct or the alternate path.

The manipulation algorithm should be applied to the situation at Figure 4.5., to prevent congestion. At the situation this figure illustrates, the algorithm will manipulate the density information of the link between ‘Node D’ and ‘Node E’, showing it as congested to the vehicles at ‘Node C’. Consequently these vehicles will follow the direct path, since the MinMax algorithm will guide them to avoid the link between ‘Node D’ and ‘Node E’. The algorithm will check the potential congestion on the critical link and prevent it with doing the same operations for the next time steps.

Manipulation algorithm is just applied to the nested and the cohesive pulse, since the disjoint pulse does not cause any congestion.

Decision Under Manipulation	Nested	Cohesive
Vehicles Unable to Arrive	1	1
Decision Direct	1	1
Decision Alternate	0	0
Vehicles Able to Arrive	15	15
Decision Direct	11	11
Decision Alternate	4	4
Total Journey Length	153+20	174+20
Blocked Time at Critical	0	0
Blocked Vehicles at Critical	0	0
Blocked Vehicles at Previous Nodes	0	0

Table 4.3. The manipulation algorithm works fine. Neither at the critical link, nor at the other arcs congestion occurs. I also want to point out that total journey lengths for both pulses do not change with the manipulation algorithm.

According to the results of the simple network we studied above, the manipulation on density information of the vehicles can prevent congestion on the network. To make a further analysis we are going to use a partially loaded network and use more than three pulses with different magnitude and lengths.

4.1.2 Partially Loaded Bridge Network

For partially loaded bridge network, we also use the same topology with the empty bridge network. We load the network with different number of vehicles, leave different idle times after loading and use different source nodes for loading the network with vehicles. The purpose of this work is to eliminate the advantage of the early commuters, which will prune the skewness of the journey length distribution. For loading the network we use all or some of the nodes from Nodes A, B, C, D as source nodes. To make the journey length distribution smoother, we try to run sufficient number of samples.

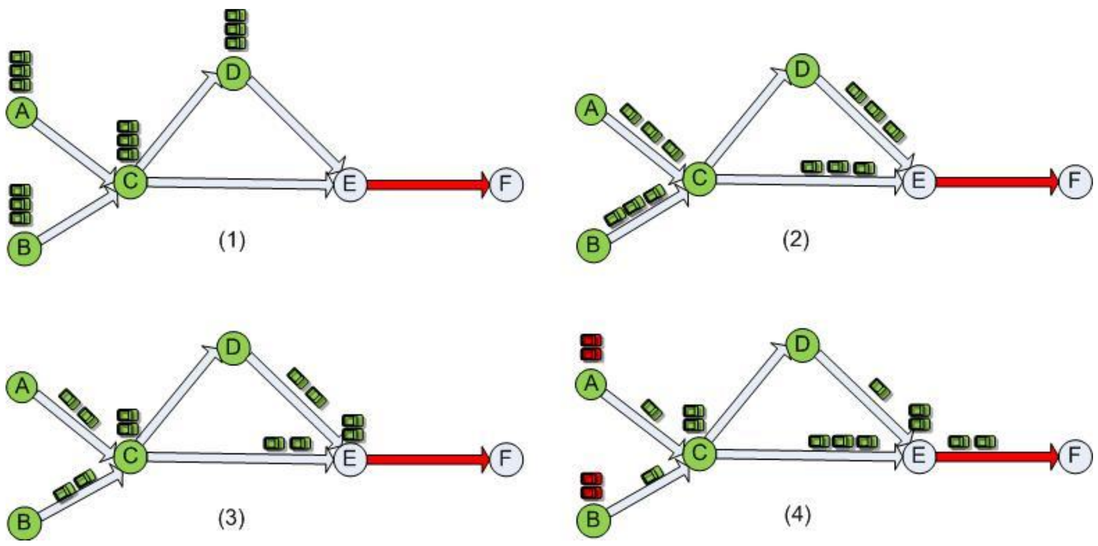


Figure 4.7 Figure above represents a partially loaded network, which is generated using four nodes as source nodes and three vehicles are born as dummy vehicles. To create more natural flow, we keep the system idle for 4 time steps. At the fourth time period a cohesive pulse is introduced to the network. Green vehicles represent the dummy vehicles and the red vehicles on the fourth sub figure represent the vehicles which are the members of the cohesive pulse.

The number of nodes generating dummy nodes, the time when the network is left idle, and the number of dummy vehicles generated initially differ for different runs. The table below shows how many different type of partially loading we have tested for the cohesive pulse.

Dummy Vehicles generated at the first 3 Nodes				Dummy Vehicles generated at the first 4 Nodes			
	1 Vehicle each Node	2 Vehicles each Node	3 Vehicles each Node		1 Vehicle each Node	2 Vehicles each Node	3 Vehicles each Node
0 Idle Time	x			0 Idle Time	x		
1 Idle Time	x			1 Idle Time	x		
2 Idle Time		x		2 Idle Time	x	x	
3 Idle Time		x	x	3 Idle Time		x	
4 Idle Time		x	x	4 Idle Time		x	x
5 Idle Time			x	5 Idle Time		x	x
6 Idle Time				6 Idle Time			x

Table 4.4. The table above shows how many different partially loaded networks were generated with different densities.

A cohesive pulse is sent to the resulting networks. The journey length and the blocked time distributions are plotted for the comparison with the same initial networks, but the vehicles of the cohesive pulse are misleaded intentionally with the manipulated road density information on the later one.

The average journey length distribution and the average blocked time distributions of 18 different initially loaded bridge networks are plotted below.

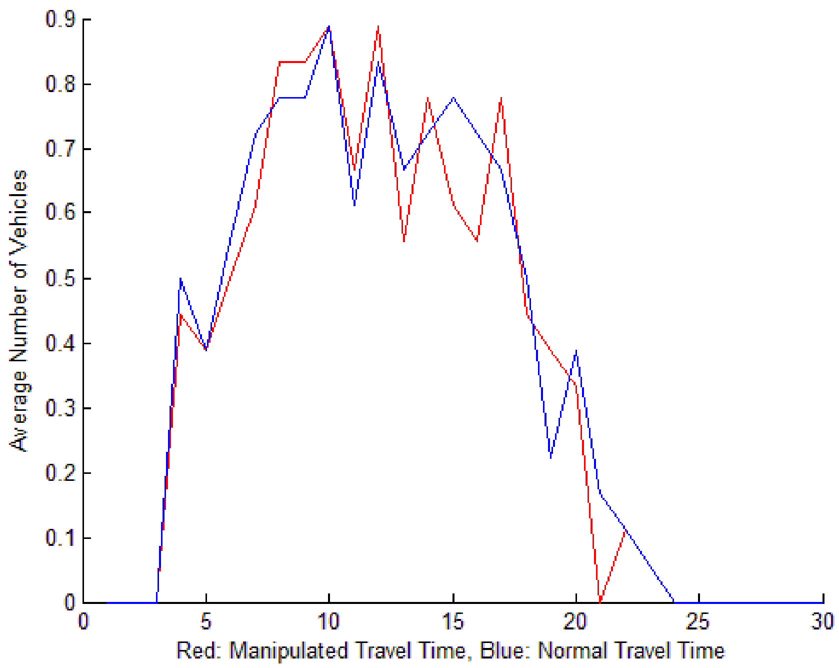


Figure 4.8. Distributions of travel times. The line drawn with blue represents the average travel time of the partially loaded bridge network and the line drawn with red represents the average travel time of the bridge network where the belonging vehicles have the manipulated information. The distributions are plotted from 18 samples where the initial density of the network varies. x axis shows the jou*rn*ey lengths of the vehicles and y axis shows, average number of vehicles that reach to the destination point with number of steps denoted by x.

When we apply traffic information manipulation algorithm to the same initially loaded networks which are also exposed to cohesive pulse, the distribution of the average travel times does not change significantly. However, the distribution of blocked times shrinks, which shows that manipulation algorithm decreases the traffic congestion on particular roads. An important consequence will be overlooked by disregarding the blocked time distribution. One can consider that, since the travel time distribution slightly changes, the information manipulation is not an effective idea for traffic networks. However, all of the vehicles on the network do not have the same source and destination points. The congestion, occurred by the majority of flow having the same source and destination, will increase the travel times of the other vehicles, which also increases the total travel time of the network.

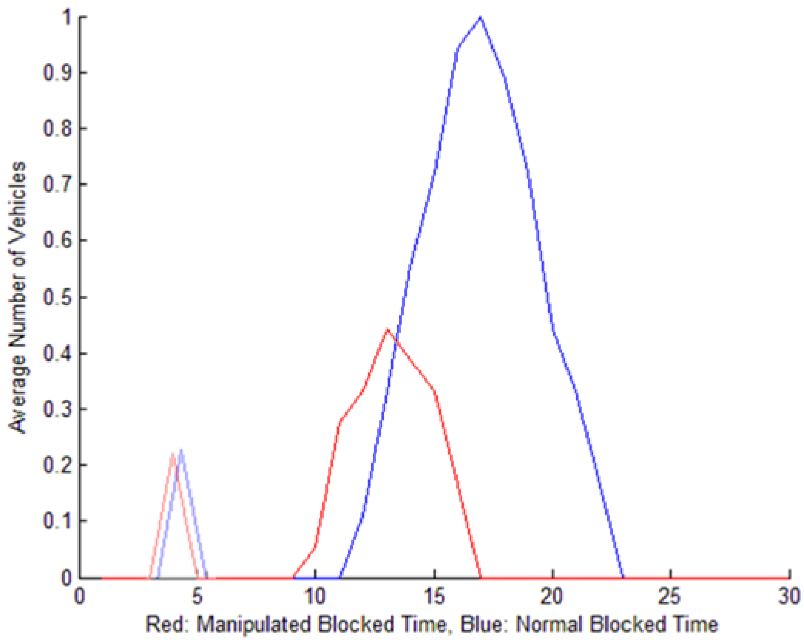


Figure 4.9. Distribution of blocked times, blue plot represents average blocked time distribution of vehicles on partially loaded bridge network, and red plot shows the average blocked time distribution of the same network of which belonging vehicles have manipulated traffic information. The distributions are acquired from 18 different initially loaded bridge networks. x axis represents the time interval of the applications lifetime in which the congestions occur, and y axis shows the number of vehicles which are blocked on average.

The information manipulation experienced on toy networks with adaptive agents, reveals an attractive consequence, which is worth studying as a future work. With the assumption we have made, it does not only cause the agents, which have manipulated information, to reach the destination point at longer travel times but also provide the other vehicles not to stuck at the traffic jam.

4.2 Braess Network

We also make an analysis of Braess Network with the application. Currently, the traffic networks are different from the traffic networks of 1968, when the Braess Theorem is suggested. The theorem represents the traffic flow as a single game, where the vehicles decide their path at the start of their journey. By this assumption the theorem claims that having a redundant road connecting two main roads increases the social cost [3] [4]. The features of the flow of Braess and of our vehicles are different. We assume that vehicles are able to receive current traffic information and be able to decide at each junction. The purpose of this simulation is; if there is still a paradoxical situation occurring on the Braess Network with constructing a redundant path to the present network?

The Braess Theorem suggests that, there can be some roads, which are able to resist the population of the traffic on it, which means, regardless the number of vehicles on the road; the vehicles are able to pass the road at a fixed time. This assumption is against to our general road assumption, so we generate a new type of road structure for the roads described at the theorem.

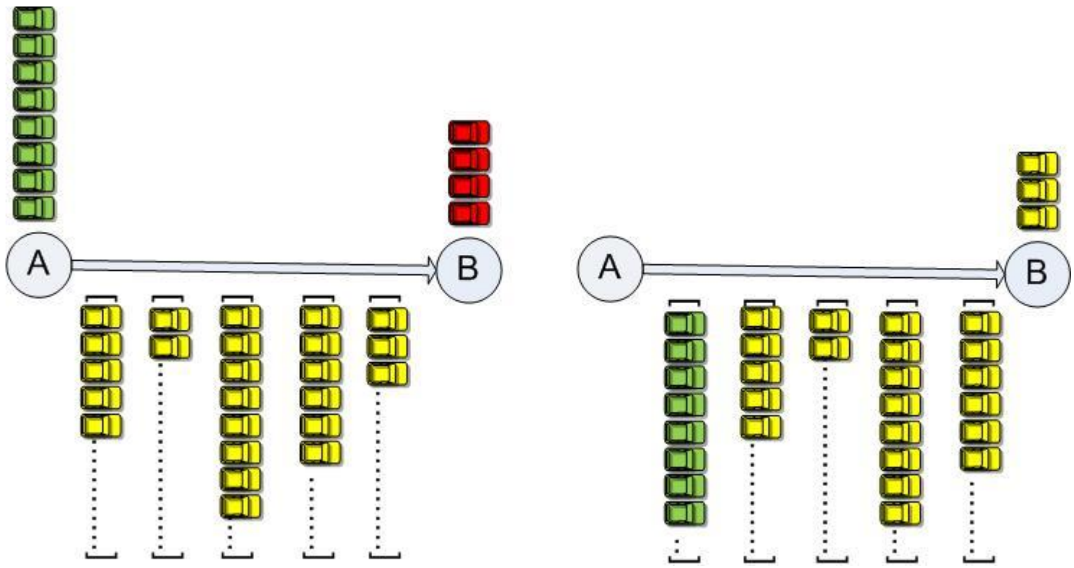


Figure 4.10. No matter the number of vehicles entering to the road, they pass it on ‘the length of the road’ steps. Green vehicles enter the road and yellow ones at the first row exit.

To satisfy congestion resistant arc assumption, we kept the arc width property of the roads infinite and kept the length of the road fixed. By doing these changes on the roads, the arc lengths stay fixed no matter how many vehicles are on that roads. Different than the bridge networks, we have four nodes on Braess Network, the road and junction capacities are limited with 10 vehicles, and the roads consist of 1 lane except for the congestion resistant arcs. Congestion resistant arcs have a length of 5. The lifetime of the application is 50. The other assumptions about nodes and arcs of the bridge networks are still valid.

Initially, the Braess Networks with and without redundant roads are shown below.

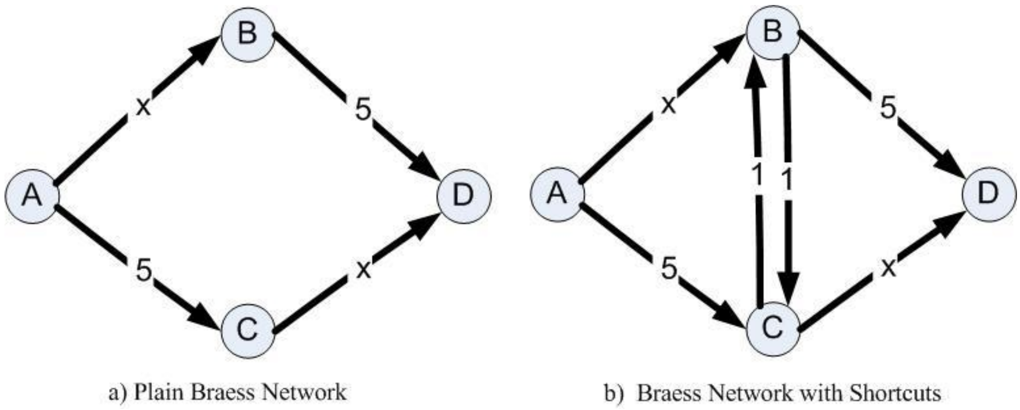
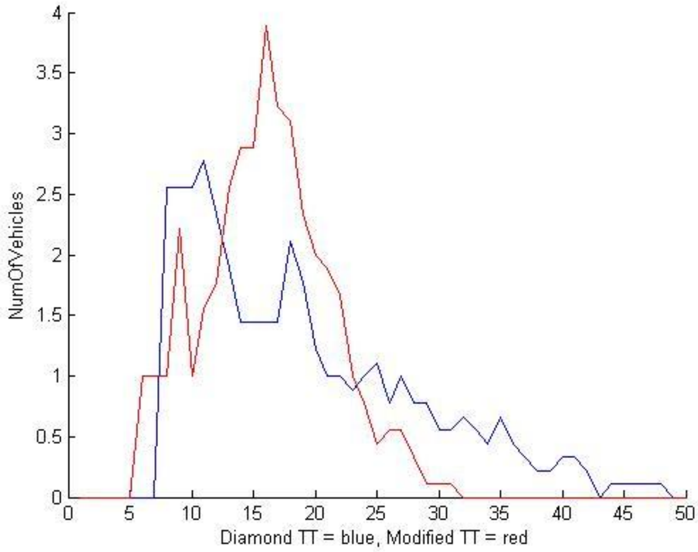


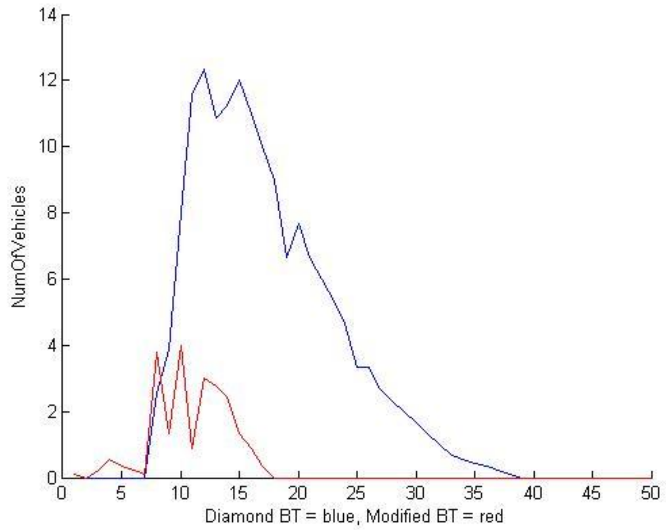
Figure 4.11. The Braess Networks with and without redundant paths. The roads which are labeled with ‘x’ are sensitive to congestion. If there are x vehicles on the road, the next vehicle entering the road will pass it on ‘x+1’ steps. However, the roads labeled with ‘5’ and ‘1’ are not sensitive to the number of vehicles on the road. All vehicles entering the road will pass the road at the labeled number of steps.

Three different types of pulses are sent again both to the ‘Plain Braess Network’ and ‘Braess Network with Shortcuts’. The magnitudes of the pulses differ between 4 and 6. However, the characteristics of the pulses are the same. The pulses are labeled as Cohesive, Nested, and Disjoint. There are not incoming arcs to Node A, which stands as the source node. The pulses appear instantly on Node A without a delay. The Node’s elastic capacity feature enables ‘Node A’ to breed the pulses of vehicles even if it is full.

There is an undirected arc between ‘Node B’ and ‘Node C’ while the original Braess Network has a directed arc from ‘Node B’ to ‘Node C’. We prefer the undirected arc to make the results more clear. The effect of the shortcut arc is depicted by showing the Total Journey Length Distribution and Blocked Time Distribution.



a) Average Travel Time Distributions



b) Average Blocked Time Distributions

Figure 4.12. The figures show the average travel time distributions and average blocked time distributions of vehicles that travel on simple Braess network and Braess network with shortcuts. The blue lines represent the vehicles that travel on the simple Braess network and the red ones represent the distribution of vehicles on Braess network with shortcuts.

Figure 4.12.a shows that with adding shortcuts to the simple Braess Network, the travel time distributions of the vehicles sent with the specified pulses bend to the left. This skew shows that more vehicles are able to arrive to the destination point faster. Moreover, Figure 4.12.b shows the blocked time distributions, where the number of vehicles blocked at a part of the network diminish significantly. These two figures

depicted above show that, the social cost decreases when the shortcuts are built to the simple Braess network. The simulation we generate has contrary results with the Braess Paradox. However, these results should not amaze us since the features of the flows and assumptions do not overlap. We can draw a conclusion with saying that the Braess Paradox will not occur on the networks, where the vehicles on it are able to receive instant traffic flow information, and enters the network gradually instead of a single arise.

