SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

A NEW SHORTEST PATH ALGORITHM FOR MANAGEMENT URBAN TRAFFIC

M.Sc. THESIS

Amenah Sufyan Mhmood THABIT

Department	**: COMPUTER AND INFORMATION ENGINEERING**
Supervisor	**: Prof. Dr. Ahmet ZENGIN**

July 2021

SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

A NEW SHORTEST PATH ALGORITHM FOR MANAGEMENT URBAN TRAFFIC

M.Sc. THESIS

Amenah Sufyan Mhmood THABIT

Department	**:**	**COMPUTER AND INFORMATION ENGINEERING**

This thesis has been accepted unanimously / with majority of votes by the examination committee on 16/07/2021.

Head of Jury	**Jury Member**	**Jury Member**

DECLERATION

I declare that all the data in this thesis was obtained by myself in academic rules, all visual and written information and results were presented in accordance with academic and ethical rules, there is no distortion in the presented data, in case of utilizing other people’s works they were refereed properly to scientific norms, the data presented in this thesis has not been used in any other thesis in this university or in any other university.

Amenah Sufyan Mhmood THABIT

14.06.2021

DEDICATION

To my dear father and mother who has always been on myside supporting, loving and caring…

To my husband Marwan for his endless love and support…

To my brothers and sisters who have been my source of inspiration …

ACKNOWLEDGMENTS

First and foremost, I thank the Almighty Allah who has provided me with the strength and patience to accomplish this work.

I am greatly indebted to my supervisor Dr. Ahmet ZENGIN who devoted his precious time to help and offer some advice, valuable information, and excellent insight throughout the development of this thesis.

My great appreciation goes to the Iraqi Parliament who provided me with the financial support needed to complete this research.

Special thanks go to the staff members in the Computer and Information Engineering department at the Sakarya University for their encouragement, motivation, kindness, and understanding.

Many thanks also go to all my colleagues and friends, precisely the Iraqi.

Finally, I am deeply indebted to my husband for supporting me throughout my academic journey.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

DSRC	: Dedicated Short-Range Communications
FCD	: Floating Car Data
GPS	: Global Positioning System
GSM	: Global System for Mobile Communication
GUI	: Graphic User Interface
ITS	: Intelligent Transportation System
IVC	: Inter-Vehicle Communication
LOC	: Level-Of-Congestion
LOS	: Level-Of-Service
OD-matrices	: Origin-Destination-Matrices
OSM	: Open Street Map
PVD	: Probe Vehicle Data
RFID	: Radio Frequency Identification
SUMO	: Simulation of Urban Mobility
TMS	: Transit Management Systems
UTM	: Urban Traffic Management
UWB	: Ultra-Wide Band
VVD	: Video Vehicle Detection
WAVE	: Wireless Access Vehicular Environment
WHO	: World Health Organization
Wi-Max	: World Interoperability for Microwave Access
Wi-Fi	: Wireless Fidelity
WPAN	: Wireless Personal Area Networks
XML	: Extensible Markup Language

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Keywords: Traffic congestion, shortest path, Dijkstra algorithm, SUMO

Nowadays, traffic jam is an interminable obstacle for the transportation growth in urban cities around the world. One of the major causes of traffic jam is raising of vehicles number due increment population, mainly in areas with bottlenecks and it also causes for society and economic losses, increase in greenhouse emissions and health damages. Furthermore, it targets communities in the most critical element of life which is the human element. Therefore, it has become essential to work to find solutions, projects and put them into practice to address the causes and mitigate the negative effects.

This thesis focuses on improved Dijkstra algorithm based on traffic congestion level. Improved Dijkstra algorithm can provide (a) real data collection from map via OpenStreetMap (b) Adding four features to SUMO simulator software (time period, rush-hour, number of vehicles, and routing algorithm) (c) it has the ability to know congestion level for roads (d) rerouting vehicles to avoided traffic congestion.

Based on the simulation results and analysis presented in the thesis, it was found that the proposed improved Dijkstra algorithm increased the performance of the road traffic flow by reducing the number of related vehicles in traffic congestion and average delay time for experiments scenarios. The improved Dijkstra algorithm has decreased related vehicles and delay time by approximately 23%. On the other hand, the improved Dijkstra reduce average fuel consumption and amount of CO_2 emission by approximately 15% and 14% respectively.

The future works is to examine the performance of improved Dijkstra algorithm on the real road traffic environment and to employ (VANTE) technology in vehicle route guidance systems, which will play a vital role in alleviate traffic congestion by exchanging traffic information between vehicle to vehicle and the vehicle to infrastructure.

The contributions and findings of this thesis may support urban planners, mobile operators, transport planners, civil engineers, and traffic congestion management researchers and in addition reduce environmental pollution.

ŞEHİRİÇİ TRAFİK YÖNETİMİ İÇİN YENİ BİR EN KISA YOL ALGORİTMASI

ÖZET

Anahtar Kelimeler: Trafik sıkışıklığı, en kısa yol, Dijkstra algoritması, SUMO

Günümüzde trafik sıkışıklığı, dünyadaki kentsel şehirlerde ulaşımın büyümesi için önemli bir engeldir. Trafik sıkışıklığının en önemli nedenlerinden biri, özellikle dar boğazlı bölgelerde artan nüfusa bağlı olarak araç sayısının artması ve toplum ve ekonomik kayıplara, sera gazı emisyonlarının artmasına ve sağlık zararlarına neden olmasıdır. Ayrıca hayatın en kritik unsuru olan insanı ve topluma zarar verir. Bu nedenle, olumsuz etkileri azaltmak için çözümler, projeler bulmak ve bunları uygulamaya koymak amacıyla çaba göstermek son derece önemlidir.

Bu tez, trafik sıkışıklığı düzeyine dayalı olarak geliştirilmiş Dijkstra algoritmasını geliştirmeye odaklanmaktadır. Geliştirilmiş Dijkstra algoritması aşağıdaki özelliklere sahiptir:

a. OpenStreetMap aracılığıyla haritadan gerçek veri toplayabilir
b. SUMO simülatör yazılımına dört özellik eklenmiştir (zaman aralığı, trafik yoğunluğu, araç sayısı ve yol algoritması)
c. yollar için tıkanıklık seviyesini hesaplayabilir ve
d. araçların trafik sıkışıklığından kaçınılacak şekilde yeniden yönlendirilmesini sağlayabilir.

Tezde sunulan simülasyon sonuçları ve analizlere dayalı olarak, önerilen Dijkstra algoritmasının, trafik sıkışıklığındaki ilgili araç sayısını ve deney senaryoları için ortalama gecikme süresini azaltarak karayolu trafik akışının performansını arttırdığı tespit edilmiştir. Geliştirilmiş Dijkstra algoritması, ilgili araçları ve gecikme süresini yaklaşık %23 oranında azaltmıştır. Öte yandan, geliştirilmiş Dijkstra, ortalama yakıt tüketimini ve CO_2 emisyon miktarını sırasıyla yaklaşık %15 ve %14 oranında azaltmaktadır.

Gelecekteki çalışmalar, gerçek karayolu trafik ortamında geliştirilmiş Dijkstra algoritmasının performansını incelemek ve araç rota yönlendirme sistemlerinde bu teknolojiyi kullanmaktır. Bu çalışma, araçtan araca ve araçtan altyapıya trafik bilgisi alışverişinde bulunarak trafik sıkışıklığının hafifletilmesinde hayati bir rol oynayacaktır.