Chapter 5

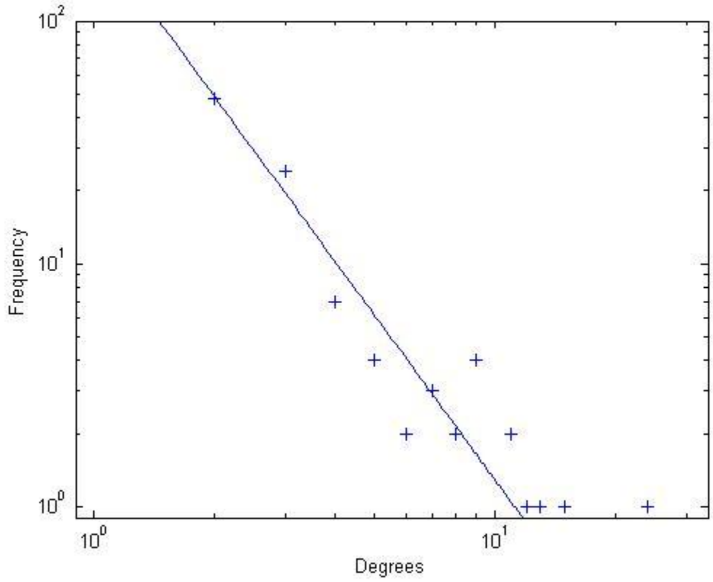
Characteristics and Features of Large Networks

The real road networks of today have similar features with the large networks such as, Random Network, Scale Free Networks and Small World Networks [11]. The content of this chapter is, analyzing the features of the prominent networks which are also valid for the different road networks we encounter in real life. We will introduce these network features with clarifying the reasons, reveal and compare some of the features with each other, which are not investigated in the literature. To make a fair comparison between the networks, we generate these networks having the same number of nodes which is referred to N, and having the same average connections which is referred to K from now on. The first part of this chapter contains the analysis of node related features of the network and the second part contains the analysis of path related features on networks having different type of weight distribution. Instead of analyzing the network topologies one by one, we will explain and compare the difference and characteristic features of each network type sequentially.

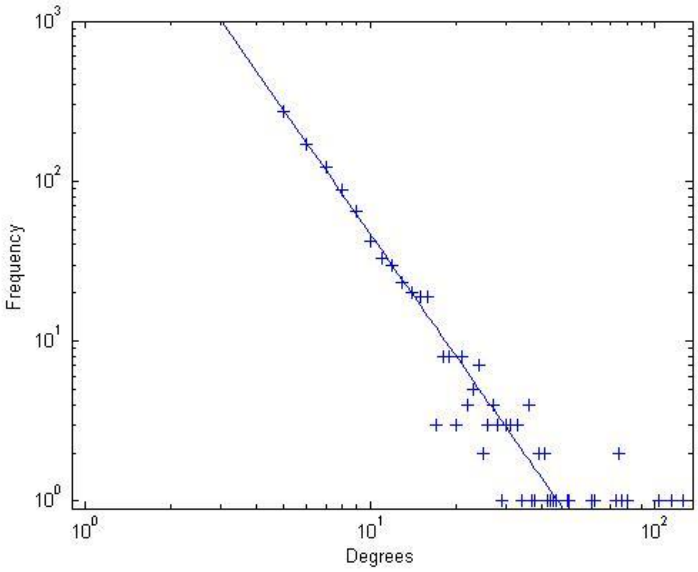
5.1. Node Related Features

Degree Distribution (K): The distribution of average number of neighbors is an important feature of a network, which can help us to differentiate network topologies. Scale Free Networks have Power Law degree distributions which make its topology consisting of few numbers of hubs and a lot of nodes having just a few neighbors. According to the power law property, node degrees and the frequency of node degrees have a linear relation on the log-log scale plot. Generally, Scale Free networks have a slope of α between '-2' and '-3' on log-log scale plot, having the axes degree and its frequency, respectively. We generate Scale Free Networks with Brabasi-Albert Model, which is based on the preferential attachment of new nodes sequentially to the established network [2].

The preferential attachment mechanism uses a seed network to generate a Scale Free network. At each step a new node is attached to the existing network with the specified number of arcs. Those arcs are linked to the nodes with the probability proportional to the degree of the node over the total degrees of the existing network. We generate Scale Free Networks with this method. The networks we worked on have 100 nodes (N) and having degree (K) of 4, or have 1000 nodes (N) having average degree (K) of 10. Degree distributions of sample Scale Free Networks generated with 100N 4K, and 1000N 10 K are shown on log-log plot below.



a) Scale Free Network with 100 N 4 K

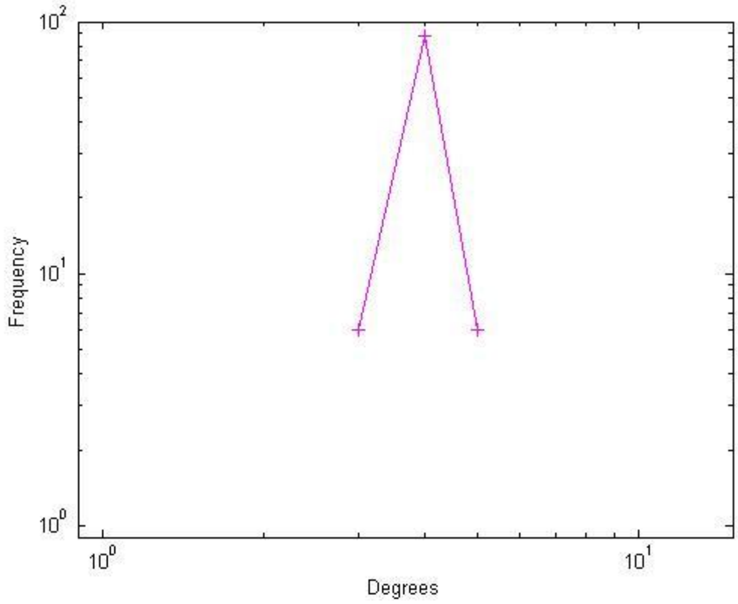


b) Scale Free Network with 1000 N 10 K

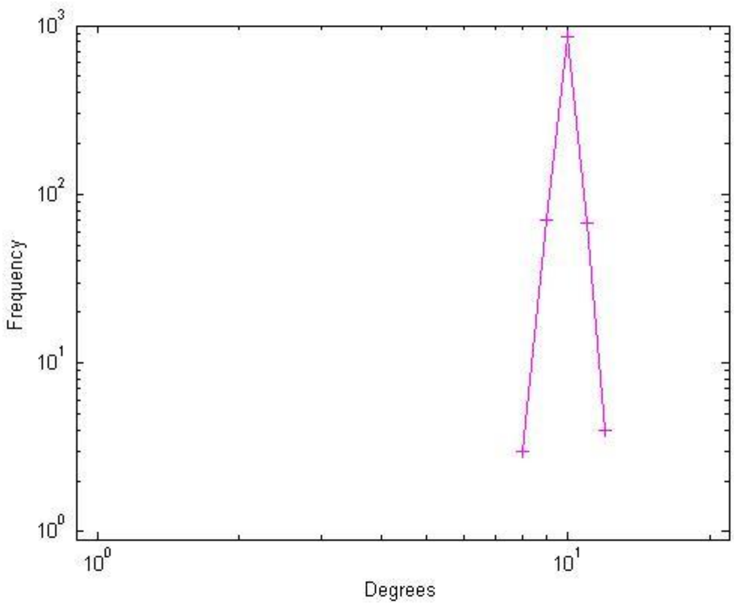
Figure 5.1. The degree distribution of Scale Free Networks on log-log plot. The figure a shows the distribution of a network having 100 nodes and 4 connections on average. The slope of the 100 N 4K network is $\alpha = -2.257$. The figure b shows the scale free network having 1000 nodes and 10 connections on average. The slope of this network's K distribution is $\alpha = -2.541$.

Small World Networks are constructed from a ring structure having nodes with equal number of connections. Each node on the ring has equal number of neighbors on both sides of it. This structure is spoiled by cutting the arcs randomly with a specified probability and relinking them with another node. The most distinguished probability is:

p = 0.015 because of its interesting consequences which will be discussed in part 5.2 [19]. This very small probability that spoils the ring structure results another characteristic degree distribution.



a) Small World Network with 100 N 4 K

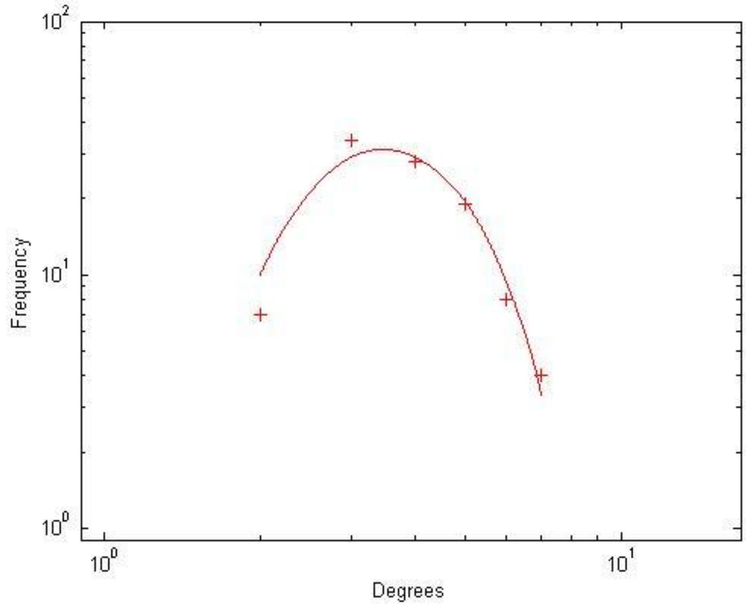


b) Small World Network with 1000 N 10 K

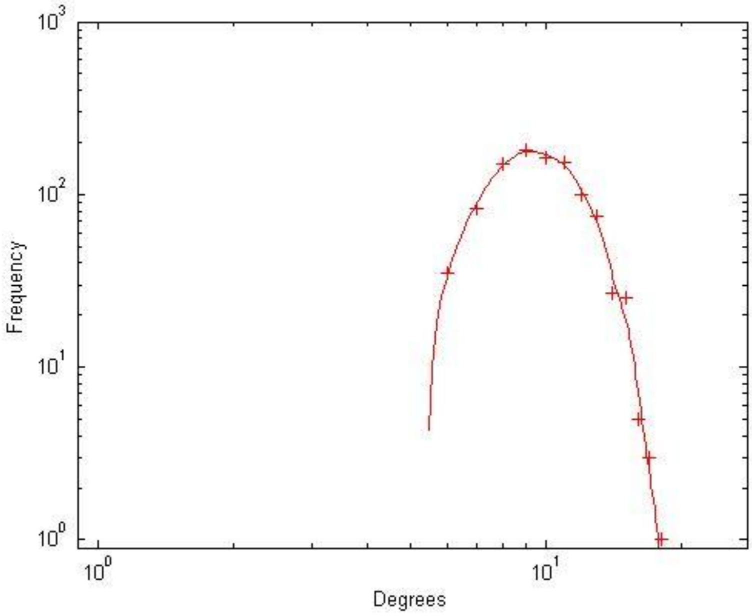
Figure 5.2. The degree distribution of Small World Networks on log-log plot with re-linking probability of 0.015. The figure a depicts the distribution of a network having 100 N and 4 K on average and figure b is a 1000 N and 10 K network. The sharp triangles depicted above show the effects of deterioration of the ring structure.

Random Networks can be created with Erdös-Renyi method which links the nodes with an arc with a given probability. However, this method can guarantee connectedness only for very dense networks. Because, the networks we work on are not dense enough to guarantee this property which may cause problems on the later analyses. We use the ring structure as a seed, also for creating the random networks.

The ring structure is created again with the selected number of nodes and node degrees, then it is spoiled with probability p = 1, which means all of the arcs are re-linked randomly and a random network is generated providing the connectedness property.



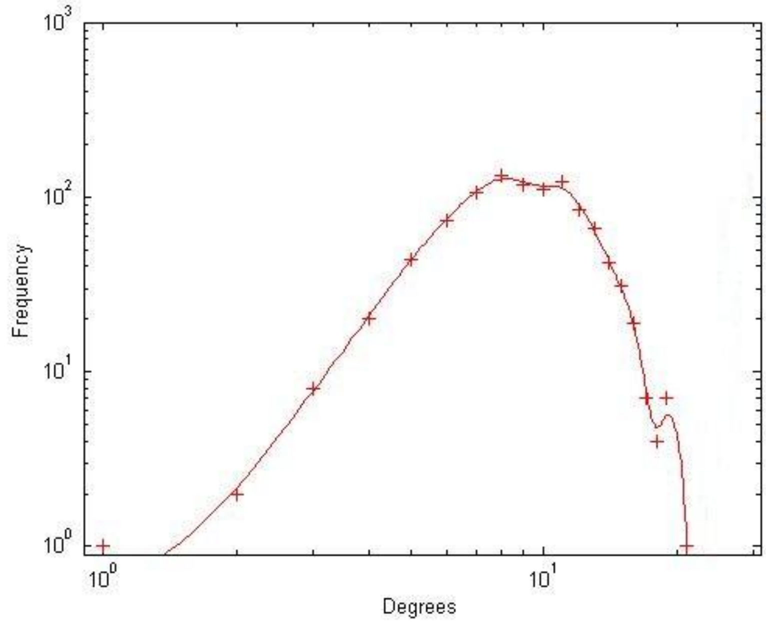
a) Random Network with 100 N 4 K



b) Random Network with 1000 N 10 K

Figure 5.3. The degree distribution of Random Networks on LogLog plot. The figure on the left depicts the distribution of a network having 100 nodes and 4 connections on average and the one on the right is a 1000 nodes and 10 connections network.

The degree distributions of random networks are similar to the normal distribution. To show the similarities of degree distributions between Erdös-Renyi and our method, the degree distribution of an Erdös-Renyi network is depicted below.



Erdös-Renyi Network 1000 N 10 K

Figure 5.4. The degree distribution of Erdös-Renyi network having 1000 nodes and 10 connections on average.

The distribution reveals two aspects of Erdös-Renyi networks. One of them is that, these networks are more irregular than the random networks we generated from the ring structure. However, by generating ample networks and making the analysis on average, the degree distributions will become similar. The other aspect seen on the distribution plot is, with having nodes with zero connections, Erdös-Renyi networks cannot provide connectedness property for 1000 N 10K networks.

Ratio of Second Neighbors (2^{nd}K/K): We define 2^{nd}K/K as the number of neighbors of the neighbors of a node, divided by number of neighbors of the node. This parameter is generated to understand the topology of the networks, knowing only the average or the distribution of node degrees is not sufficient. The figure below will make a better expression of 2^{nd}K/K.

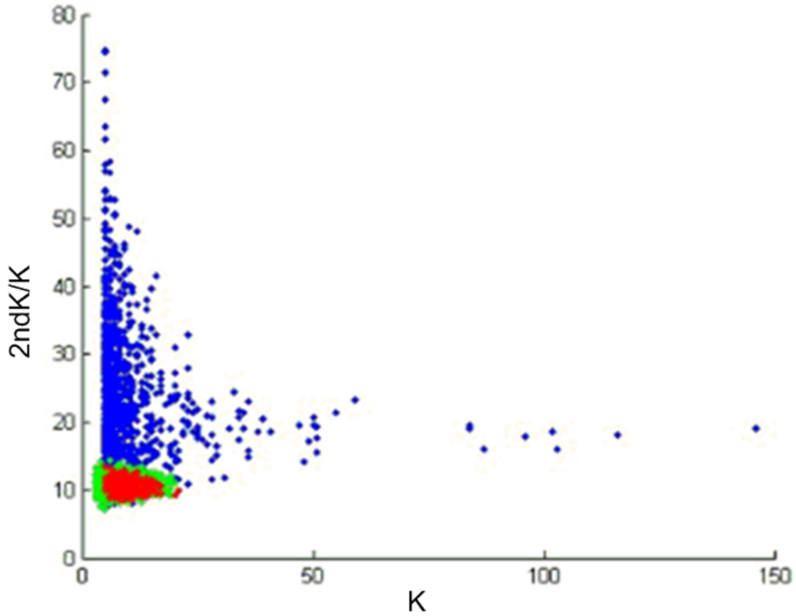


Figure 5.5. The distribution of second neighbors of the nodes divided by the number of first neighbors. All of the distributions are samples of 1000 N 10 K networks. The distribution represented with blue dots belongs to a Scale Free Network. The green one belongs to an Erdös-Renyi random network, and the red one belongs to a random network which is obtained from a ring structure.

2^{nd}K/K distribution shows the topological difference between Scale Free and Random Networks. The distribution above shows the ratio of the second neighbors to the first neighbors of the nodes in a network. For a regular ring structure, 2^{nd}K/K is equal to K, since each node has the same number of neighbors. The reason we do not compare Small World Network's 2^{nd}K/K value with the others is that, this network topology has a 2^{nd}K/K value of K for most of its nodes, and for a very few number of nodes this value is very close to K. Both of the two random networks have similar characteristics as having 2^{nd}K/K distributed around K. The only difference occurs because Erdös-Renyi random networks have larger standard deviation comparing to the Random Networks we create from the ring structure. For Scale Free Network, the characteristic of the distribution is totally different than the Random Network. Scale Free networks have few hubs and many nodes which have very few connections. As a result of this topology, there are many blue dots with small K. These nodes have widely distributed 2^{nd}K/K. They are mostly connected to the hubs, which enables them to reach many numbers of nodes with two hops. On the contrary, the nodes with large Ks have small 2^{nd}K/K values comparing to the nodes having small Ks, since these nodes are mostly connected to the leaves of the network. For a better understanding, the figure representing second neighbors of the nodes ($2^{nd}K$) is shown below.

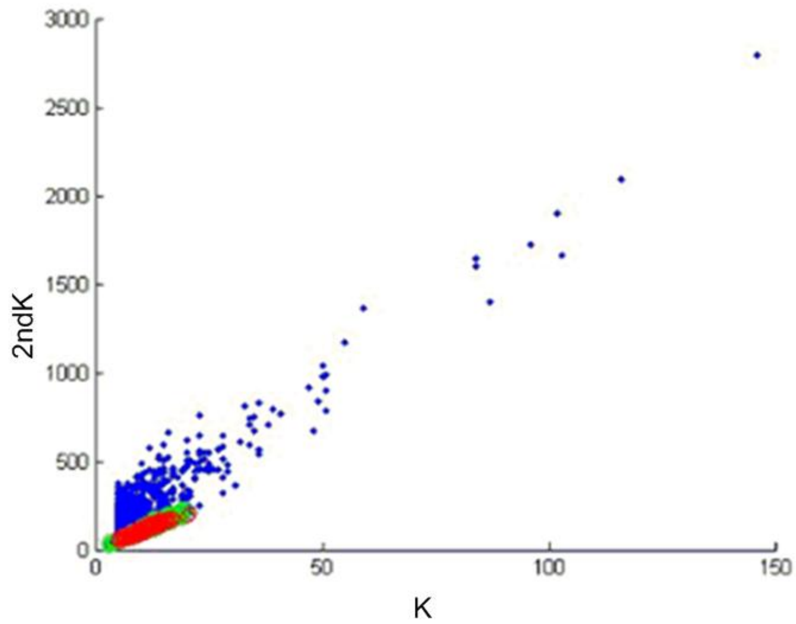


Figure 5.6. The figure above shows the number of second neighbors of the nodes of the network with varying K. All of the distributions are samples of 1000 N 10 K networks. The distribution represented with blue dots belongs to a Scale Free Network. The green one belongs to an Erdös-Renyi random network, and the red one belongs to a random network which is obtained from a ring structure.