Bu tezin katkıları ve bulguları, şehir plancılarına, mobil operatörlere, ulaşım planlayıcılarına, inşaat mühendislerine ve trafik sıkışıklığı yönetimi araştırmacılarına destek olabilir ve ayrıca çevre kirliliğini azaltabilir.

CHAPTER 1. INTRODUCTION

With the continuous progress of urbanization in world, the vehicles are still rising with increase population has resulted in many worried issues such as traffic congestion, rising death rates in road incidents and increasing economic cost to drivers, as well as air pollution [1].

Depending onto the TOMTOM traffic Analytics in 2020, where direct highlights congestion levels in 416 cities in 57 countries. Although on continuous increment in traffic congestion all through the last decade were brought to an unexpected stop by the Covid-19 widespread which has turned life patterns and transportation direction upside down over in the world. By contrast, 13 cities saw their traffic jams increase as shown in Figure 1.1. [2].

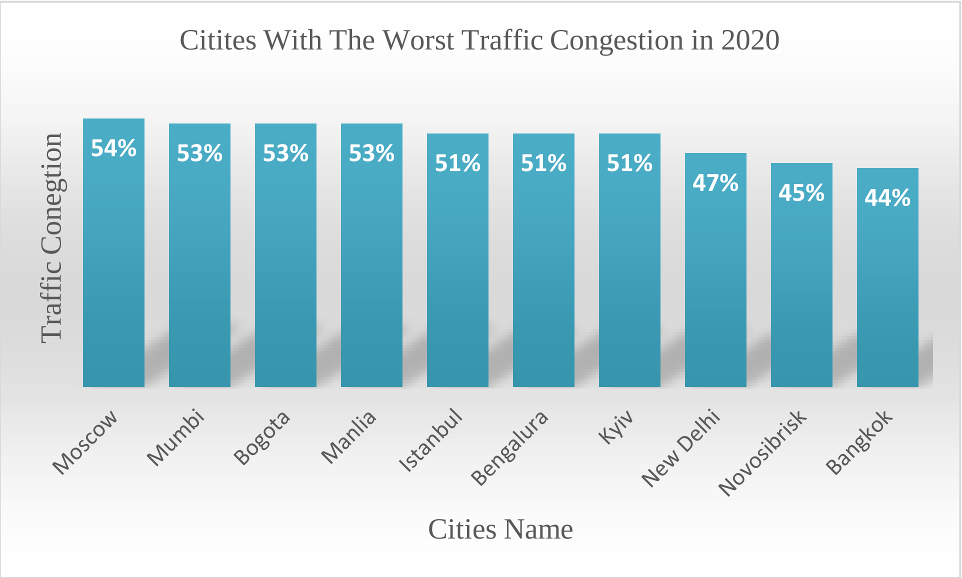


Figure 1.1. The worst traffic jam in 13 cities [2]

From Figure 1.1., shows the Moscow is a top spot then a three-way balance for second spot with Mumbai in India, Bogota in Colombia and Manila in the Philippines all registered congestion levels around of 53 percent, after that three-way balance also with Istanbul in Turkey, Bengaluru in India and Kyiv in Ukraine came in levels of 51 percent as consideration 0% free congestion and 35% is traffic congestion.

In the year 2020, the World Health Organization (WHO) published that annually, more than one million people die on the roads in world, and that 20 to 50 million suffer non-fatal injuries, Economic costs associated with these crashes are estimated at between 1% and 3% of a country's local product. As per current estimates, around 90% of road accidents are caused by personal error, inattention, fatigue or delayed reactions [3].

There are many conventional solutions to alleviate road traffic problems represented by building and expanding roads, increasing and encouraging people to use public transport, and constructing more car parks. Although these are considered possible solutions, their implementation consumes too much time, effort and money. Therefore, one alternative solution is to employ the new technologies in the field of traffic management system (TMS) that considered to be one of the most important pillars of an Intelligent Transportation System (ITS) [3].

1.1. Research Motivation

Traffic problems, including traffic congestion, are one of the most important international phenomena that the countries of the world experience as the tax of technological and civilized development and progress in our present era. Traffic problems are becoming increasingly complex and are a cause of worry for Governments. Therefore, the first and main motivation of this research comes from the fact that the researcher lives in Mosul city, Iraq, which is the second largest city in Iraq. Mosul suffers from increasing traffic congestion due to the rise in population, number of vehicles on the road network and the lack of a traffic management system.

The researcher experiences this problem every morning when she goes to the work and sees how traffic congestion impacts on traffic flow and road incidents. decided to select this topic with a few of assisting and enhancing road traffic performance, reducing traffic congestion and providing a safe and comfortable journey for drivers.

The second motivation is the need to conserve our environment and reduce pollution produced by vehicles on the road network.

1.2. Problem Description and Solution Proposed

Traffic jam is one of the most challenging matter in urban cities that generate many problems such as time delay, fuel wastage and air pollution due to emission of CO2 and other pollutants in air which results in different health danger. Additionally, it also alters the human life by increment level of stress. To alleviate the above problems, this study proposed improved Dijkstra's algorithm via adding traffic congestion level factor and reroute vehicles to arrive same destination.

This study has faced various technical challenges during improved Dijkstra algorithm, the main challenges are listed below:

a. Understanding OpenStreetMap (OSM): to utilized the part of the map through use “Export” button that would like to download data for and save the data in OSM file.
b. Simulation for Urban Mobility (SUMO): have need to full grasp of the XML syntax to implement vehicle behavior which can easy to understand of SUMO as any missing connections when generating the selected route ID can make the simulation crash and generate errors.
c. MATLAB: is most challenge for main processing in this study via read SUMO output files and utilized for improved Dijkstra algorithm.

1.3. Research Aim and Objectives

This study aims to improve Dijkstra algorithm for traffic management system in urban cities to find optimum path in case of a traffic jam. The improved Dijkstra algorithm can (a) provide real data collection from map via OpenStreetMap (b) it has the ability to know congestion level for roads, (c) provide an alternative route for vehicles based on real-time, and (d) it has the characteristic of working under different vehicles. The objectives of this study are:

1. To explore technologies and application that are used in the Intelligent Transportation System.
2. To explore the literature on road traffic congestion concepts as well as, review the methods that have used in traffic congestion detection and avoidance.
3. To explore the literature on Dijkstra algorithm and modification that utilized to find optimum path on the roads.
4. To build traffic congestion parameter and utilized to improved Dijkstra algorithm.
5. Simulating data using SUMO simulators to evaluate the performance of the proposed improved Dijkstra algorithm.
6. Comparison classic Dijkstra and improved Dijkstra in traffic congestion based on performance metrics such as the number related vehicles in traffic jam and average delay time.

1.4. Research Contributions

The research contributions presented in this thesis can be summarized as follows:

1. Extensive literature review on traffic congestion detection and avoidance systems and modification Dijkstra algorithm.
2. Enhancement Dijkstra algorithm that helps drivers to reach their destination quickly and safely by suggesting optimal alternative routes to avoid road congestion on the road network.
3. Building a road traffic mobility simulator using appropriate software that can deal with any city/county roadmap.

1.5. Thesis Structure / Outlines

This study is organized into five chapters including the current chapter. This section provides an overview of each following chapters.

Chapter 2: Urban Traffic Management: Background and Literature - provides a general concept of urban traffic management including an introduction to Intelligent Transportation System and its technologies and applications and literature review of traffic congestion with cover detection and avoidance methods on the road network. Also, literature review on previous research studies concerned modification of Dijkstra algorithm.

Chapter 3: System Overview and Research Methodology – This chapter gives a brief description of tools that have used to improve Dijkstra algorithm. Then the methodology of the study is also described. including the matrices generation, the calculating traffic congestion level, and the integrated framework. the traffic information collection method, and the reroute vehicles algorithm, are also demonstrated in this chapter.