The number of second neighbors of some nodes exceed N, since we also count the overlapping second neighbors. K and $2^{nd}K$ values are directly proportional for each network.

Clustering Coefficient (C): The clustering coefficient of a node A is defined as the probability that, two randomly selected neighbors of node A are also neighbors of each other. In other words, it is the fraction of pairs of node A’s neighbor that are connected to each other by arcs. [8] A high average clustering coefficient for a network shows that, there are strongly connected node clusters. For example A ring structure which has an average K greater that two, will have a clustering coefficient of 1, since all of the neighbors of the node are also neighbors between each others. The figure below shows the clustering coefficient of ring structure and small world networks with varying re-linking probability.

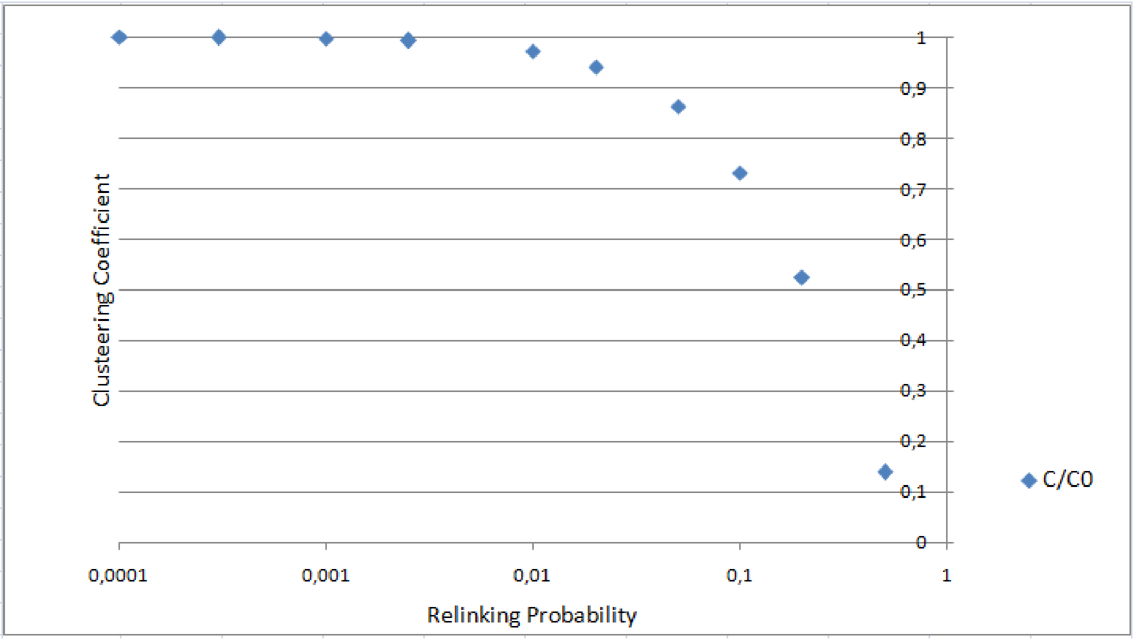


Figure 5.7. The figure shows the average clustering coefficient of the Small World Networks having different re-linking probabilities on log-log plot. Clustering coefficient of small world networks does not decrease before the re-linking probability exceeds the value of 0.015. Clustering coefficients are calculated on 1000 N 10 K Small World Networks, The re-linking probabilities are 0, 0.0001, 0.0003, 0.001, 0.0025, 0.01, 0.015, 0.02, 0.05, 0.1, 0.2, 0.5. The clustering coefficient is calculated by taking the average of 20 samples for each re-linking probability.

Clustering Coefficient for Random and Scale Free Networks are very low comparing to small world networks. However, Scale Free Networks have greater clustering coefficient values than Random Networks. The fact that increases clustering coefficient for scale free networks is, higher probability of having an arc between the nodes which are connected to the same hub.

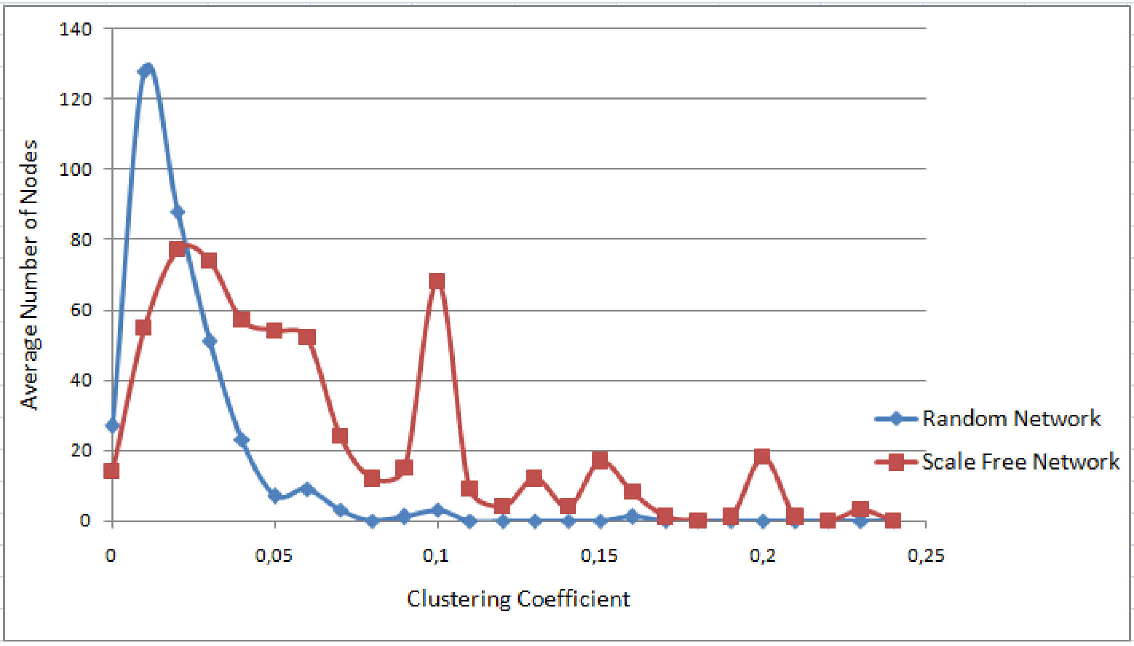


Figure 5.8. The average clustering coefficient distributions of Random and Scale Free Networks. The average is acquired by taking 20 samples for each network topology. Each of the networks have 1000 N 10 K. on average.

5.2. Path Related Features

The networks that we analyze differ also on the path related features. We will work on Scale Free, Random, and Small World networks, comparing the path related features of different networks by defining a path selection method first. Then we will calculate the distance between all pair of nodes of the networks by specified path selection method. We use two path selection methods, which are Shortest Driven Path and MinMax Driven Path. We will introduce and reveal the path lengths of each network calculated by both of the methods.

5.2.1 Shortest Driven Path Length

Shortest Driven path length of a network is obtained by calculating the shortest distance in terms of number of hops for each pair of nodes. If there are more than one Shortest Driven paths between two nodes, one of the equal distanced paths is selected randomly.

For Small World Networks, Shortest Driven path length of the network decreases suddenly, even if the re-linking probability is low.

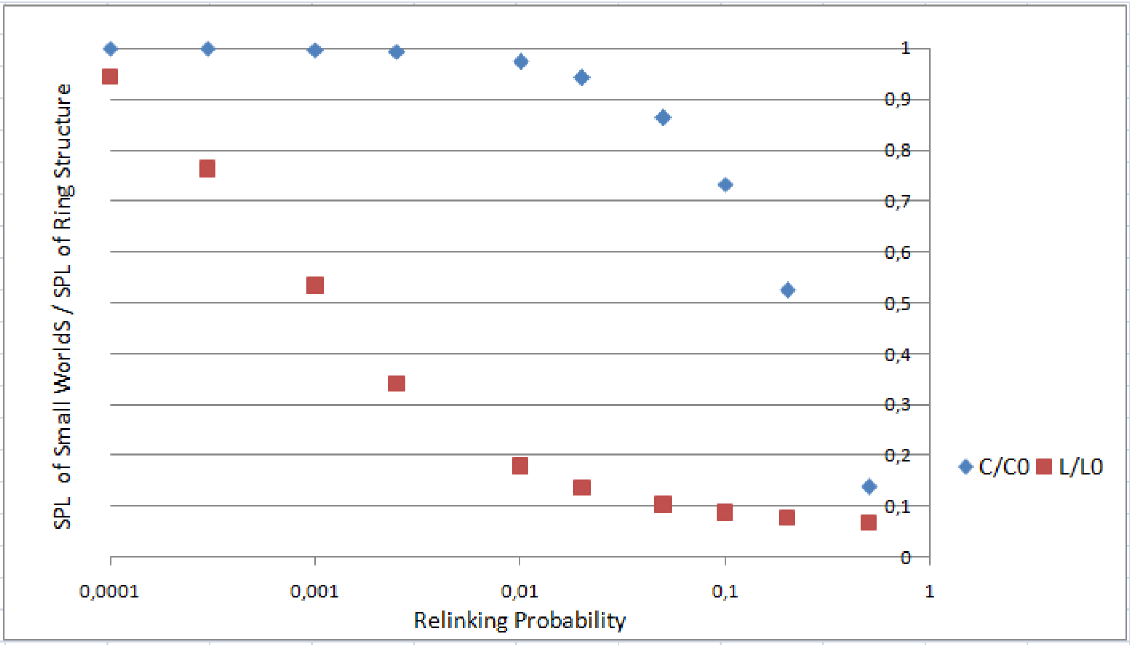


Figure 5.9. The red dots on the figure show the ratio of average Shortest Driven path length of the Small World Networks having different re-linking probabilities to the path length of the ring structure on log-log plot. Shortest Driven path length of small world networks decreases instantly even if the re-linking probability is 0.0003. Shortest Driven path ratios are calculated on 1000 N 10 K Small World Networks, The re-linking probabilities are 0, 0.0001, 0.0003, 0.001, 0.0025, 0.01, 0.015, 0.02, 0.05, 0.1, 0.2, 0.5. The Shortest Driven path ratio is calculated by taking the average of 20 samples for each re-linking probability.

The figure above shows the evaluation of the Shortest Driven path ratio of the small world networks. Even if the re-linking probability is very low, the average Shortest Driven path length of the network decreases instantly. The plot showing the clustering coefficient of the network is kept in this figure intentionally to recall the Small World Phenomena, [20] which shows with re-linking very small number of arcs of the ring structure, both the regularity is conserved and the Shortest Driven path length of the network is decreased to a value which is very close to the random network's Shortest Driven path length.

We will also compare the Shortest Driven path distributions of Scale Free and Random Networks.

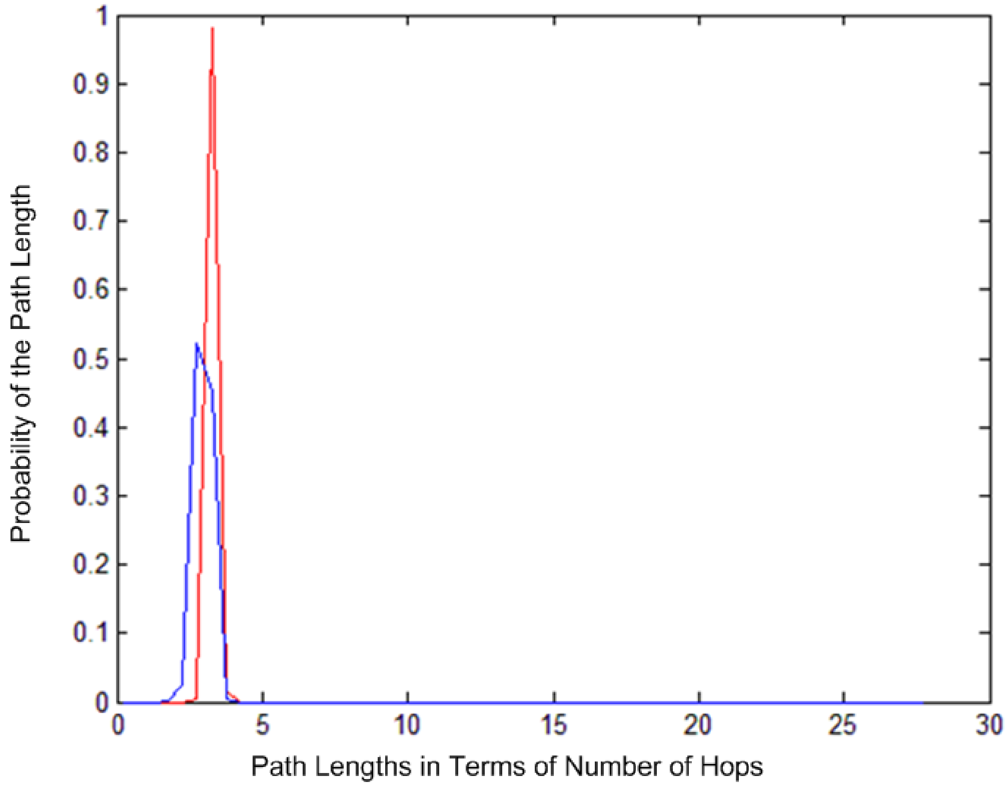


Figure 5.10. Average Shortest Driven Path Length Distribution of 1000 N 10K Scale Free and Random Networks. The average values of 20 networks for each topology are drawn. Blue plots are path length distributions of Scale Free Networks and red lines are path length distributions of Random networks.

The Shortest Driven path length distributions of Scale Free and Random Networks show that, Shortest Driven path length for Random network distributes on a narrower interval than Scale Free network. Because each of random network's nodes has almost same degree, every node has similar average Shortest Driven path lengths. Since the degrees are varying for the Scale free network, the Shortest Driven path distribution is on a wider interval. Having hubs enables the Scale Free Network to have lower average Shortest Driven path lengths comparing with random networks.

6.2.2. MinMax Driven Path Length

We are only able to analyze MinMax Driven Path Length on the weighted networks. MinMax Driven path selection method tries to avoid the arcs which have the highest weights. This algorithm compares the highest weighted arcs of all possible paths between a pair of nodes. It picks the path whose highest weighted arc is the lowest comparing to the other paths' highest weighted arcs. Then it returns the number of hops of the selected path.

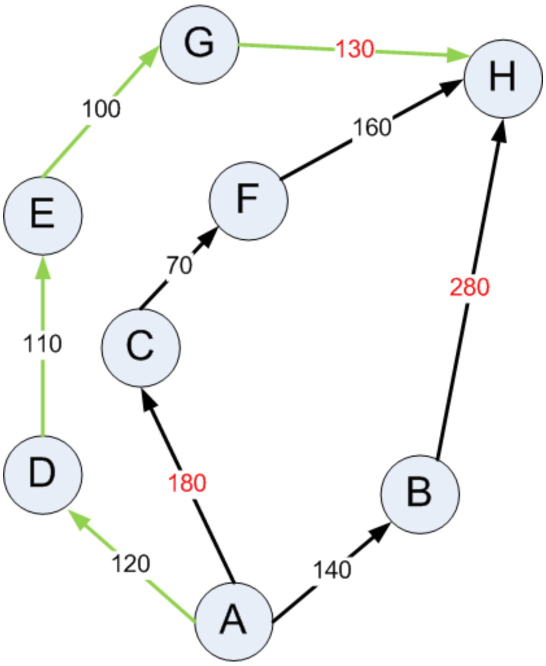


Figure 5.11. The figure demonstrates the MinMax Driven path selection method. The path A-D-E-G-H is selected because of it's the highest arc length, which is between Node G and Node H, is 130. Comparing to the other paths' highest arcs, the arc between Node G and Node H is the lowest. The MinMax Driven path selection returns 4 for the path length between Node A and Node H.

MinMax Driven path selection method does not consider the number of hops or the sum of weights on the path while deciding the path between two nodes. It only tries to minimize the highest weighted arc of the selected path. We also analyzed MinMax Driven path ratio for the Small World Networks. The probability which is crucial among the average Shortest Driven path length and clustering coefficient is also valid for MinMax Driven path.

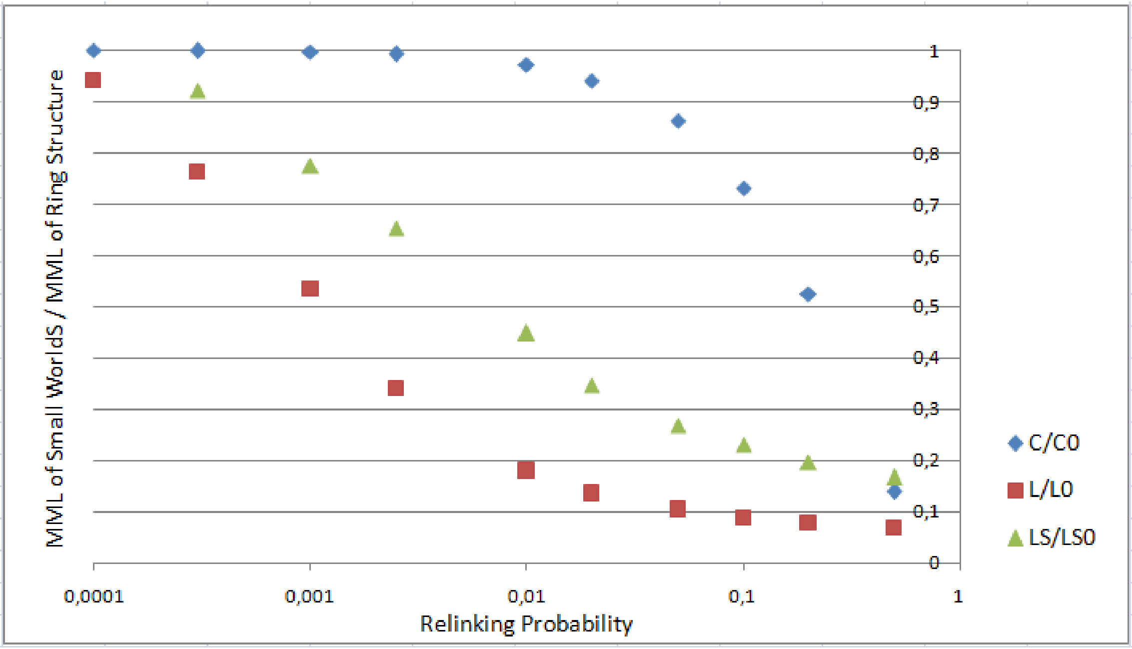


Figure 5.12. The green dots on the figure show the ratio of average MinMax Driven path length of the Small World Networks having different re-linking probabilities to the path length of the ring structure on log-log plot. MinMax Driven path length of small world networks also decrease, but the slope is not steep as Shortest Driven path. However, we can see that both MinMax Driven path and Shortest Driven path have a similar reaction against re-linking. MinMax Driven path ratios are calculated on 1000 N 10 K Small World Networks, The re-linking probabilities are 0, 0.0001, 0.0003, 0.001, 0.0025, 0.01, 0.015, 0.02, 0.05, 0.1, 0.2, 0.5. The MinMax Driven path ratio is calculated by taking the average of 20 samples for each re-linking probability.