Chapter 4: Simulation and Evaluation Results – The chapter begins by explaining the contributions that have added to the SUMO simulation software. Then it gives a detail of the experiments scenarios that are conducted over the selected road network topologies including the scenario's determinants, simulation scenario initialization, and simulation parameters. The simulation results for both classic Dijkstra and improved Dijkstra in traffic congestion case have been also discussed in terms of the average delay time and the number of related vehicles. Finally, the performance of the improved Dijkstra is evaluated by comparing it with classic Dijkstra algorithm.

Chapter 5: Conclusions – presents the conclusions of the study and the recommendations for further research directions.

CHAPTER 2. BACKGROUND AND LITERATURE

Urban traffic is a real dilemma and attach to the social and economic activities of the city even deemed the lifeblood of urban evolution. Wherefore it has become one of the general affair concerned by the government and the whole society [4].

Many research has been done on this issue; the problem has often been dealing with by management of real capacity rather than the conventional concept of more road building [6]. This demands efficient traffic management tools and implementation traffic control systems that one of the important technical that regulating traffic flow, improving safety, and decreasing emissions within Urban Traffic Management (UTM) environment [5], [6].

An urban traffic management system (UTM) is a tool that can integrate data from several sources, manage the data in a database, and utilize this information in real-time to manage traffic in a manner that can lessen traffic congestion [8].

The objective of UTM is to create and execute an intelligent traffic management system that will detect and avoid congestion, take care of emergencies, keep cars safe and avoid accidents. More importantly, it aims to reduce gas emissions, fuel consumption and energy consumption in vehicles in order to make transportation greener. It contributes by 27% to pollution represented by carbon dioxide emissions, and it is considered as the fastest growing segment of carbon dioxide emissions in developing countries [7]. A conceptual of UTM has illustrated in Figure 2.1.

This chapter provides an introduction of UTM and in Section 2.1 present a brief description of Intelligent Transportation Systems, major areas of ITS deployments, the technologies and applications that are used with ITS. In Section 2.2 provides definitions of traffic congestion, causes of traffic congestion, and traffic congestion measurement metrics, as well as this section provide of methods that are used in traffic congestion detection, avoidance, and literature review. Section 2.3 presents shortest path algorithms, Dijkstra algorithm with example, and the literature for Dijkstra algorithm.

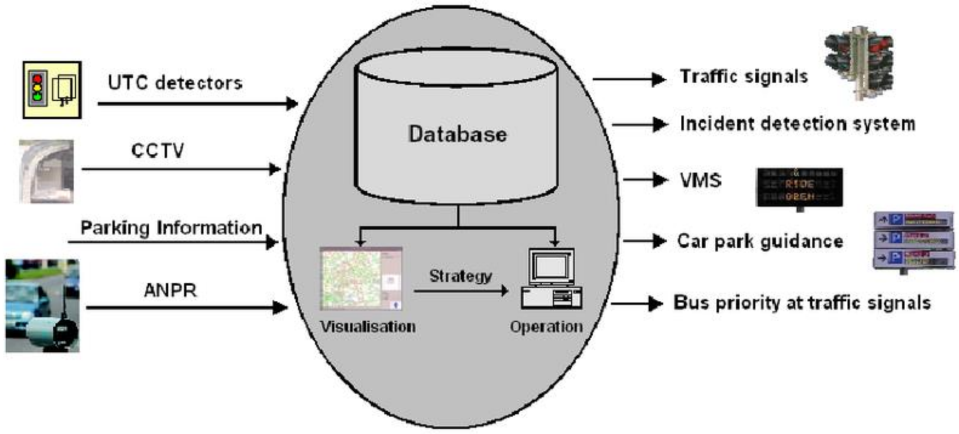


Figure 2.1. Urban traffic management systems [8]

2.1. Intelligent Transportation Systems (ITS)

The ITS system is a collection of application that is designed to provides services for transportation and traffic management through intelligent information and communication technology. A major benefit of ITS is the reduction of many urban problems like air pollution, long driving times, fuel consumption, traffic jams, accidents, safety, and high maintenance costs which have increased dramatically in recent years due to the growth of automobile use [9], [10].

In the world of artificial intelligence, machine learning, communication, the Internet, and many other emerging areas of engineering and information science, computer science concepts become the foundation for ITS, which provides sophisticated tools, combines advanced technologies and using them in a transportation system which is lead to solutions that will improve quality of life [13]-[15]. In addition, two major motivations for transportation's future are national productivity and international competitiveness. both closely interface to the effectiveness of transportation system [11]. Figure 2.2. shows a typical ITS (Intelligent Transportation Systems).

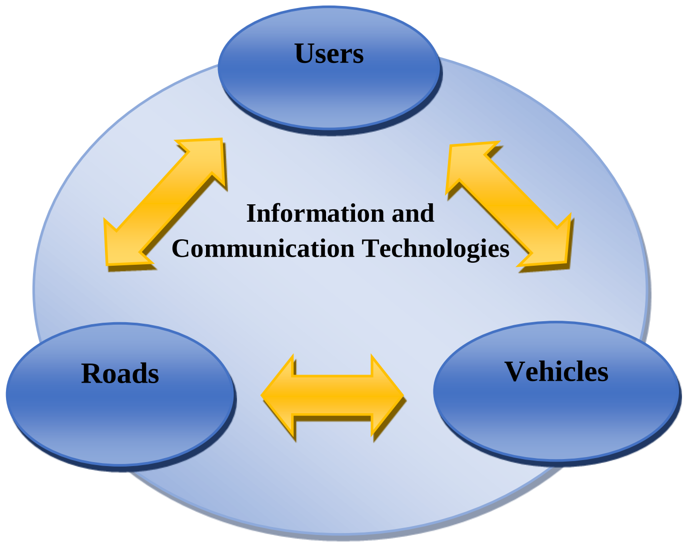


Figure 2.2. Intelligent transportation system (ITS)

The first example of an Intelligent Transportation System was the electric traffic signals implemented in 1928. The evolution chronology of ITS has taken place through three stages; Preparation (1930-1980), feasibility of the study (1980-1995), and product development (1995-present) [12]. Major areas of ITS in Metropolitan deployments are following:

1. Arterial and Freeway Management system
2. Freight Management system
3. Transit Management Systems (TMS)
4. Incident Management Systems
5. Emergency Management Systems
6. Regional Multimodal and Traveler Information Systems
7. Information Management (IM) Systems.

2.1.1. Intelligent transportation system technologies

Due to emergence of many technologies, this study listed intelligent transportation technologies as follows:

2.1.1.1. Wireless communications technologies

Recently, wireless communication technology has evolved rapidly and started to have a significant impact on the performance of transportation systems by alleviating traffic congestion and increasing road safety by reducing the number of road accidents. The present section highlights the technologies that are used with ITS, which can be categorized into three groups based on range; Short range, Medium range, and Long range [12].

a. Short range communication

Short range ITS technologies are characterized Short Range Communication fast network acquisition, low latency, high reliability, priority, interoperability, security and for privacy in transportation system but for short distances. They are referred to as Wireless Personal Area Networks (WPAN). The examples for short range are ZigBee, Bluetooth and Ultra-Wide Band (UWB).

b. Medium range communication

These technologies might have a vital role in the future of ITS, especially with vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications like Wi-Fi (Wireless Fidelity), DSRC (Dedicated Short-Range Communications) and WAVE (Wireless Access Vehicular Environment).

c. Long range communication

This kind of technology has the ability to provide coverage over a large area, where it can be used to exchange data amongst infrastructures of Intelligent Transportation Systems. These technologies are expected to have promising solutions regarding traffic congestions and accident warning. The example for Long range communication is GSM (Global System for Mobile Communication) and Wi-Max (World Interoperability for Microwave Access).