One of the most important features of the MinMax method is; the distribution of path lengths doesn't change even if the weight distribution of the network expands, narrows, or change the characteristics, as long as it remains symmetric. The figures below are MinMax Driven path length distributions for Scale Free networks having 1000 N 10 K. The first figure shows MinMax Driven path length distribution of Scale Free networks whose arc weights are distributed using Gaussian distribution having the mean $\mu = 1$, and standard deviations $\sigma = 0.1$, $\sigma = 0.2$, $\sigma = 0.3$ respectively. The second figure also shows MinMax Driven path length distribution of Scale Free networks whose arc weights are distributed using Gaussian and Uniform distributions. For both of the figures, there are only slight changes on the MinMax Driven path distribution for the networks.

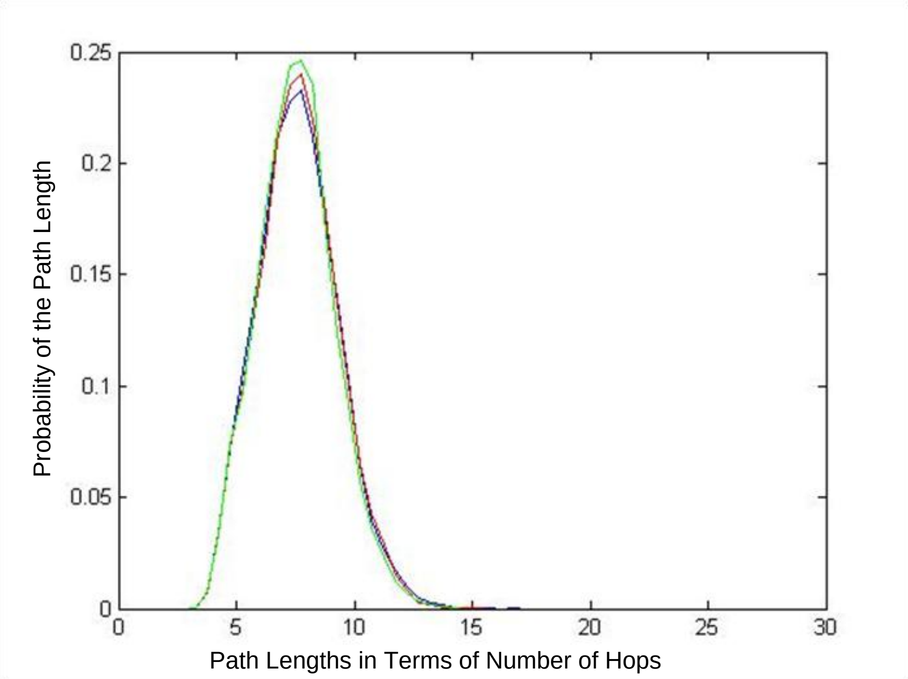


Figure 5.13. The average MinMax Driven path distributions of 20 Scale Free Networks having 1000 N and 10 K. The weights of the networks are distributed using Gaussian distribution having mean $\mu = 1$, and standard deviations $\sigma = 0.1$, $\sigma = 0.2$, $\sigma = 0.3$ respectively

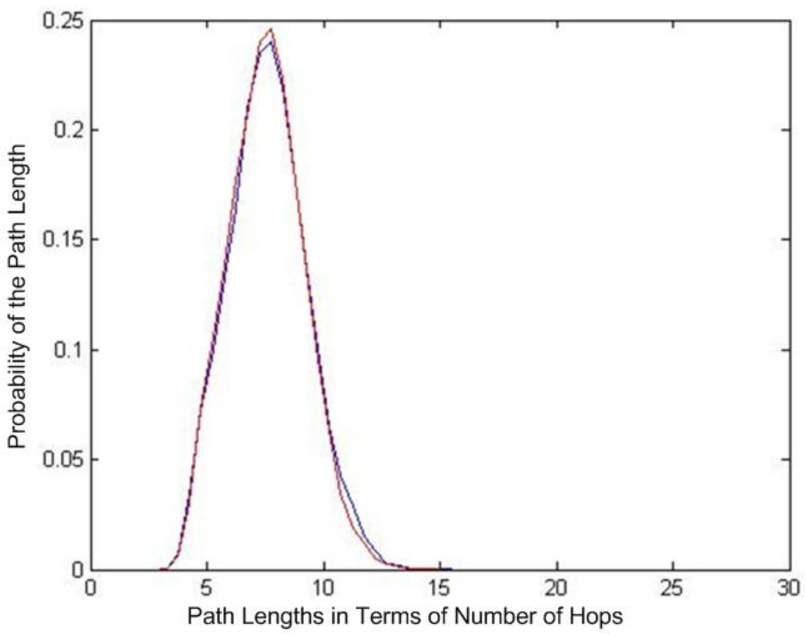
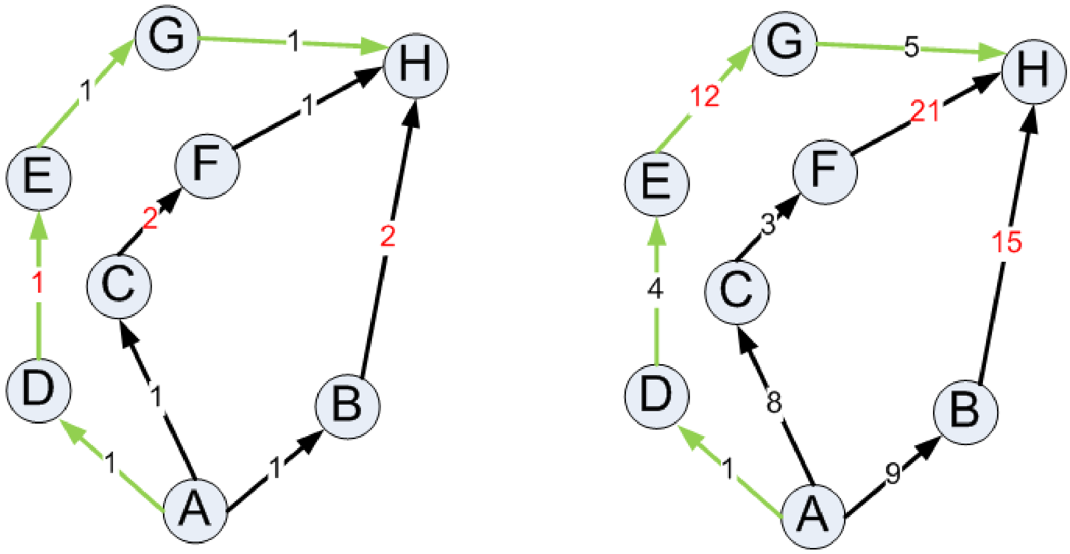


Figure 5.14. The average MinMax Driven path distributions of 20 Scale Free Networks having 1000 N and 10 K. The weights of the networks are distributed using Gaussian and Uniform having mean $\mu = 1$, and standard deviation $\sigma = 0.2$.

The changes on the arc weight distributions don’t affect on the MinMax Driven lengths. However, the usage of MinMax Driven path selection method is more favorable than the Shortest Driven path selection only if the arc weights are widely distributed [5]. Assume that there is a weighted network where the number of hops shows the physical distances between nodes, and the arc widths are the congestion rates. If a driver ignores the traffic congestion, he selects the path to his destination point with Shortest Driven path selection method. If he cares about avoiding the traffic, he will select MinMax Driven path selection method. Since the arc weights specify the congestion on the same distanced road pieces, if all of the arcs on the network are congested or empty with the similar rates, selecting the MinMax method causes the driver to reach his destination point with more number of hops than the Shortest Driven path method on average which results waste of time. Whereas, if some of the roads of the network are extremely congested, some of the roads of the network are empty and congestion of the remaining roads is distributed between the two extremes, the driver can save time with avoiding the congested roads. He will use a physically longer road, but he will arrive to his destination faster. The traffic flow obeys the Power Law distribution in some real world networks [11], which will make the MinMax Driven path selection method is the most advantageous comparing with symmetric arc weight distributions that we used.



a) Narrowly distributed arc weights b) Widely distributed arc weights

Figure 5.15. a) The arcs of the network are narrowly distributed, so the probability of the MinMax method to fail is higher, by transmitting the driver to his destination in a longer time than the Shortest Driven path method. b) The arc weights of the network are widely distributed, so the probability of the MinMax method to success is higher, by transmitting the driver to his destination in a shorter time than the Shortest Driven path method.

The figure below shows the comparison of the average path lengths for Scale Free and Random Networks with using MinMax Driven path selection method.

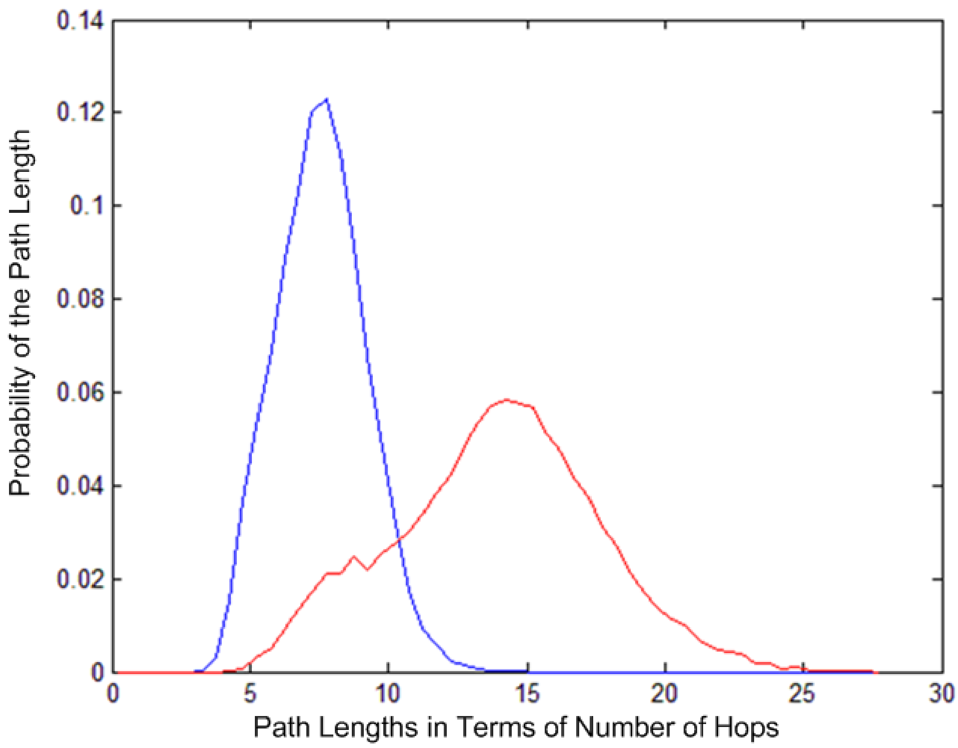


Figure 5.16. Average MinMax Driven Path Length Distribution of 1000 N 10K Scale Free and Random Networks. The average values of 20 networks for each topology are drawn. Blue plots are path length distributions of Scale Free Networks and red lines are path length distributions of Random Networks.

The average path length of MinMax Driven path selection is lower for Scale Free Networks than Random Networks. This results with an inference that applying MinMax Driven path selection on Scale Free networks will be a more powerful competitor against Shortest Driven path selection method than Random Networks. We also want to make an analysis about the swell that exists on the shorter path lengths and the tail on the longer path lengths for MinMax Driven path length distribution.

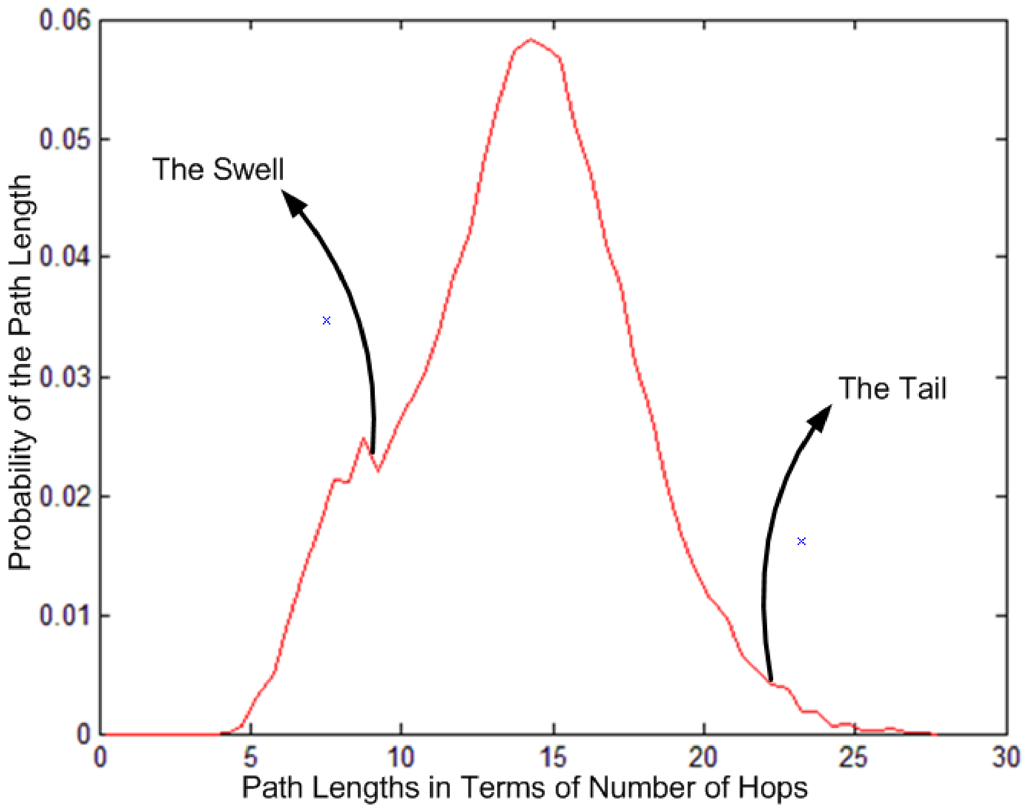


Figure 5.17. This figure shows the Swell and the tail part of the MinMax Driven path distribution.

The swell and the tail of the MinMax Driven path length distribution exist on both Scale Free and Random Networks. The figure above is one of the distinctive figures of the swell and the tail. The reasons of these shapes can be explained with the redundancy of networks. Assume that there are alternative paths between two nodes on a weighted network. The network which has redundant paths, has numerous long alternative paths and less number of short alternative paths in terms of number of hops. For the long alternatives, the probability of having one of the extreme arc weights on the path is higher than a shorter alternative path. To clarify, if we select two paths randomly on the network, we will come up with the highest arc weight on the longer path more probably. This probability explains the reason of the tail on the MinMax Driven path distribution. The opposite is also valid and explains the reason of the swell. The swell and the tail are not clear on the Scale Free Network's MinMax Driven length distribution, since the

MinMax Driven path lengths are shorter, and are distributed on a narrow interval comparing with the Random Network's.

We also need to make a comparison between path lengths of MinMax Driven Path and Shortest Driven Path.

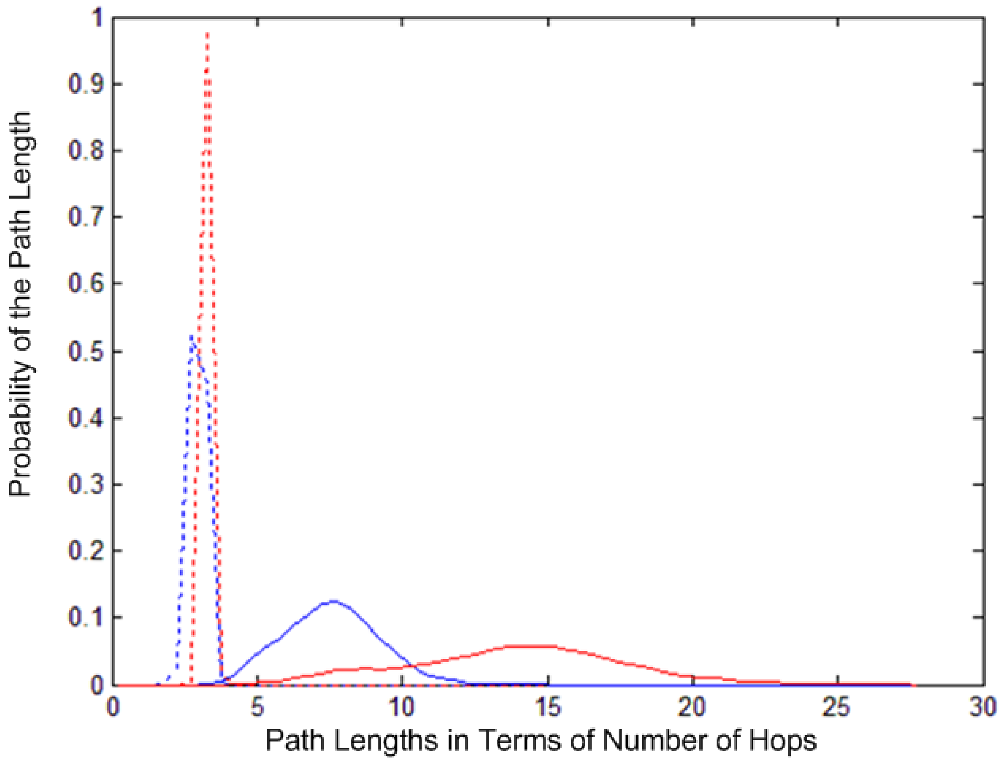


Figure 5.18. Average Path Length Distribution of 1000 N 10K Scale Free and Random Networks. The average values of 20 networks for each topology are drawn. Weight distribution of the networks are derived from uniform distribution has a mean of 1. Dashed lines represent Shortest Driven path length distributions and continuous ones are min-max path length distributions. Blue plots are path length distributions of Scale Free Networks and red lines are path length distributions of random networks.

The average path length for MinMax Driven length distributions are significantly more than Shortest Driven path distributions whose reasons are revealed in detail at the previous sections. We can also conclude that, using MinMax Driven path length is more suitable on the Scale Free networks, since the average path length of the Random Networks are higher than the average path length of Scale Free Networks.

We have analyzed some important features of prominent networks in this chapter. The features we analyzed not only demonstrate the topologies, but also inspire us to find ways to control the traffic flow on the networks. In the next chapter we will use Shortest Driven Path Lengths and MinMax Driven Path Lengths in order to generate a method which tries to decrease congestion on different networks.

Chapter 6

Generating Fair Networks with Road Closures

Road networks usually evolve, or are established by the authorities [8] [18]. The resulting networks in general have very similar characteristics with the networks we have mentioned in Part 6.1. [2]. As a consequence, there are many redundant paths on the road networks which may result with the traffic congestion. In Part 4.2., we have showed that adding a redundant road to the network, decreased both the total travel times and the blocked time of the vehicles on the network. However, in the application we used all of the agents who are adaptive to the traffic density and trying to avoid from the congested roads. In real life, most of the individuals are not able to see the traffic information and the greatest majority insists on the roads that they are used to drive along even if it is congested. Because of these reasons, large traffic networks are still closer to the assumptions of Braess Theorem, than the assumptions we have made in Chapter 4.1. We will discuss about Braess Theorem in this chapter, and then mention about a network efficiency measure called ‘Price of Anarchy’ [20]. Lastly, we will suggest a different network efficiency measure named ‘Ratio of Justice’.

6.1 The Features of Braess Theorem

Having alternate paths on transportation or an internet network is considered to be a favorable feature for the network intuitively. However, Dietrich Braess in 1968 [4], became a pioneer by introducing the Braess's Paradox. This phenomenon tells that adding new roads to the traffic network may increase the total cost of the network.

Before introducing the Braess Paradox, we should mention about the Equilibrium Traffic.