2.1.1.2. Sensing technologies

In transportation systems, sensors have embedded, wireless sensor nodes are usually low-cost and low power, and there are also capabilities for data processing and wireless communications such as microchips and RFIDs (Radio Frequency Identification) etc. There are two types of infrastructure sensors: indestructible (like in-road reflectors) and devices built into or above the road surface (such as on buildings, posts, and signs) which is easily maintained through manual distribution or automated injection machinery. Thus, timing and synchronization are important factors for such systems. The sensor has a variety of capabilities, including monitoring the temperature, humidity, movement of the vehicle, pressure, noise, speed, and direction. A sensor node is consisting main four components, Power Unit, transceiver, Sensing and processing unit [13]. Figure 2.3. shows the Sensor node architecture.

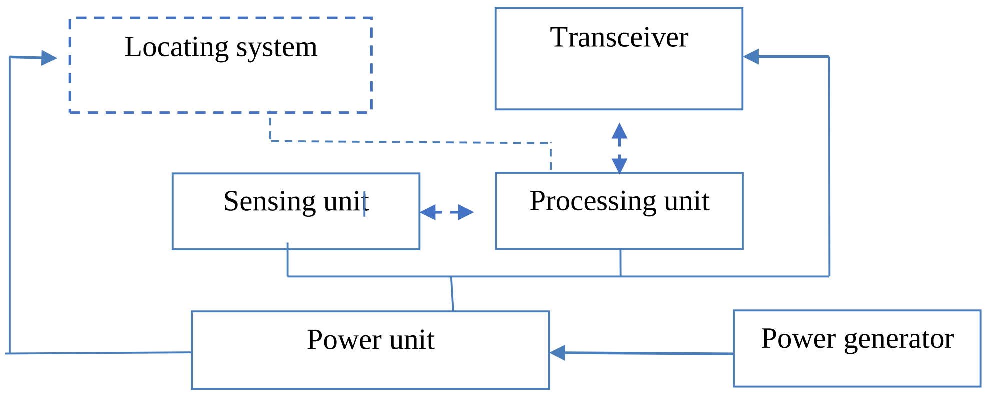


Figure 2.3. Sensor node architecture

2.1.1.3. Inductive loop detection

It is a type of detection system that uses magnets to activate electrical currents in a wire that can be used for various kinds of signal reception, including vehicle detection, vehicle passage, count, and occupancy. Today, inductive loop detection is still the most widely applied method of vehicle detection in traffic and parking applications. They work with vehicles moving at varying speeds or stopped at different points, and can be arranged in one lane or across many lanes [12]. Figure 2.4. shows the inductive loop detection.

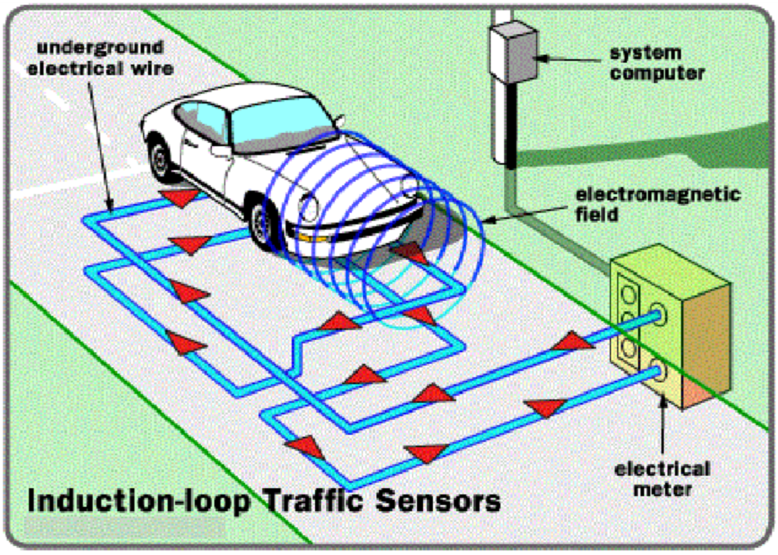


Figure 2.4. Inductive loop detection [14]

2.1.1.4.Video vehicle detection (VVD)

An effective method of detection is VVD, which has four components: video imaging, proper cabling, video image processing unit, and vision software. Among the features of the system is low cost for data collection, analysis, and accuracy, as well as improving driver capability and safety [15]. Figure 2.5., shows the video vehicle detection.

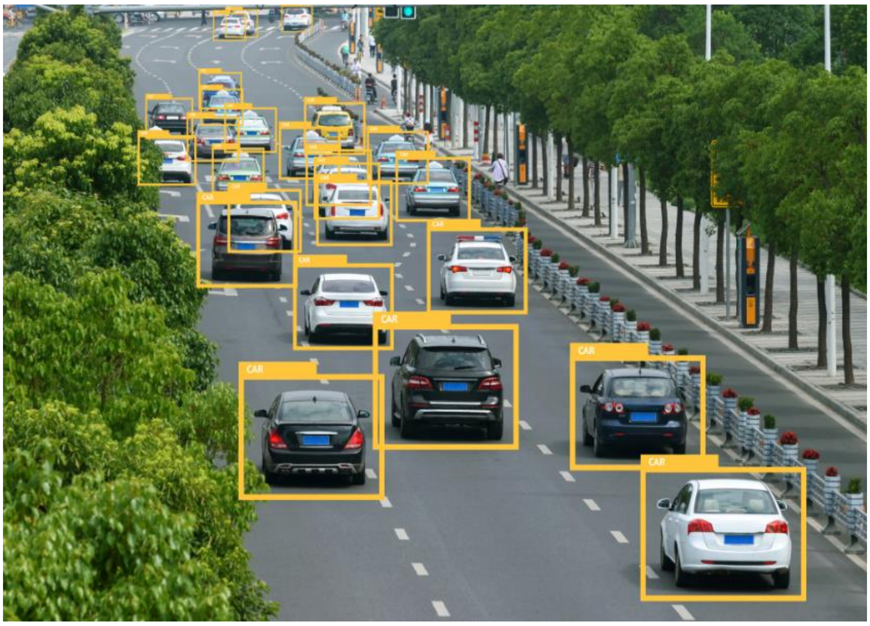


Figure 2.5. Video vehicle detection [16]

2.1.1.5. Bluetooth detection

By placing wireless sensors along the route, Bluetooth devices calculate travel times and deliver data to various networks starting from the origin to the destination. With Bluetooth, it is cheap and delicate to calculate travel time and analyze start and destination points [22].

Due to increased Bluetooth usage onboard vehicles, along with the spread of portable electronics, travelling time and estimation become more accurate and important as more data is collected over time. In addition, it is also possible to measure traffic thickness by capturing the sound of all tire noises, motor noises, engine idling noises, honks, and air turbulence noises [17].

2.1.2 Intelligent transportation system applications

Many applications have been designed for ITS which to progress the throughput, performance, and efficiency of the transportation system as well as to increase roads services to make the trip easy and comfortable. As shown in Figure 2.6. ITS applications can be categorized into four main classes [18]–[20].