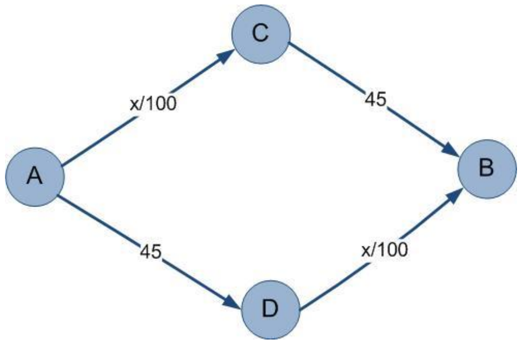
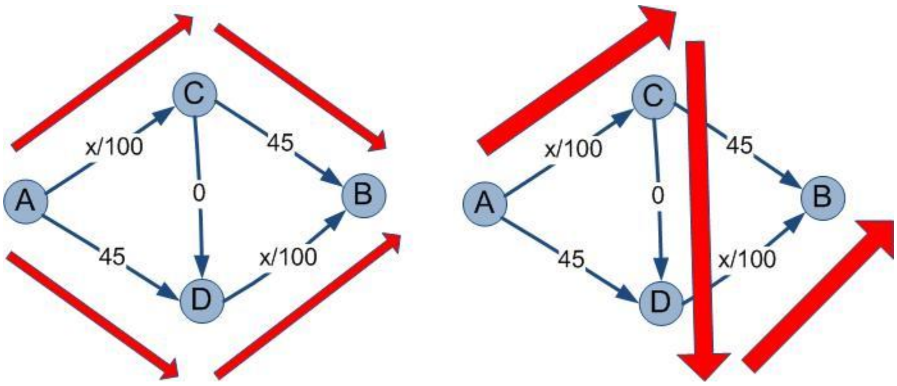


Figure 6.1. The directed traffic network. Vehicles are leaving from Node A and trying to reach Node B. Arc label x/100 represents the travel times when there are x vehicles using the arc. The arc label 45 represents that each vehicle passes the arc in 45 travel time units regardless of the number of vehicles using it. Each driver tries to reach Node B as fast as possible and the equilibrium point which no more cars can increase its travel time occurs when they divide evenly over the two possible paths.

Suppose we have a directed graph and vehicles leaving from Node A to Node B represent morning commuters. There are two possible paths from Node A to Node B. Also there can be enormous number of drivers who can have different strategies. If we list all of the drivers and strategies and apply the ones, which decrease the time to reach to the destination point, after finite iterations, we come up with a point that no driver can decrease his travel time by changing his strategy. This is called Nash Equilibrium [8].

If there are 4000 cars on the network above, the equilibrium point will be 2000 cars choosing the A-C-B path, and 2000 cars choosing A-D-B path. By dividing the number of cars evenly, none of the drivers can choose a path which will decrease the travel time he experiences. Since every driver wants to reach to the destination over the fastest path, equilibrium point will give the lowest total cost for the network on Figure 6.1.

However if the government builds a fast highway between C and D, which has a journey length of 0, regardless of the number of cars using it, all of the drivers use the highway, which is recently built, to decrease their journey length. When the Nash Equilibrium is reached again, no drivers can decrease their journey length with altering their strategy. Moreover, the total cost of the network becomes higher than the previous situation before the fast highway between Node C and Node D is constructed.



a) The Social Optimum b) The Nash Equilibrium

Figure 6.2. A fast highway is built between Node C and Node D. Journey length of 0 is assigned to the network to make the illustration simple. There are 4000 cars moving from Node A to Node B. On the Social Optimum traffic pattern the Social Optimum Cost = $\sum_{i=1}^{2000}(45+\frac{2000}{100})$+ $\sum_{i=1}^{2000}(\frac{2000}{100}+45)$ and The Nash Equilibrium Cost =$\sum_{i=1}^{4000}(\frac{4000}{100}+0+\frac{4000}{100})$.Social Optimum Cost is 26,000 units and Nash Equilibrium Cost is 32,000 units.

The results of Figure 6.2 show that adding a redundant arc to the network may not decrease the total cost of the Network. The reflection of Braess's Paradox to the real life was applied in Seul, South Korea by destructing a highway, which has the same characteristics with the arc between Node C and Node D. Then the traffic congestion decreased in the city. [8] At some other cities such as Stuttgart, Germany and New York City Braess's Paradox have been used as a policy decision on road closures.

6.2 The Price of Anarchy

According to the path selection algorithms, every agent tries to minimize its total travel time, rather than an altruistic behavior, which will minimize both his and the total system's travel time. To measure the cost between the strategies in which each individual urges to have the highest possible profit unilaterally and the strategy of individuals try to decrease the total cost of the system, we should see the effect of a recently constructed or a closed road on a network by calculating the ratio of the total cost of the Nash Equilibrium over the total cost of the Social Optimum. This ratio is called as the Price of Anarchy [20]. We can also calculate the price of anarchy on different network topologies to understand which network is more resistant to the drivers' selfish path selection behaviors.

$$\text{POA} = \frac{Nash\ Equilibrium}{Social\ \ Optimum} \text{ for the given flow F.}$$

Assume that a flow of 16 vehicles are sent from Node A to the Node D on the network below.

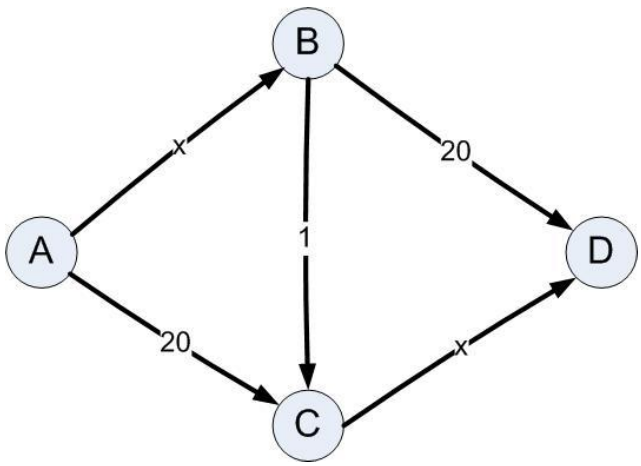
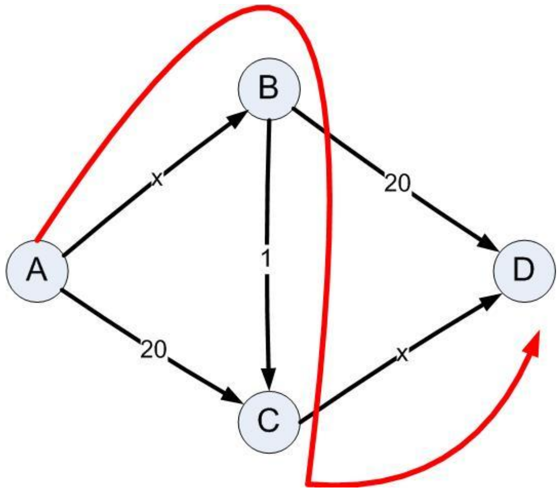
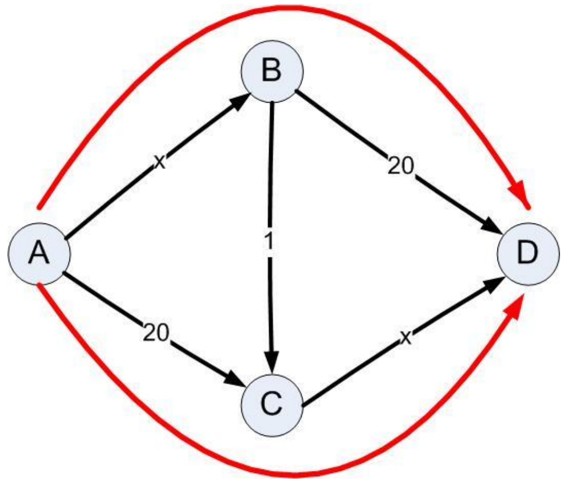


Fig 6.3. The roads labeled with x represent the congestion delay occurs on the roads directly proportional with the number of vehicles using the road, and the roads labeled with numerical numbers are insensitive to congestion or the density of the flow passing through it. No matter how many vehicles are driving through, the length or the delay of the road is constant.

The Nash Equilibrium is reached when all of the vehicles are unable to decrease its journey length by altering the selected path, and the Social Optimum is reached when no more vehicles can decrease the total cost by changing its strategy individually.





a) Nash Equilibrium b) Social Optimum

Figure 6.4. a) Represents the flow when the Nash Equilibrium is reached. All of the vehicles follow the path of A-B-C-D. b) Represents the social optimum. 16 vehicles are divided into two where 8 of them follow the upper path and 8 of them follow the lower path.

$$\text{POA} = \frac{Nash\ Equilibrium}{Social\ \ Optimum} = \frac{16\ x\ (\ 16 + 1{+}16\ \ )}{8\ x\ (8{+}20) + 8\ x\ (8{+}20)} = \frac{528}{448} = 1.18$$

Note that the 'Price of Anarchy' is depending on the flow, source, and destination nodes of the flow at this toy network. We can consider decreasing POA to 1, as a desirable phenomenon for a network in which the selfish path selection behavior of vehicles does not cause an increase on the total cost. From a different point of view, there is no need for headquarter, traffic lights, or toll roads to urge the traffic to different parts of the network for decreasing the total congestion on the network. Briefly, while generating or renovating road networks; POA should be taken into account to decrease the total cost of the network.

6.2 The Ratio of Justice

Now, consider a large network on which we want to define a measure similar to the Price of Anarchy. Different than the toy networks, the flow, which is the total number of vehicles on the network, is vague. Also the measure should be defined by calculating the flow between each pair of nodes.

In order to suggest a ratio which is similar to POA, we first need to find the most frequently used roads of the network, if there is a homogenous flow on it. Assume that the network has (N-1) vehicles, and each vehicle of a node tries to reach another distinct node. We send all of the vehicles of the network to their destination nodes sequentially with a pre defined path selection method, also waiting a vehicle to arrive its destination node before sending the other one. After all of the vehicles have reached to their destinations, we store the frequencies of the roads that they used. By this method we acquire the arcs that become more congested than the others, which we call these arcs that the arcs which have the highest betweenness centrality.

If there are arcs on a network which are excessively preferred by the vehicles, it possibly results with congestion caused by the same reason of the Nash Equilibrium where each individual tries to decrease its journey length with a self centered path selection algorithm. To decrease the cost, of which is a result of individual's selfish path selection strategy, we can close the roads which have high betweenness centrality to prevent this situation until to the point when individuals' selfish path selection strategy will not affect the total cost.

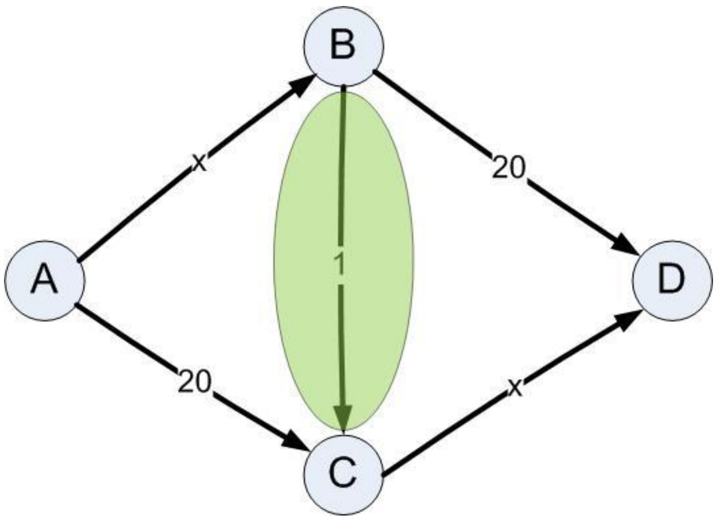


Figure 6.5. Removal of the arc, between Node B and Node C, which has the highest betweenness centrality, compensates the Nash Equilibrium with the Social Optimum.

Instead of using the Price of Anarchy formula, we suggest another method which also makes the network fair for all vehicles on it. We have suggested two path selection algorithms to use on the networks. One of them is the Shortest Driven path method and the other one is MinMax Driven path method. Figure 5.18. shows the average path length distribution of each method used to find a path between each pair of nodes. To calculate the number of steps between each pair of nodes, we send just one vehicle from the source node to the destination and do not send any other vehicle until the previously sent vehicle reaches its destination, and then we store the path lengths of each vehicle. The Shortest Driven path lengths distribution becomes a sharp, narrow distribution ranges on a small interval. To clarify, if all of the vehicles on a network uses Shortest Driven path selection algorithm, it will cause the Nash Equilibrium effect on the network, since all of the vehicles are trying to minimize own journey length. For the ratio we suggest, we will call the cost of Shortest Driven path selection as 'Selfishness Cost'. However, when we analyze the MinMax Driven path length distribution, we conceive that vehicles sacrifice from selecting the path to a node, which makes the physical distance smaller. By selecting MinMax Driven path, they are avoiding the most congested road piece in order to reach their destination as quickly as possible, but if we consider this situation with a different point of view, they are also avoiding to make the congested road pieces more crowded. This behavior resembles the social optimal behavior in which individuals do not tend to move in order to save their profits. We will use min-max path selection cost as 'Altruistic Cost' in the ratio we suggest.

We are looking for the networks or trying to modify the ones we have, with constructing or closing the roads, to have the 'Ratio of Justice' of 1. If we ignore the geographic obstacles and assume the flow of the network is homogenous, we can achieve the desired ratio of justice with closing the roads sequentially from the highest betweenness centrality to the lowest. By closing the roads, which are frequently passed by the vehicles, we are increasing the vehicles journey lengths, but also distribute the congestion by redirecting the flow to the longer paths, which reduces the gap between altruistic cost and selfishness cost. ROJ is defined below.

$$ROJ = \frac{Selfishness\ Cost}{Altruistic\ \ Cost}$$

Assume that we have two types of networks with different topologies. One of them is a Scale Free Network, and the other is a Random Network. We use normal distribution for assigning the weights of the arcs. Both the shortest betweenness centrality and the

MinMax betweenness centrality are calculated. Shortest and MinMax betweenness centrality results put the most frequently used road pieces forward, depending on the path selection criteria. The shortest betweenness centrality is derived by calculating the Shortest Driven path for each pair of nodes in the network. Intuitively, the arcs which correspond to shortcuts or bridge like structures between large subnetworks, expected to have larger utilization frequency, will be attained by using that method. Since the priority of MinMax Driven path selection method is not directly related with using the shortcuts, MinMax betweenness centrality will return the arcs with less utilization frequency than the shortest path betweenness centrality, because of the method's path selection criteria is based on arc weights and the distribution of paths is larger comparing to the Shortest Driven path's.

Closing the most frequently used roads will direct the traffic flow to the other paths and close the gap between selfishness cost and altruistic cost for the network. To understand the reactions of the networks to the road shuts, we used three different methods. In the first one, we closed the roads respectively according to shortest path betweenness centrality, and then analyze average Shortest Driven path and MinMax Driven path distances on the networks. The other method used is closing the roads according to MinMax betweenness centrality and analyzing average Shortest Driven and MinMax Driven path distances. The last one is closing the roads both according to shortest and MinMax path betweenness centrality. As a consequence of the last method; we acquire two different network topologies by closing the roads with different closure criteria. We only show the results of the first method here since using the shortest path betweenness centrality on road closures has more influential effects on the networks. To see the results of the other road closure methods Appendix B can be checked.

The figures below show average path lengths according to the Shortest Driven path road closure method. Fifty samples are used for each network topology, and error bars are placed according to the standard deviation of the path length distributions. Five percent of the roads, which have the highest betweenness centrality, are closed at each step according to the betweenness centrality method. The roads that are closed are never opened at the later iterations. If the non existence of a road makes the networks disconnected, we do not close that road. As a result we end up with a tree after all of the redundant roads are closed. There are different closure methods that we worked on, which are keeping a line instead of a tree at the end, or letting the network to be disconnected. The resulting figures of other road closure methods are also in Appendix B.

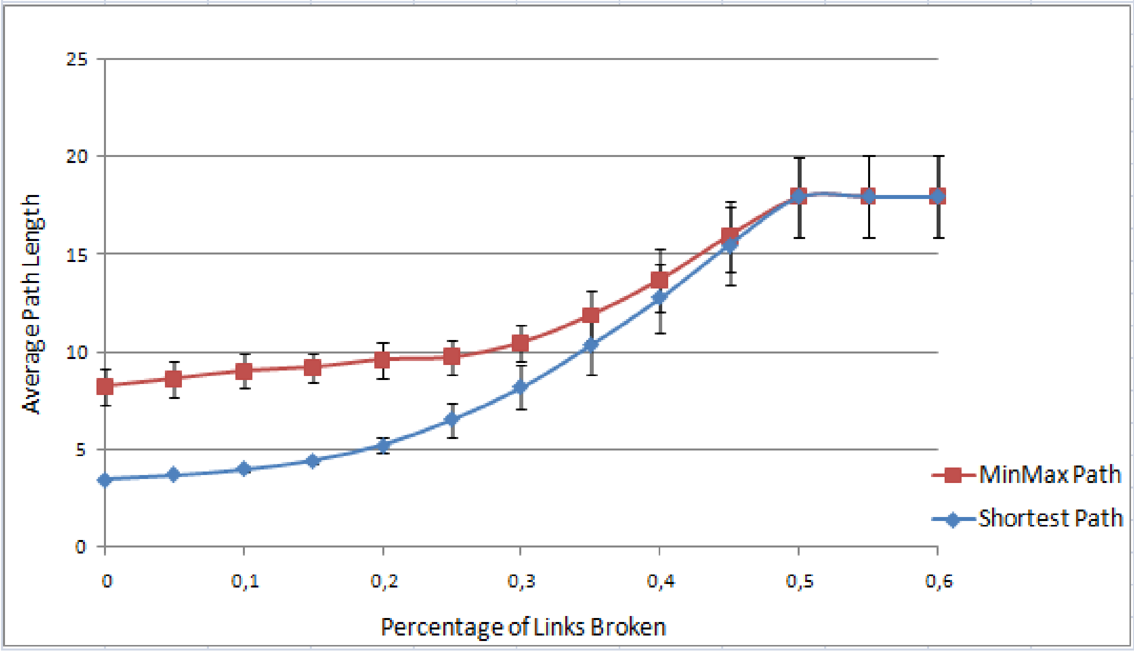


Figure 6.6. The average path lengths of fifty undirected **Random Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path length and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to shortest path betweenness centrality. Error bars are placed with respect to standard deviations. The effective interval is between 20% and 25% of the closed roads

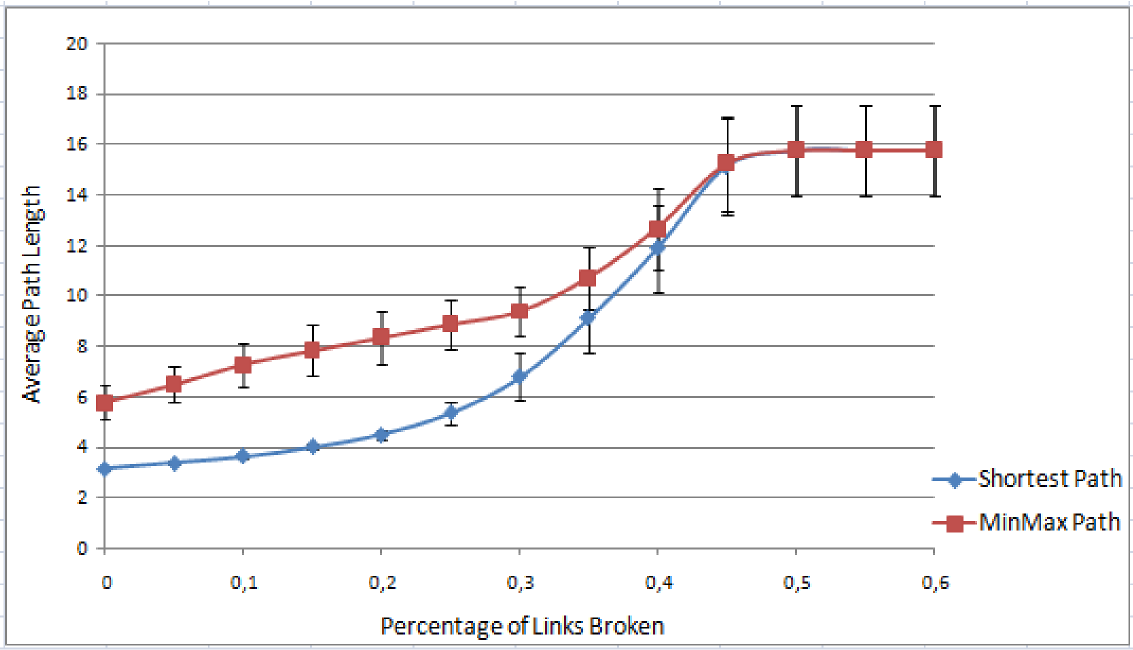


Figure 6.7. The average path lengths of fifty undirected **Scale Free Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path length and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to shortest path betweenness centrality. Error bars are placed with respect to standard deviations. The effective interval is less than 5% .