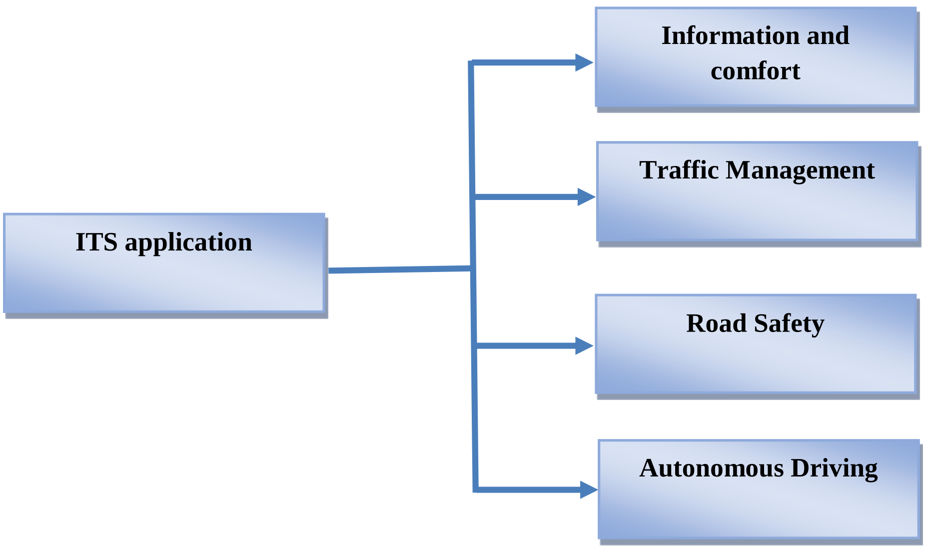


Figure 2.6. ITS applications

2.1.2.1. Information and comfort applications

As part of this application, the drivers will receive vehicle maintenance notes regarding unsafe situations detected or that the car needs maintenance based on the data collected from the sensors in-vehicles. Additionally, this application contains features for providing global Internet access to travelers of vehicles, such as comfort trip services, weather, and entertainment.

2.1.2.2. Traffic management applications

Via these applications, drivers can manage traffic flows and provide various cooperative navigation services in real time and global traffic map. Which can be developed and maintained using ITS messages collected and analyzed.

Roadside units or road sensors collect traffic data, which is transferred wirelessly to trusted remote data centers and these data include vehicle location and driver behavior information. There are many types of traffic management applications, including speed limit notifications, green light optimal speed advisories, electronic toll collection and vehicle highway management.

2.1.2.3. Road safety applications

In the area of road safety, wireless V2X (vehicle-to-everything) communications are used between ITS entities (e.g., vehicles, road infrastructure, etc.). To reduce traffic accidents and protect drivers and pedestrians from road hazards, ITS units periodically broadcast safety messages. (e.g., accidents, detected road hazards).

2.1.2.4. Autonomous driving applications

In 2030, this application is expected to be a technological leap forward for smart transportation.

Based on five levels of automation, this new application will control the vehicle's senses and drive functions that show in Figure 2.7.

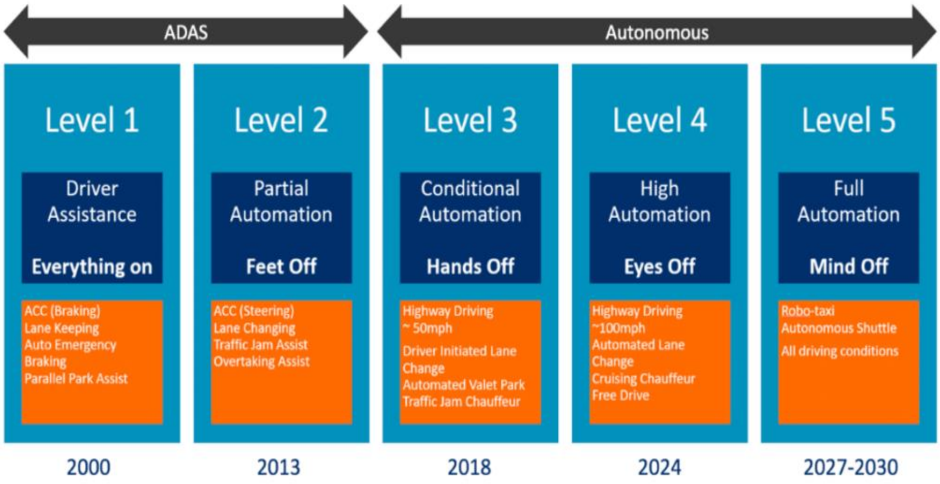


Figure 2.7. Automatous driving level [21]

2.2. Traffic Congestion

With the continuous progress of urbanization in Turkey, vehicle ownership is still rising with increase population for example the population in Sakarya city in Turkey (which is study case in this thesis) increased a 2.6% yearly [22]. As that results, the number urban traffic accidents are increased occur frequently as shown in Figure 2.8. [23], that causing traffic congestion, delay of arrival time, rising fuel consumption and air pollution.

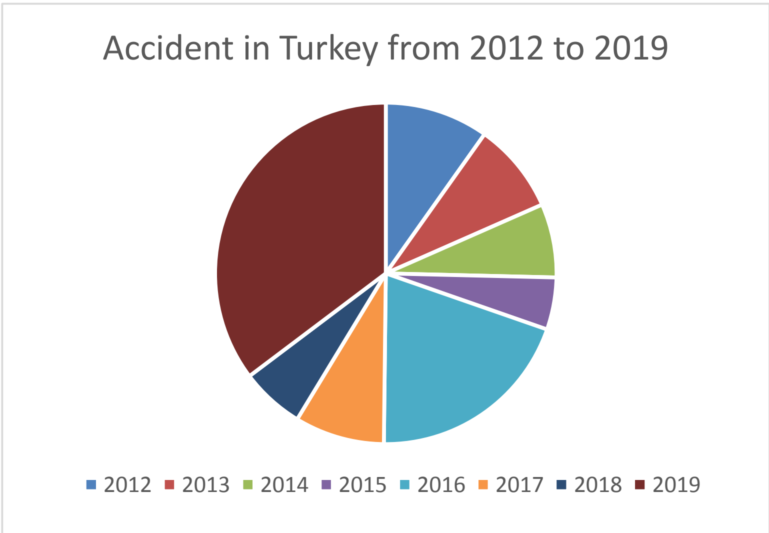


Figure 2.8. Accident Rates in Turkey [23]

World Health Organization (WHO) statistics indicate that about 10 000 lives are lost each year due to road traffic accidents in Turkey, where the road traffic fatality rate is 13 per 100 thousand people. A lack of immediate action on the part of the government will result in road traffic related injuries becoming the fifth leading cause of death in the year 2030 [24].

Traffic congestion may be a major problem faced by many cities that happens when the traffic demand approaches or exceeds the accessible capacity of the road network. Fundamentally, it can characterize a road network condition during that occurs each day during a particular time [25].The UK department of transport announces traffic congestion as a difficult concept to define due to its multi-dimension (Department for Transport, 2018). Various authors define traffic congestion totally different ways from their own points of view [26]. In the context of travel time, a traffic jam occurs when a large number of vehicles block the normal flow of traffic, causing travel time to increase [4]. By adding the inconvenience of interrupted traffic flow to the road user's cost, we can define congestion [27]. As well as contributing to the social depletion, traffic congestion causes air pollution, traffic accidents, noise pollution, in addition to other problems that effect to the environment. Figure 2.9. shows the traffic jam in Sakarya city in Turkey.

Figure 2.9. Traffic jam in Sakarya

2.2.1. Causes of traffic congestion

It has classified traffic congestion into two main groups based on the event that causes the congestion [28]:

2.2.1.1. Recurrent congestion

This refers to a specific area where traffic congestion frequently occurs every day at the same time. It is caused by several reasons such as bad timing for a traffic light, rush hour, bottlenecks, frequent ramps, and special events like national occasions [25].

2.2.1.2. Non-Recurrent congestion

This type can happen anywhere depending on the capacity of traffic and the road condition that changes with time. It has much or little impact on the transportation system depending on road conditions. It occurs as a result of various factors such as road accidents, special events like football matches, bad weather conditions like heavy rain, snowy or icy roadways, road works, and driver behaviour [3], [25], [27]. Federal Highway Administration (DOT-FHWA) statistics indicate that nonrecurring congestion accounts for more than half of all traffic congestion, while recurring congestion accounts for 40% [29].

2.2.2. Traffic congestion measurement metrics

Among the most fundamental factors used to evaluate the performance of a transportation system, it is congestion, which can be measured in many ways, including travel time, speed, level of services, delay time, throughput, and congestion level [27], [30].

2.2.2.1. Travel time (TT)

TT refers to the time needed by a vehicle to finish the trip from starting point to destination point. It is considered an important factor for Intelligent Transportation System (ITS), where it provides facilities to help road users in selecting the optimal route to arrive their destination in the shortest time. Travel time information can be exploited to estimate the future traffic flow as well as it being used in providing alternative routes for drivers to avoid getting stuck in a gridlock. Most existing Intelligent Transportation Systems (ITS) rely on travel time data to detect and control traffic congestion.

2.2.2.2. Speed

It is the average of speed at which a vehicle movement in a limited time. It plays an vital role in traffic congestion measurements. Speed information can be collected from vehicles via various methods. Global Positioning System (GPS) is considered the most frequently used method that provides a vehicles speed, based on the number of satellites orbiting around the earth.

2.2.2.3. Level-of-service (LOS)

LOS is a measure of the amount of congestion suspensions on a particular road. have classified the road traffic flow status into six groups based on velocity and density factors [27], which start from A (free congestion) to F (exceedingly congested) as demonstrated in Table 2.1.

Table 2.1. Level of service (LOS) for road traffic

	Density (Vehicle/ml/lane)	Velocity	Description
A	0 – 12	>60	Free flow
B	12 – 20	>57	Stable flow
C	20 – 30	>54	Slight delays
D	30 – 42	>46	Acceptable delays
E	42 – 67	>30	Approaching unstable
F	67 – 100	<30	Forced flow

2.2.2.4. Delay

It is the rate of wasted time for vehicles during traffic jam occurs of trip. Deducting the arrival time from the logical time needed for the trip will result in the distance travelled.

2.2.2.5. Throughput

This is the number of elements that are passing through a system in a given time. In other words, it is the productivity of a system. In transportation, the throughput can be defined as the number of moving vehicles on a road during a time unit (e.g., one hour).

2.2.2.6. Level-of-congestion (LOC)

Is a pointer that describes the condition of a given street on the road network, which can be represented as a continuous value between 0 and 1. This study has used LOC metric to predict the traffic congestion status for road networks.

2.2.3. Traffic congestion methods

2.2.3.1. Traffic congestion detection methods

As mentioned above an intelligent transportation system aims to manage road traffic efficiently through enhancing public safety, detecting and alleviating congestion, providing accurate information for travellers, and reducing environmental pollution. These aims can be achieved by collecting road information which plays a vital role in ITS, helps in planning, policy, traffic congestion management and research purposes. To do so, there are different developed methods and techniques, which are useful in providing real-time information such as vehicle location, velocity, travel time, traffic state and queue length.

These can be classified into two sets: conventional detector methods and floating car data methods. This study uses the floating car data method to collect road information from vehicles in order to measure and determine traffic congestion locations in road networks via 3G_4G technologies [31].

a. Conventional detection method

This type detects traffic congestion based on sensors that are installed along the roads using various technologies. It provides information gathered from vehicles that pass the detection area, such as average speed, level of congestion, density, and queue length. These sensors can be classified into two groups based on sensor location on the road: intrusive sensors, and non-intrusive sensor.

The former is composed of a data recorder and a detector placed on the roads and represents the applications of advanced technologies in traffic control. Whereas the latter is principally based on remote observation and provides details that fulfil the requirements of many existing freeway and surface street application [32].

b. Floating car data detection method

Floating car data (FCD), also called Probe Vehicle Data (PVD), is another method for gathering real-time road information by tracking the vehicle via mobile phone or GPS which acts as a mobile sensor node that exchanges their data periodically with the traffic center. This information includes position, speed, and direction of travel, it helps in measuring, predicting, monitoring, and managing road traffic systems, as well as locating traffic congestion [30]. The FCD technique is generally classified into three types as shown in Figure 2.10.

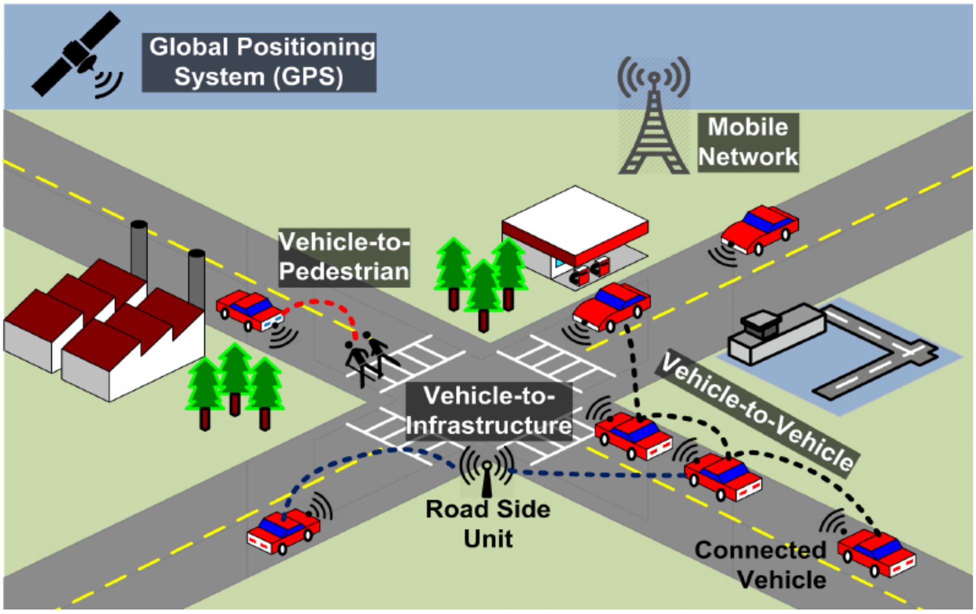


Figure 2.10. Floating car types [17]

- GPS-based FCD

Global Positioning System (GPS) is a satellite-based navigation system that support geolocation, speed, and time for vehicles.

In the past, GPS was used in multiple fields such as civil aviation, marine navigation, communication, and civil and military mapping. Recently, with the widespread use of GPS devices in vehicles, it has proven to be a source of real-time information for intelligent transportation systems (ITS). To improve the performance of traffic management system, this needs to be accurate traffic information. Today a large number of vehicles have embedded GPS in their on-board units and have the ability to provide high-quality traffic information for road traffic management systems, hence, it has become an important positioning technology for providing real-time information (speed, location, travel time) for intelligent transportation systems. The GPS-based FCD method has many advantages such as real-time tracking, travel time provision, effective work under any weather conditions, low cost, and easy maintenance, However, there are some limitations such as low accuracy [33].