For the number of nodes and average connections of the networks that we experience, after closing half of the roads, the networks are converted into trees. Closing the roads with the shortest path betweenness centrality results with similar consequences for both Scale Free and Random networks. One of the main reasons for that is, closing the roads with shortest path betweenness centrality closes the shortcuts and continuous closing operation leads us to a new network where the remainder paths are longer than the previous state. To sum up, shortest betweenness centrality method spoils the topology of the network with the quickest way, and this is the most important reason why we do the analysis with it instead of MinMax path betweenness centrality method.

Another important issue is to find the ratio of justice for a network. The ratio of justice becomes 1 for the networks whose all redundant paths are eliminated by closing the roads. You can see the result on Figure 6.7 and Figure 6.8. After pruning 50% of the roads, the MinMax Driven and Shortest Driven path lengths overlap for both Scale Free and Random Networks. However, by converting the network into a tree, we maximize both the selfishness cost and altruistic cost, with eliminating alternative paths, which will be unpleasant for the residents of the network. We need to find an ‘effective interval’ for each network, where the topology of the network is slightly changed and the gap between selfishness cost and altruistic cost diminishes. The effective interval for the Scale Free Network we worked on is between 0% and 5%, and for the Random Network it is between 20% and 25%.

According to the ratio of fairness, Scale Free Networks are more equitable than the Random Networks. Based on this information, we can suggest that constructing road networks similar to the Scale Free Networks will be more effective in term of fairness features we introduced. If the road network’s topology is similar to the random network, road closures which alter the topology similar to Scale Free network may be applied.

Chapter 7

Conclusion and Future Work

The traffic congestion is an important issue both on academia and social life. There are numerous number of papers concerning this issue. We hope to make some contributions to the literature with the results and inferences of this M.Sc. thesis.

Adaptive agent based application we develop works fine for small networks. By manipulating the information, which is received by a group of agents, we succeed to decrease the average congested time of the vehicles on the network. The results do not show a significant difference for average travel times of the manipulated and regular version of the network, since all agents have the same destination point. However, performance of the information manipulation algorithm is sufficient for now, because it prevents the congestion on the road part which is defined as the critical arc for the network.

The experiment we have made on the Braess network also has a valuable consequence showing the importance of having the instant traffic information. Adding a road to a network improved both the average travel times and time spent in traffic jam, whereas it was reducing on the original problem. The difference is a result of adaptive agents and continuously provided traffic information which enables the agents to select their paths by an "involuntarily cooperative" path selection strategy.

We also analyzed prominent large networks assuming that the drivers on the networks neither adaptive nor cooperative. The results show that Scale Free Networks are more successful than Random Networks in order to absorb the congestion which is a result of non cooperative behavior of the vehicles on the road.

Then we define a ratio indicating the fairness of the network. To clarify, by keeping the ratio of fairness closer to 1, the travel times of the drivers will be more equally distributed. As a result the total congestion is believed to decrease.

We haven't done any numerical experiments for the ratio of fairness except for analyzing the effects of road closures on the ratio of fairness. We are keeping the influential experiments of ratio of fairness for the future work.

We will do the experiments for the information manipulation on larger networks, which have some agents whose source and destination point is different than the majority. We expect also to achieve lower average travel times, since the vanished congestion will diminish the travel times of the vehicles which are moving on a different direction than the majority of the population.

We will also analyze the large networks, especially the path length distribution of MinMax method of the networks, when the arc weights are power law distribute. We are interested in power law weight distribution of arcs since there are some real networks exist.

Bibliography

[1] Arijeet M., 2013, Agent-Based Traffic Simulation Using Microscopic Modeling.

[2] Barabási, Albert-László. "Scale-free networks: a decade and beyond." Science 325.5939 (2009): 412-413.

[3] Braess, Anna Nagurney, and Tina Wakolbinger. "On a paradox of traffic planning." Transportation science 39.4 (2005): 446-450.

[4]Braess, Priv-DozDr D. "ÜbereinParadoxonaus der Verkehrsplanung." Unternehmensforschung 12.1 (1968): 258-268..

[5] Cieplak, Marek, Amos Maritan, and Jayanth R. Banavar. "Optimal paths and domain walls in the strong disorder limit." Physical review letters 72.15 (1994): 2320-2323.

[6] Cohn, Nick, Edwin Kools, and Peter Mieth. "The TomTom Congestion Index of Europe 2013." 19th ITS World Congress. 2013.

[7] De Montis, Andrea, et al. "The structure of inter-urban traffic: A weighted network analysis." arXiv preprint physics/0507106 (2005).

[8] Easley, David, and Jon Kleinberg. Networks, crowds, and markets. Vol. 8. Cambridge: Cambridge University Press, 2010.

[9] Jha, Mithilesh, Samer Madanat, and Srinivas Peeta. "Perception updating and day-to-day travel choice dynamics in traffic networks with information provision." Transportation Research Part C: Emerging Technologies 6.3 (1998): 189-212.

[10] Larsson, Torbjörn, and Michael Patriksson. "Simplicial decomposition with disaggregated representation for the traffic assignment problem." Transportation Science 26.1 (1992): 4-17.

[11] Lämmer, Stefan, BjörnGehlsen, and Dirk Helbing. "Scaling laws in the spatial structure of urban road networks." Physica A: Statistical Mechanics and its Applications 363.1 (2006): 89-95.

[12] LeBlanc, Larry J., Edward K. Morlok, and William P. Pierskalla. "An efficient approach to solving the road network equilibrium traffic assignment problem." Transportation Research 9.5 (1975): 309-318.

[13] Lin, Wei-Hua, and Hong K. Lo. "Investigating Braess' Paradox with Time-Dependent Queues." Transportation Science 43.1 (2009): 117-126.

[14] Louf, Rémi, Pablo Jensen, and Marc Barthelemy. "Emergence of hierarchy in cost-driven growth of spatial networks." Proceedings of the National Academy of Sciences 110.22 (2013): 8824-8829.

[15] Peeta, Srinivas, and Athanasios K. Ziliaskopoulos. "Foundations of dynamic traffic assignment: The past, the present and the future." Networks and Spatial Economics 1.3-4 (2001): 233-265.

[16] Porta, Sergio, Paolo Crucitti, and Vito Latora. "The network analysis of urban streets: a dual approach." Physica A: Statistical Mechanics and its Applications 369.2 (2006): 853-866.

[17] Shiraishi, Tomoyoshi, et al. "Development of a microscopic traffic simulation model for interactive traffic environment." Proc. 11th World Congress on Intelligent Transport Systems and Services (ITSWC 2004), Nagoya, Japan. 2004.

[18] Strano, Emanuele, et al. "Elementary processes governing the evolution of road networks." Scientific reports 2 (2012).

[19] Watts, Duncan J., and Steven H. Strogatz. "Collective dynamics of ‘small-world’networks." nature 393.6684 (1998): 440-442.

[20] Youn, Hyejin, Michael T. Gastner, and HawoongJeong. "Price of anarchy in transportation networks: Efficiency and optimality control." Physical review letters 101.12 (2008): 128701.

[21] Zheng, Hong, et al. A Primer for Agent-Based Simulation and Modeling in Transportation Applications. No. FHWA-HRT-13-054. 2013.

Appendix A

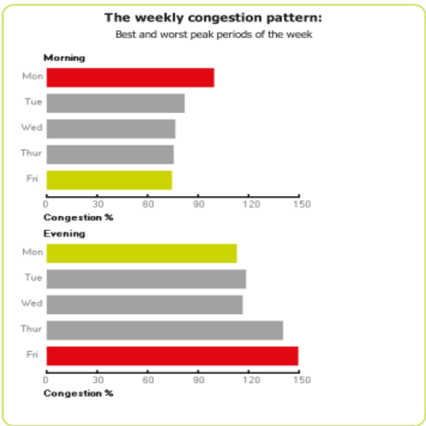
Istanbul

Congestion level

57%

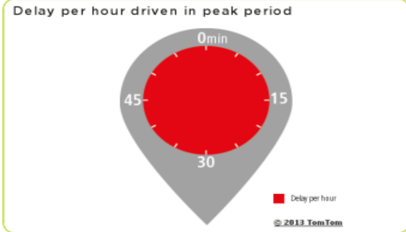
Ranking

Ranking of city compared to all cities in the report	2/59
Congestion level on highways	59%
Congestion level on non-highways	55%
Delay per hour driven in peak period	64 min
Delay per year with a 30 min commute	118 h

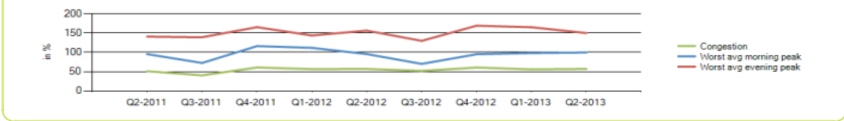


Most congested specific day	Thu 13 Jun 2013
Total network length	1 254 mi
Total network length highways	249 mi
Total network length non-highways	1 005 mi
Total vehicle miles	295 665 mi

Delay per hour driven in peak period



Comparison per quarter



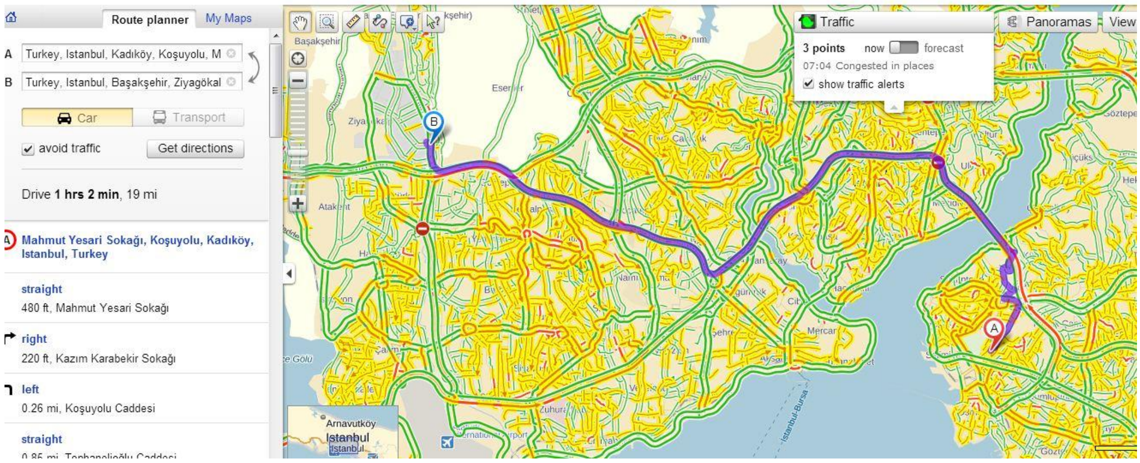
Top 10 cities

Rank	CI change	City	Country	Congestion	Morning peak	Evening peak	Highways	Non-Highways
1	▼	Moscow	Russia	65%	114%	133%	63%	66%
2	---	Istanbul	Turkey	57%	81%	127%	59%	55%
3	▼	Warsaw	Poland	44%	89%	95%	40%	49%
4	---	Palermo	Italy	40%	65%	67%	32%	47%
5	▼	Marseille	France	40%	74%	81%	25%	50%
6	▲	Rome	Italy	36%	84%	67%	28%	40%
7	▲	Paris	France	36%	77%	72%	35%	36%
8	▲	Stockholm	Sweden	36%	75%	85%	34%	38%
9	▲	Brussels	Belgium	34%	71%	92%	30%	37%
10	▲	Lyon	France	31%	66%	66%	27%	38%

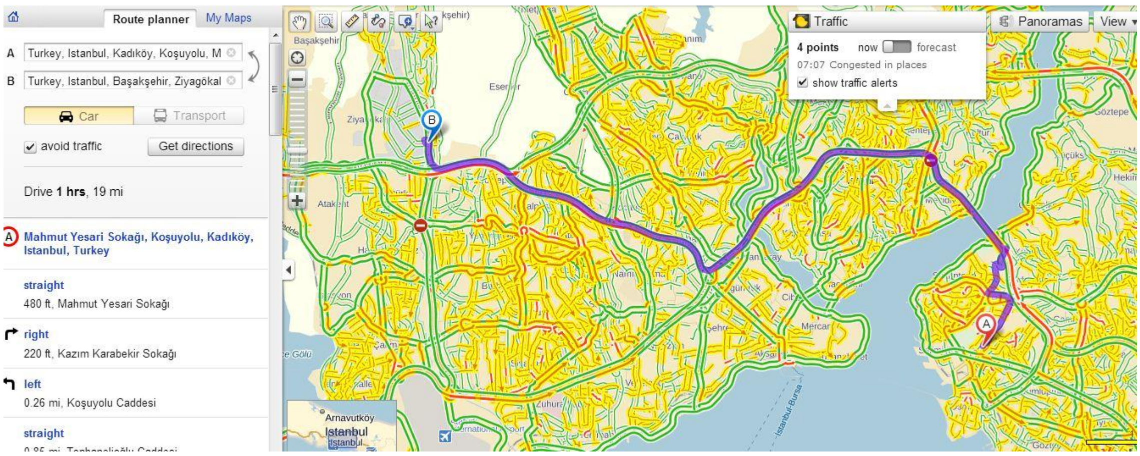
The Yandex experiment we have performed to show the congestion variation the traffic of Istanbul