- Cellular phone- based FCD

Cellular phone or cellular probe technology is one of the efficient techniques for gathering real-time road information. Currently, most vehicles contain at least one mobile phone that might be utilized as a sensor on the road network. cellular probe technology provides travel time information as well as the location and the speed of the vehicle by tracking the movement of a cell phone on the road network. This technique is useful as it does not require the installation of costly equipment on the road, and due to the increasing number of cell phones in vehicles, the accuracy of traffic road monitoring has greatly improved [34].

- VANET-based FCD

VANET is a promising technology for intelligent transportation system. In VANET, the traffic data is exchanged in two ways, between vehicles, which refers to vehicle to vehicle type (V2V), or between vehicle and infrastructure type (V2I). In general, V2V communication is used to detect traffic congestion and accidents, measure the level of congestion, and provide traffic conditions by using Inter-Vehicle Communication (IVC) as well as providing Internet access and infotainment. While V2I is used to provide real-time information regarding traffic status and weather, and suggest alternative routes to guide drivers to avoid bottlenecks [35], [36].

2.2.3.2. Traffic congestion avoidance methods

Traffic congestion avoidance mechanism is considered one of the main functions for the Vehicle Route Guidance system that aims to help drivers by providing them with useful information related to the traffic status such as accidents, traffic congestion, roadworks, general events (e.g., football match, national occasions), and bad weather conditions (e.g., fog, snow). And also, to reduce travel time by providing the optimal paths from the departure point to destination, the current studies on vehicle route guidance system use different approaches to solve conventional routing problems. It can be classified into two major classes: Centralized approach and Decentralized approach [37], [38].

a. Centralized approach

Centralized vehicle route guidance systems aim to help road-users and make their journeys easy and safe through providing the optimal route for each driver. With this approach, the central server must have full knowledge of all vehicles traffic on the road map. It uses communication equipment that is installed on the roadside to collect and disseminate information between vehicles and the server by using wireless communication technology (e.g. Wi-Fi, WiMAX, or Cellular network). It is assuming that can accept and carry all data from anywhere on the map. This may help to get more accurate navigation decisions, especially over longer distances. Moreover, centralized methods do not demand high computing speeds, full map knowledge, and they can share very little information with another vehicle (e.g., an optimal route to the intended destination). Currently, the majority of institutions, car manufacturing companies, and researchers are interested in 5G technology which is a centralized system. can be implemented in two ways: Reactive and Proactive method [39].

- Reactive method

Reactive system provides recommendations for the optimal path to the drivers only after traffic congestion occurred. Currently, there are many of commercialised navigation systems such as Google map, TomTom, and Garmin that are using to guide vehicles on the road networks. These systems depend on the centralised servers to collect and disseminate traffic information. It uses various sources such as travellers' mobile phones, roadside units, vehicles-equipped GPS, and social media [25]. Although these systems, have the ability to predict traffic condition and calculate the shortest path by using Internet-based solutions. However, it is considered as a reactive solution, that means it capable of re-routing vehicles only after congestion happens. Another problem, when traffic congestion occurs, they provide the same route for all vehicles that are moving on the same route, which means they will switch the congestion from one road to another instead of alleviating it [28].

- Proactive method

During recent years, there has been a rising number of research studies on proactive method to improve road traffic performance and calculate the alternative route using various factors such as route density, the probability of journey time, velocity, and route disruption possibility [28].

Proactive method depends on three main functions; the first one is to collect real-time traffic information (speed, position, and direction) via vehicles. Secondly, use this information to predict the level of congestion for each road. Finally, apply a re-routing mechanism for vehicles who are approaching the congestion area. It aims to decrease in travel time, CO_2 emission, and fuel consumption which indicates the effectiveness of this system in controlling traffic congestion. The main drawback with Proactive method, it does not work effectively with a huge traffic road network as it requires more time for providing alternative routes [40].

This study has used the traffic flow and road-length variables to calculate the congestion level for each road, and congestion threshold = 0.6. Choosing the accurate congestion threshold value is crucial as it has an impact on the performance of the service.

b. Decentralised approach

In the decentralised approach the path guidance decision is processed in-vehicles based on some road factors such as link travel time prediction. That means the traffic management centre does not have any role in route guidance selection but could send information regarding the current traffic conditions to the drivers. Decentralised route guidance systems are cost-effective. However, these systems could not be considered the optimal because each driver makes their own decision and that could affect the other drivers' journeys and road traffic condition. This thesis focuses on the centralised approach.

Table 2.2. The literature review for traffic congestion

Reference	Proposes
Schrank et al [41]	The roadway congestion index (RCI) was developed to assess the severity of congestion on a regional scale. In this study, the daily vehicle miles per lane-mile (DVMT), weighted by the type of road (freeway or arterial), are compared with total mileage expected in the area under congested conditions. A value of greater than 1.0 indicates areawide congestion, weighted by the type of road (freeway or arterial). As a result, travellers find it difficult to relate this index to their experiences, and forecasts of future conditions are difficult as well.
Zhibin Li et.al [42]	They proposed the QL-based VSL control strategy that aimed at reducing system travel time at freeway recurrent bottlenecks via modified CTM was used to model the traffic flow under the influence of VSL control. The simulation results suggested that the proposed strategy effectively reduced traffic congestion at freeway recurrent bottlenecks. The total travel time was reduced by 49.34% and 21.84% in the stable and the fluctuating demand scenarios.
Shubhangi M. Deshmukh et al [38]	To suggest the alternative route for avoiding traffic congestion using smart devices embedded in cars and a road authority can cover their entire road network without investing in infrastructure with the system. A system like that helps driving route planners and navigation systems to avoid traffic congestion. It also identifies the shortest way to travel to save time and fuel. It is an efficient, smart, reliable and low-maintenance system.
Schafer et al [43]	Utilize GPS information to detect blocked or congested road segments, as well as consider a road congested if the average speed of vehicles is below 10 km/h.
Fernando et al. [44]	This architecture proposes using external data sources to detect traffic jams. For distributed traffic information systems, this architecture uses complex-event processing technology to detect congestion on the motorway. While the authors consider digital maps for implementation, it may be better to consider virtual segmentation.
Yuwei et al. [45]	This paper proposes two methods of estimating the travel time of buses using T-window average and N-window average as well as a congestion indicator. This approach does not support vehicles which move in lane reserves.

Table 2.2. (Continued)

Bauza et al. [46]	This paper introduces a collaborative traffic congestion detection technique based on vehicle-to-vehicle communication and fuzzy logic without using sensors to detect traffic congestion.
Francisco et al. [44]	The proposed system detects and distributes traffic congestion collaboratively. A GPS device is used to determine the current location, speed, and direction of vehicles at fixed intervals. The technology also utilizes location privacy and aggregation to reduce bandwidth usage.
Schunemann et al [47]	V2x proposes a vehicle-to-X approach so drivers can share traffic information and optimize their routes based on that information.
Leontiadis et al [48]	This paper proposes a mechanism for crowd-sourcing traffic data and rerouting vehicles accordingly.

2.3. Shortest Path Algorithm

Using the shortest path algorithm is beneficial for managing traffic patterns and guiding drivers along the shortest route to a final destination, as well as decreasing the overall costs of setting up traffic networks [54]. The shortest path is typically represented by using graph theory. In mathematics, a graph is a set of vertices and edges and the shape of a graph can be categorized as directed or undirected depending on whether or not it can be moved along its edges by both sides or by only one side. Also, edges' lengths are commonly called "weights" and the weights are used in order to calculate the shortest path between two points [49]–[51].