Congestion caused by morning commuters is measured on 4/12/2013

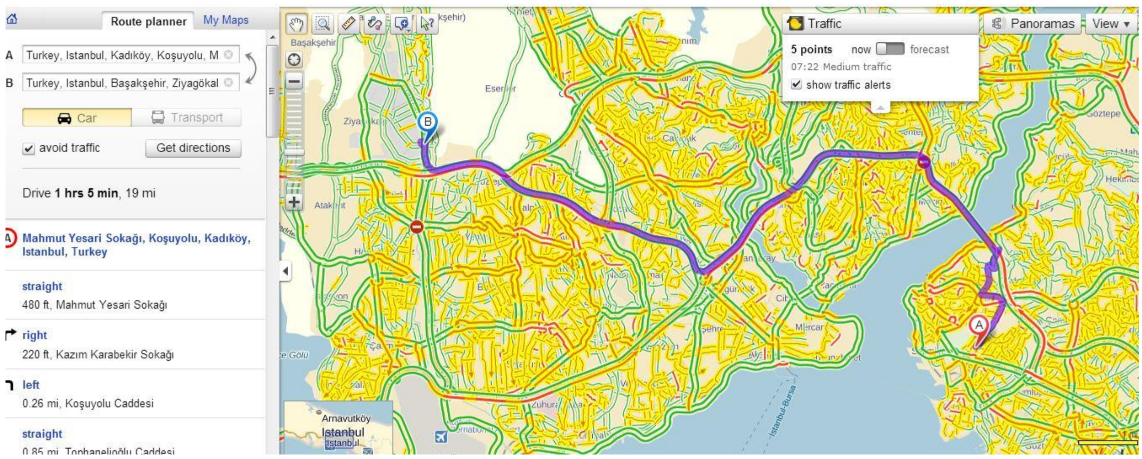
07:10



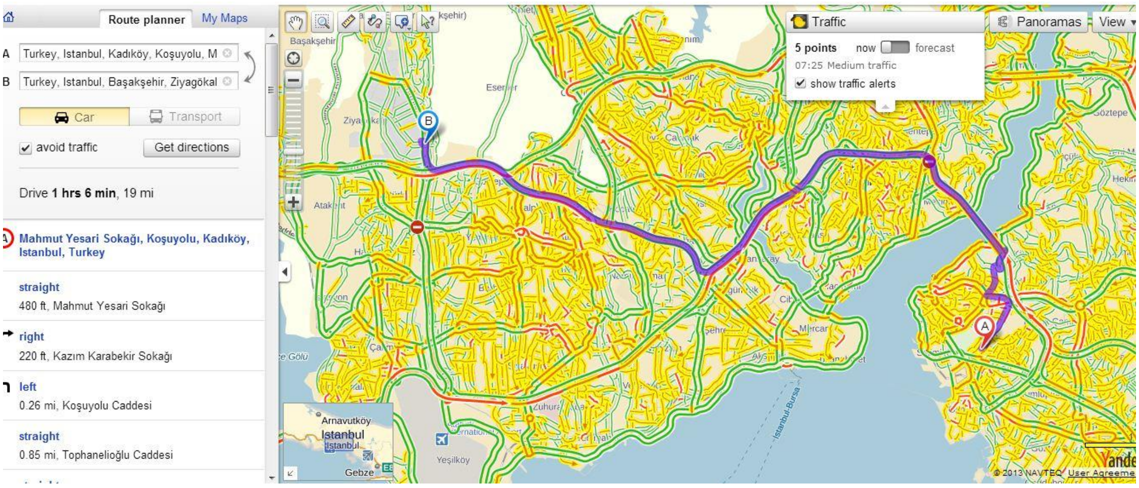
07:15



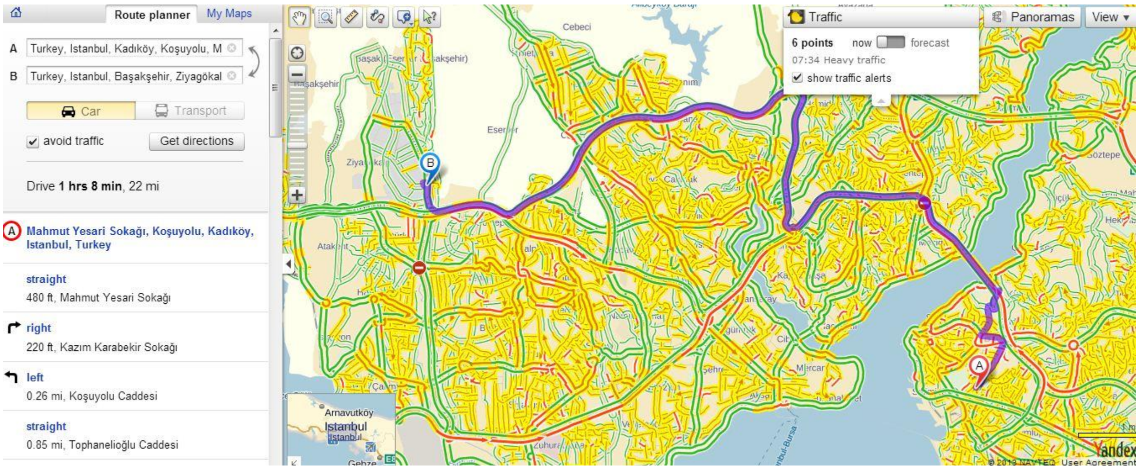
07:25



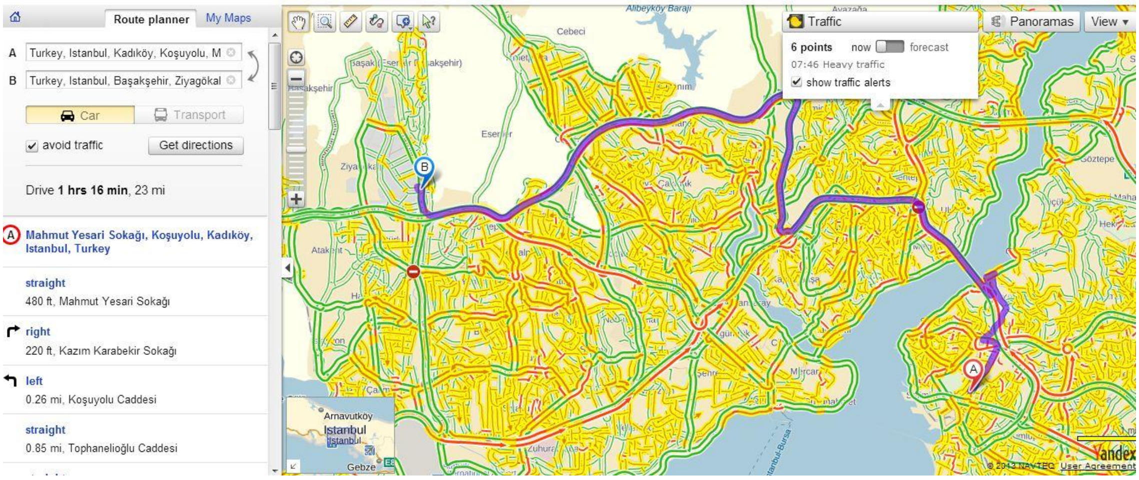
07:30



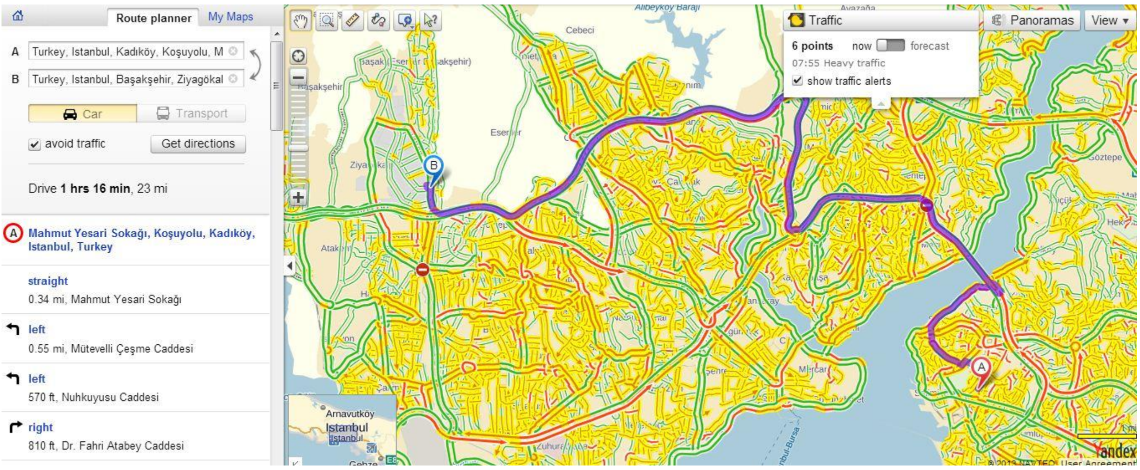
07:40



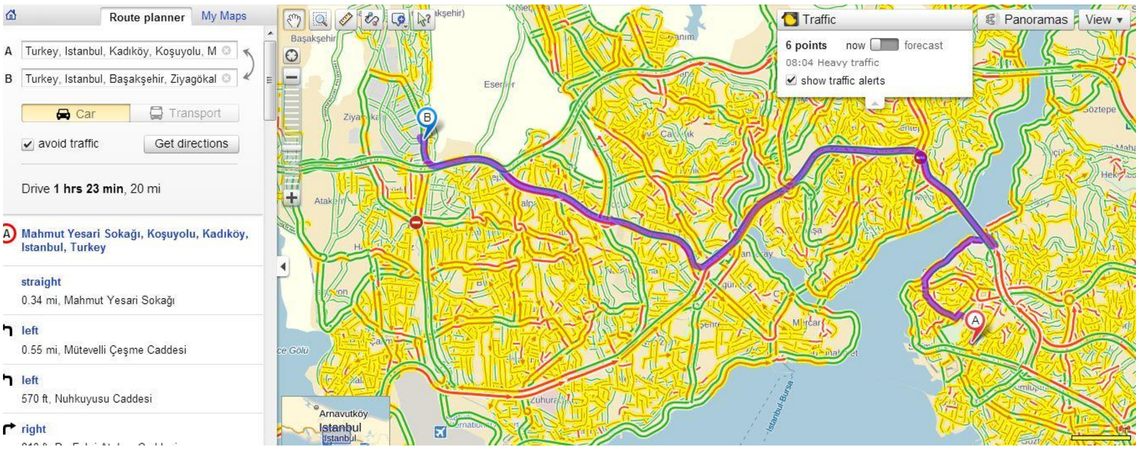
07:50



08:00

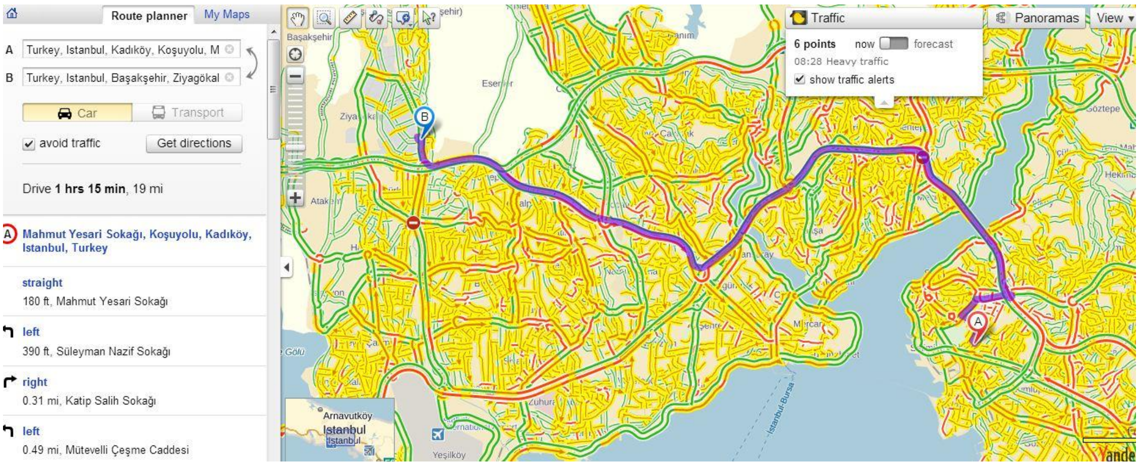


08:10



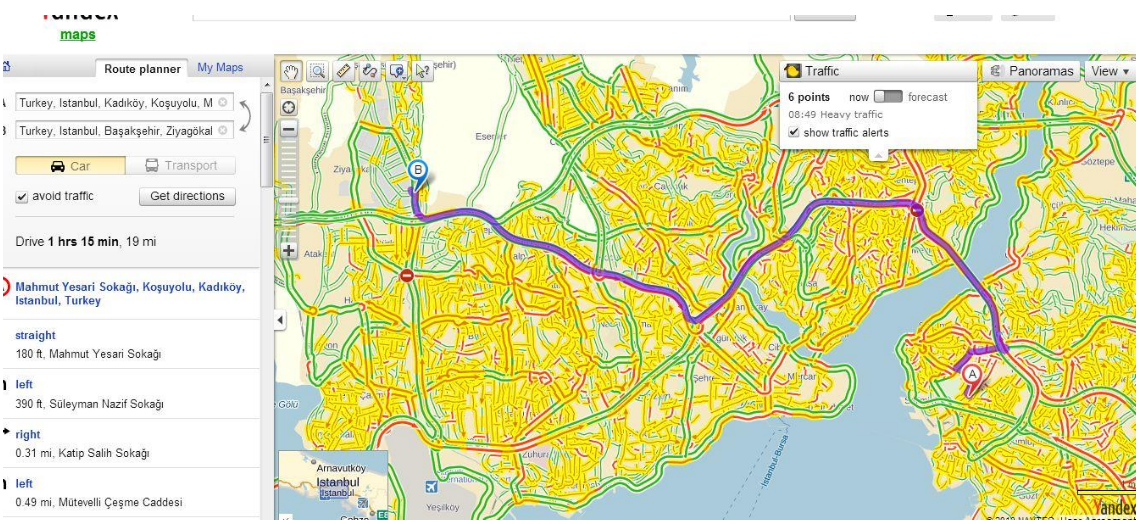
08:25

08:35

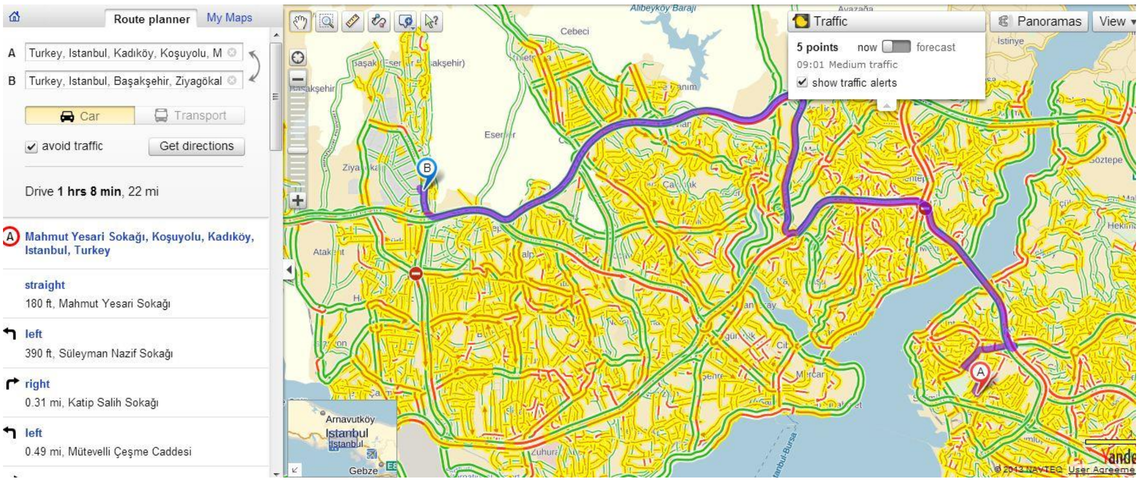


08:45

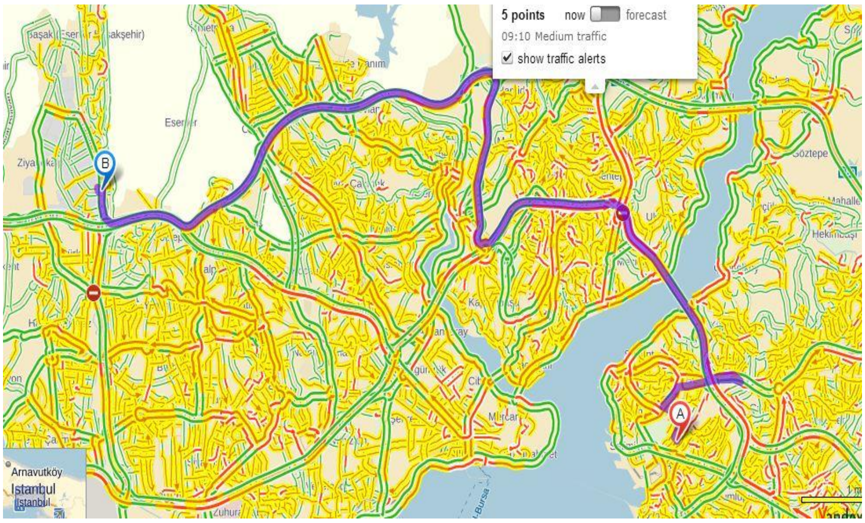
08:55



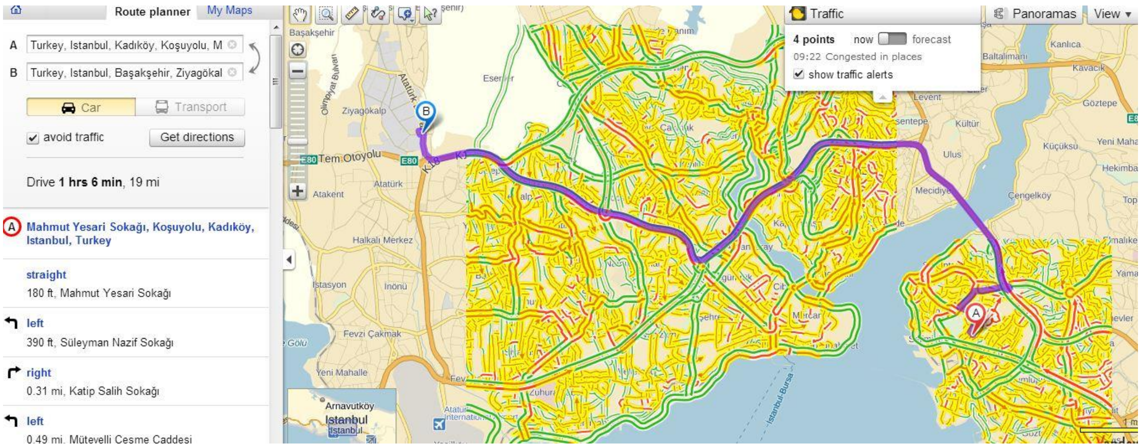
09:05



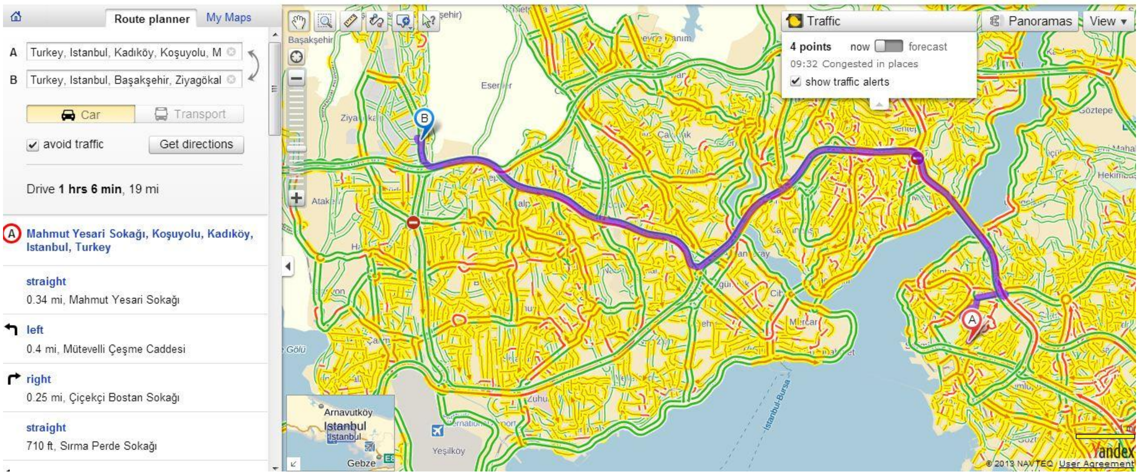
09:15



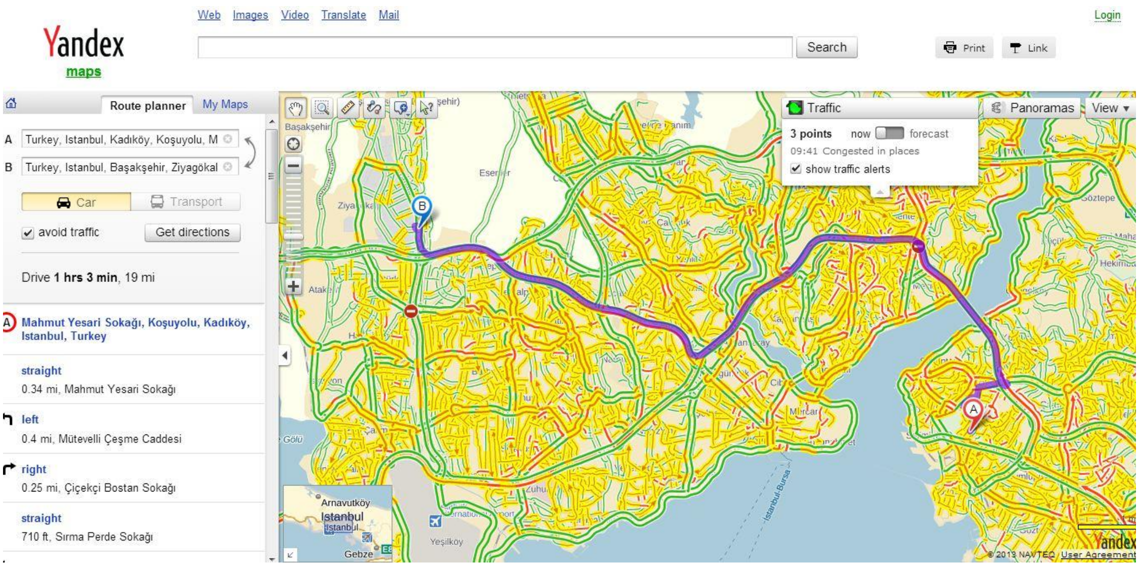
09:26



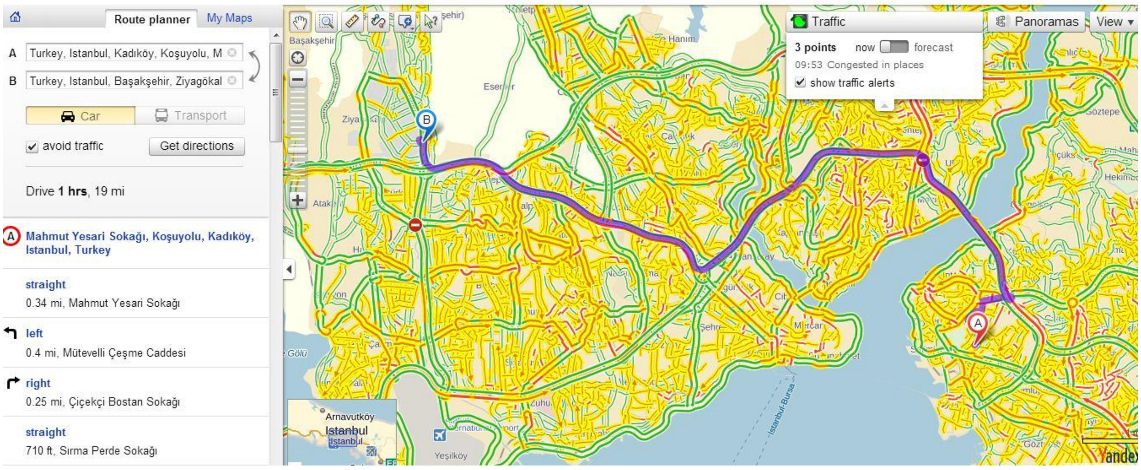
09:35



09:46

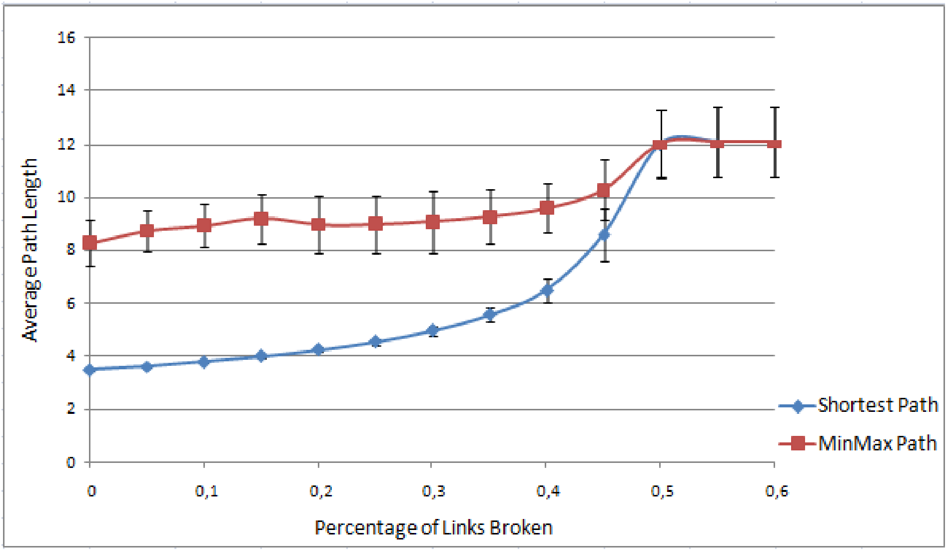


10:00

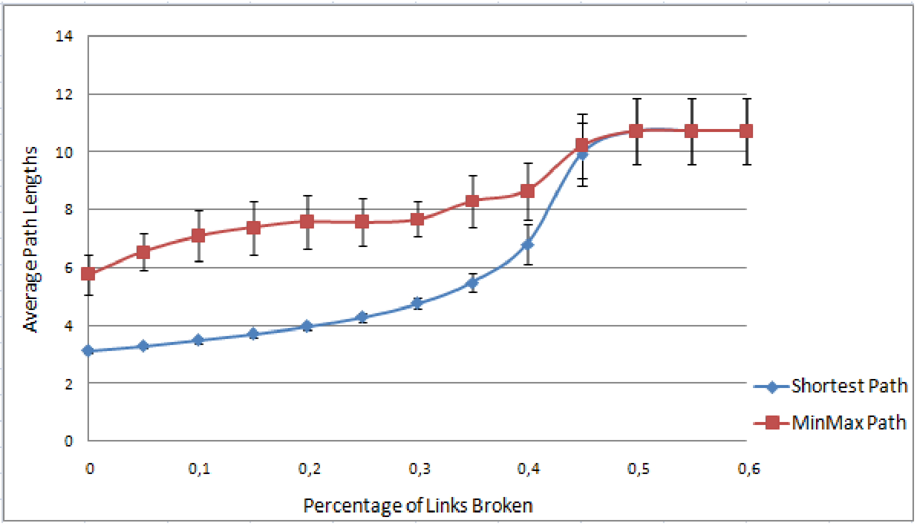


The congestion data of Istanbul network is less than the normal values, since Yandex map is adaptive to congestion.

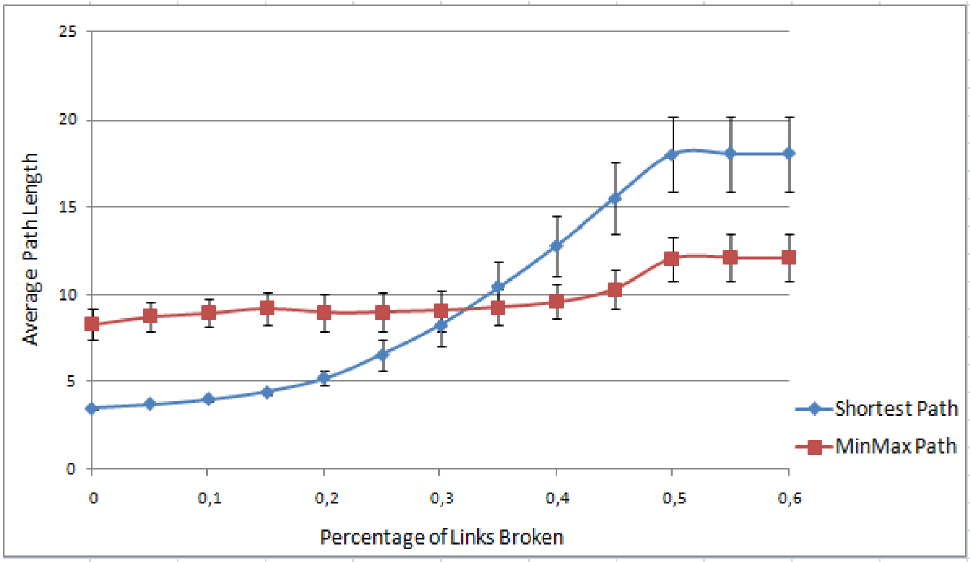
Appendix B



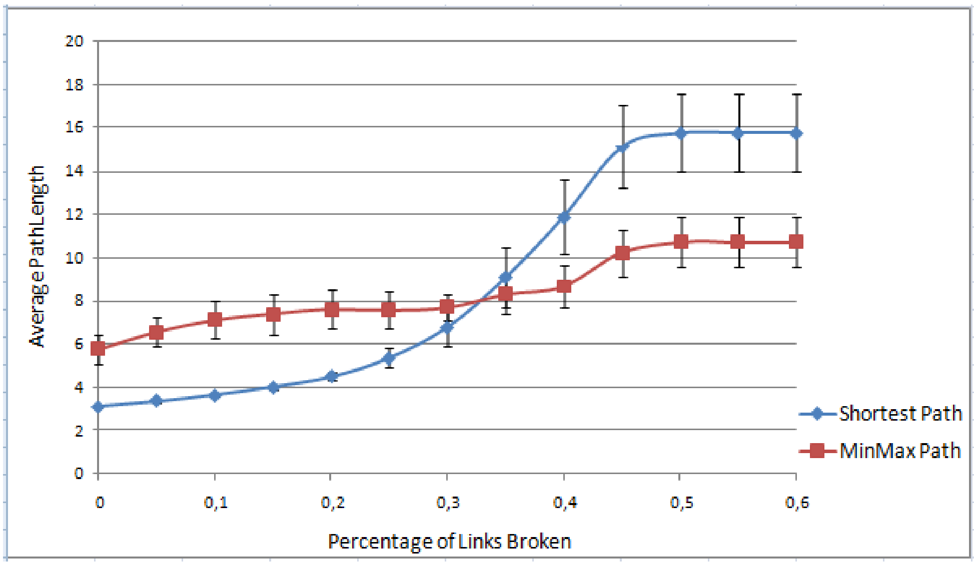
Appendix Figure F.1. The average path lengths of fifty undirected **Random Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to **min-max path betweenness centrality**. Error bars are placed with respect to standard deviations. The effective interval is between 25% and 30% of the closed roads



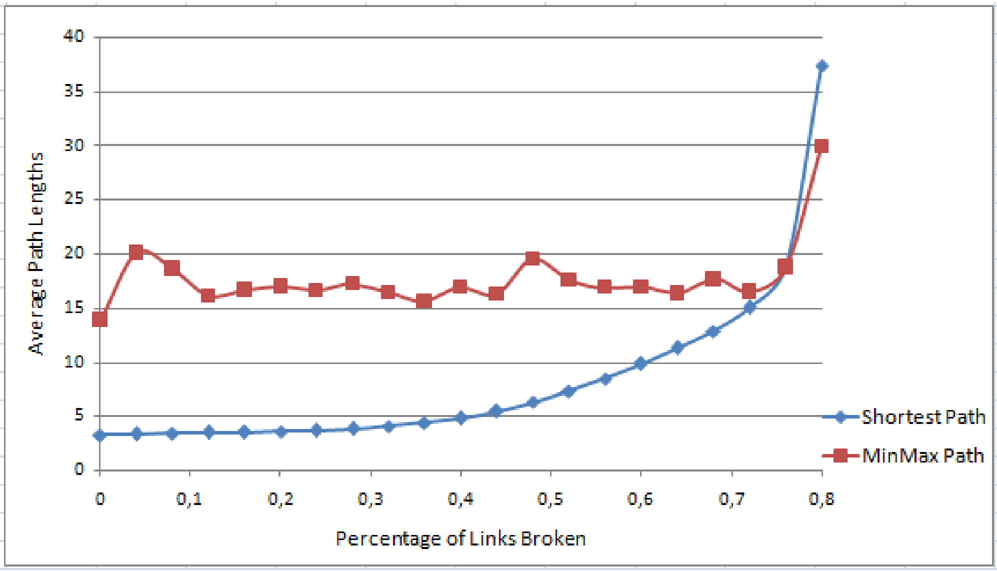
Appendix Figure F.2. The average path lengths of fifty undirected **Scale Free Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to **min-max path betweenness centrality**. Error bars are placed with respect to standard deviations. The effective interval is less than 5% of the closed roads



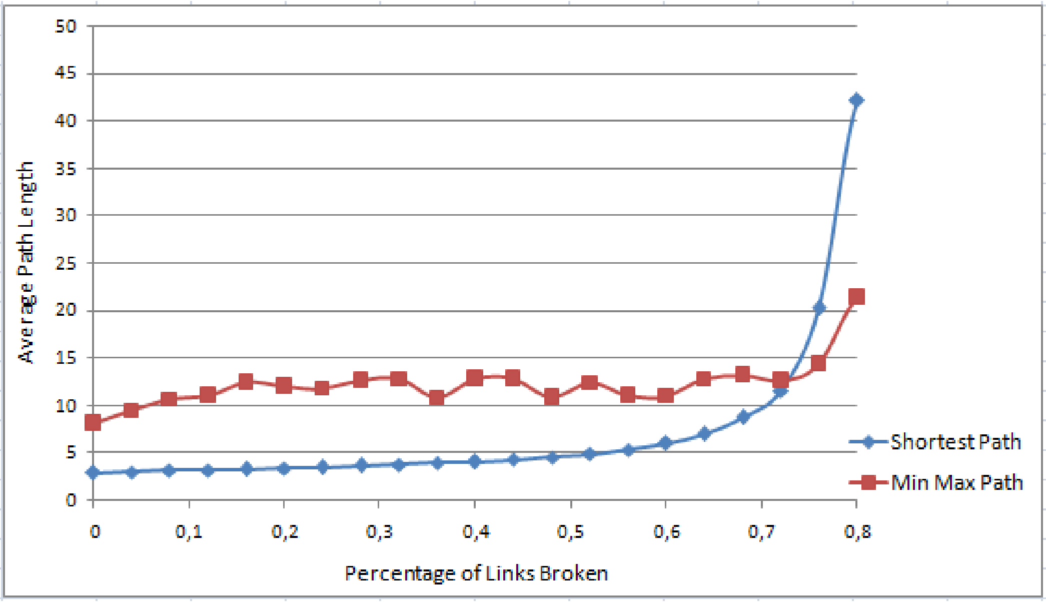
Appendix Figure F.3. The average path lengths of fifty undirected **Random Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths. Since the resulting networks are totally different, average Shortest Driven path lengths become more than min-max path lengths. The underlying reason is, the shortest path betweenness centrality road closure method always imposes the longer paths to remain on the network. Error bars are placed with respect to standard deviations. The effective interval is between 20% and 25% of the closed roads



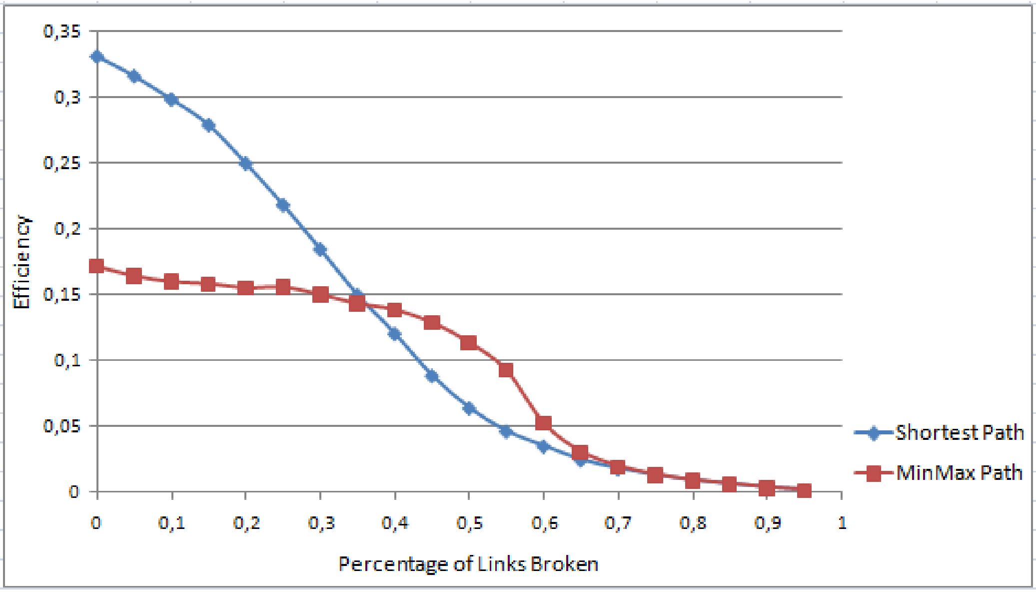
Appendix Figure F.4. The average path lengths of fifty undirected **Scale Free Networks** having 100 nodes and 4 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 5% of the roads are closed. Selecting the roads that are shut is decided according to **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths. Error bars are placed with respect to standard deviations.



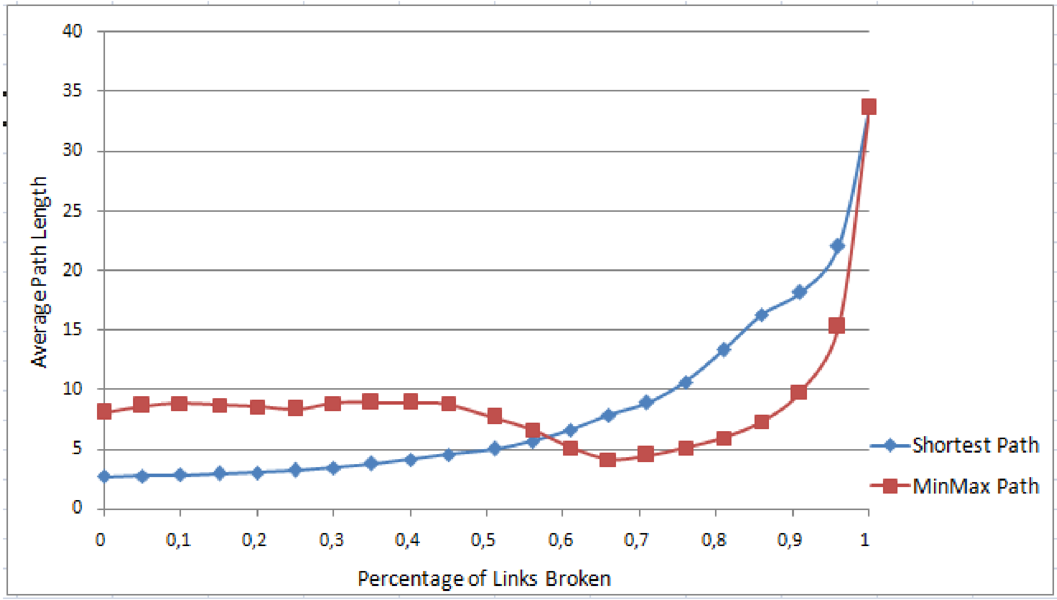
Appendix Figure F.5. The path lengths of an undirected **Random Network** having 1000 nodes and 10 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 10% of the roads are closed. Selecting the roads that are shut is decided according to **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths.



Appendix Figure F.6. The path lengths of an undirected **Scale Free Network** having 1000 nodes and 10 connections on average. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 10% of the roads are closed. Selecting the roads that are shut is decided according to the **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths.



Appendix Figure F.7. The average efficiency of ten undirected **Random Networks** having 100 nodes and 4 connections on average. Emerging disconnected nodes are allowed for this road closure method. The efficiency is defined as € = (1/PathLength). For the disconnected nodes PathLength -> ∞, € -> 0. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 10% of the roads are closed. Selecting the roads that are shut is decided according to the **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths.



Appendix Figure F.8. The average path lengths of ten undirected **Random Networks** having 100 nodes and 4 connections on average. All nodes are connected by a line initially, and closing a road which belongs to the line is prohibited. The weights are normally distributed. μ=1, σ=0.3. Average Shortest Driven path lengths and min-max path lengths are calculated after each 10% of the roads are closed. Selecting the roads that are shut is decided according to the **shortest path betweenness centrality** for average Shortest Driven path lengths, and decided according to **min-max path betweenness centrality** for average min-max path lengths.

Appendix C

PseudoCode 1:

```
LinkedList<Agent> AgentAtNode = new LinkedList<>();
Link[][] LinkMap = new Link[realMap.length][ realMap.length];
LinkMap = RealMap;
LinkDistThreshold = A;
while(t <ApplicationLife){
        Get.newAgents();
        AgentsAtNode.add(newAgents());
        for (int i = 0; I < linkMap.length ; i++){
                for(int j = 0; j < linkMap.length ; j++){
                        if(linkMap[i][j] != null){
                                for(int k = 0 ; k < linkMap[i][j].getLinkQueue.size; k++){
                                        if(linkMap[i][j].setLinkQueue.get(k).setLinkPos == 1){
                                                linkMap[i][j].getLinkQueur.get(k).setlinkpos--;
                                                for (int m=0; m < k ; m++){
                                                        linkMap[i][j].getLinkQueue.get(m).
                                                                        setlinkpos--;
                                                        }
                                                linkMap[i][j].getLinkQueue.get(k).
                                                                setProcessed(True);
                                                AgentsAtNode.
                                                   addFirst(linkMap[i][j].getLinkQueue.get(k));
                                                linkMap[i][j].getLinkQueue.remove(k);
                                                realMap[i][j]--;
                                                }
                                        }
                                }
                        }
```

```
for(i= 0 ; i < AgentsAtNode.size(); i++){

AgentsAtNode.get(i).setSeens();
}
for (int i = 0; i < AgentsAtNode.size();  i++){
        if( AgentsAtNode.get(i).getProcessed.isequals(True)){
                AgentsAtNode.get(i).setProcessed == False;
        }
        else{
                AgentsAtNode.get(i).setPath(findSP(source,dest,seenMap));
                if(AgentsAtNode.get(i).getMethod.isEqualsto("Combined") &&
                        getWMax(Path)-getWMin(Path) > LinkDistThreshold ){
                                AgentsAtNode.get(i).findMinMax(source,dest,seenMap)
                        }
        }
}
```

PseudoCode 2:

```
while(t  < ApplicationLife) {
        for all Links{
                for all Vehicles at FirstRow{
                        justArrived.set(True);
                        Link.to.allVehiclesatNode.add
                        RemoveVehicles at FirstRow;
                        realMap[Link.from][Link.to]--;
                        }
                FirstRow = queue.getFirst();
                Queue.removeFirst();
                }
        for all Nodes{

        for all VehiclesatNode{
                if ( vehicle.justArrived.equals (False)){
                        Find Shortest or MinMax Path
                        Vehicle.source.vehicletoNodeList.put (Path[1].vehicle)
                        }
        }
        for each List on vehicle to  NodeListMap{
                Divide the list to lists of size LinkWidth;
                Link.addLast(lists);
                realMap[link.From][link.To] = realMap[linkFrom][link.To] + sum(lists);
        }
}
```