There are several kind of shortest path algorithms as follows [51]:

1. Dijkstra Algorithm
2. Floyd-Warshall Algorithm
3. Bellman-Ford Algorithm
4. Genetic Algorithm

One of the most famous algorithms for shorter paths is the Dijkstra algorithm that wrote the source code in 1956 and published it in 1959 by Edsger W. Dijkstra [58].

The Dijkstra algorithm can address all the problems of single-source, directed-weighted, and non-negative graphs [57]. Although that, in some cases, such as when traffic is congested, the shortest path may not be the best. So that in this study, Dijkstra algorithm used to find the optimal route by adding traffic congestion level [59].

As a traditional algorithm, Dijkstra's works as follows: [52], [53].

1. Set distances for the source node to zero and for the rest of the nodes to infinity.
2. Make the source node the current node.
3. Calculate the distance to all otherwise unvisited neighbour's nodes by adding the current node's distance to the weight of the edge between unvisited neighbour and current node.
4. Comparing the measured distance with the current distance from the neighbouring node, set it as the new distance.
5. Next, consider all the unvisited neighbours of the current node, then mark the current node as visited.
6. Once the destination node has been marked as visited, the algorithm has ended.
7. Else, select the least-distanced unvisited node, set it as the current node, and repeat the process starting with step 4.

To clarify, the Figure 2.11. has displayed example how Dijkstra's algorithm finds the shortest path. In this example, the source node is (A), and the goal node is (C) [54].

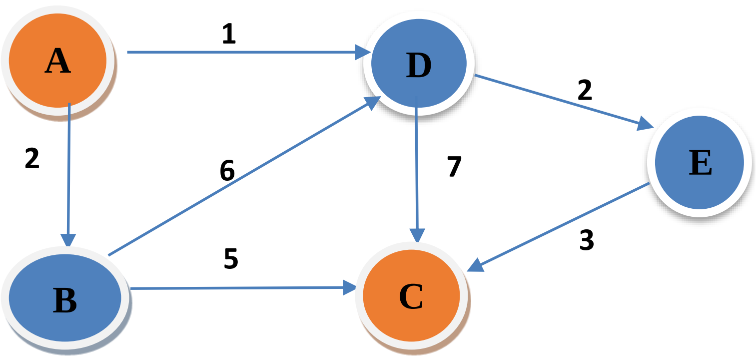


Figure 2.11. Dijkstra example [54]

Table 2.3. Dijkstra example

Vertex	Shortest distance form A	Previous vertex
A	0	
A	1	D
D	2	E
E	3	C

From Table 2.3., the Shortest Path is **A** ⟶ **D** ⟶ **E** ⟶ **C**

Dijkstra’s Algorithm has a very wide range of applications such as [50], [53], [55]

1. Digital Mapping Services in Google Maps
2. Social Networking Applications
3. Telephone Network
4. IP routing to find Open shortest Path First
5. Flighting Agenda
6. Designate file server
7. Robotic path

Numerous efforts are being made to improve Dijkstra's algorithm and its practical application. G. Qing et al had dealt with the problem of finding the best path among selecting points. The authors propose deciding the ideal way starting with a set of ways for the same path toward acknowledging the time needed by the AGV to perform every move with achieving those ends [56]. Ismehene Chahbi et. al suggested an efficient route guidance algorithm for VANETS that employments the MVDR beamforming system which can improve scope what's more lessen radio impedance for remote networks. The recommended algorithm plans should streamline scope that more connectivity from claiming drivers with government-funded hotspots same time guaranteeing a sensible downright traversed separation. Execution outcomes indicate that algorithm outperforms the course direction framework utilizing omnidirectional antennas and the briefest way algorithm [57]. In a study by Ali Alyasin et. al. has presented a new system to track an optimal path road network, based on Dijkstra's algorithm, using a microcontroller with the Raspberry Pi 3 along with Python to determine the optimal path for the smart mobile robot to reach the target. Dijkstra's algorithm was applied for gradient training in order to get the correct path. It is clearly apparent that this method is highly accurate when selecting the shortest path for robot movement. The mobile robot will follow the shortest path and reach the target with the least weight and cost possible [52]. Omoniyi Ajoke Gbadamosi et.al presented a study to recognize alternative routes in situations based on modified Dijkstra algorithm where the costs associated with generating shortest paths are so enormous that the gains from application of the classic algorithm are negligible [58]. Another study on enhanced Dijkstra's algorithm was conducted by Li Wenzheng et.al. By analyzing this property, the authors claimed that the Dijkstra's algorithm only needs to compute one distance for each position. This can save considerable time for the Dijkstra's algorithm and means the Dijkstra's algorithm can be speed up by several orders of magnitude or more through IDA [53]. A similar method is presented by Mingjun et.al. The improved algorithm incorporates road length and traffic factors in order to meet the needs of drivers on the optimal path. The results showed a successful combination by using road length and traffic factors to determine the optimal path for drivers [59]. To find an optimal route and solve the obstacles in finding parking spaces, Yujun et al. suggest an excellent method based on Dijkstra algorithm to search for an optimal parking lot [60].

CHAPTER 3. RESEARCH TOOLS AND METHODOLOGY

Urban cities face a huge problem of traffic jams currently with the number of vehicles increases without good balance between road capacity and population, traffic jams have delayed society's and the economy's progress. Not only that, accidents caused by traffic jams can result in the loss of life. Due to this, the Dijkstra algorithm has been proposed to minimize traffic congestion in Sakarya city, calculate traffic congestion and suggest alternative routes to avoid traffic congestion [1].

This chapter discusses the research tools and methodology to enhanced Dijkstra algorithm via adding the level-of-congestion metric to the shortest path to detect and avoid traffic congestion on the road network.

3.1. Research Tools

This study has depended on tools to provide a realistic environment to simulate vehicles on the road network and create different scenarios of traffic congestion, these tools shown in Figure 3.1.

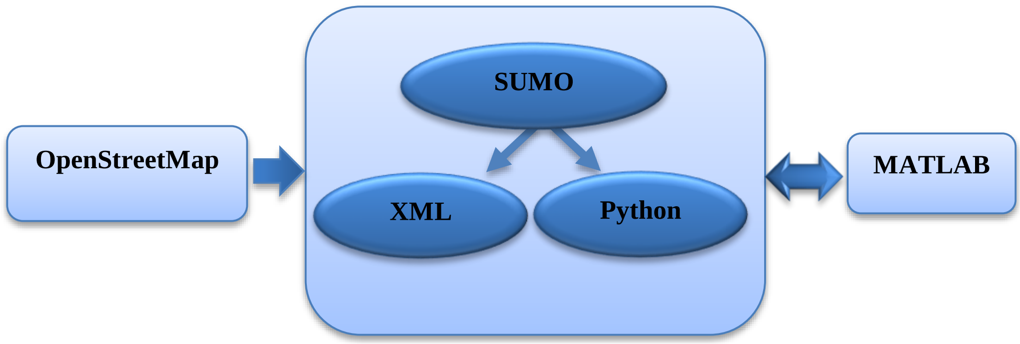


Figure 3.1. Research tools

3.1.1. OpenStreetMap

In 2004, OpenStreetMap (OSM) was issuance in the UK, it is an open data source which any one can access the map data [64]. OSM editor is a one of the most easy and fast programs to get the map data, therefore used OpenStreetMap (OSM) in this study. The two main factors driving OSM growth are the lack of comprehensive map data in most parts of the world and affordability of portable navigation devices.

Approximately two million OSM editors are regularly updating regional maps around the world in every minute so the map exported from (OSM) is far more detailed than any other commercial map and even more comprehensive than Google Maps [64]. Figure 3.2. shows part of Sakarya city map that used in this study.

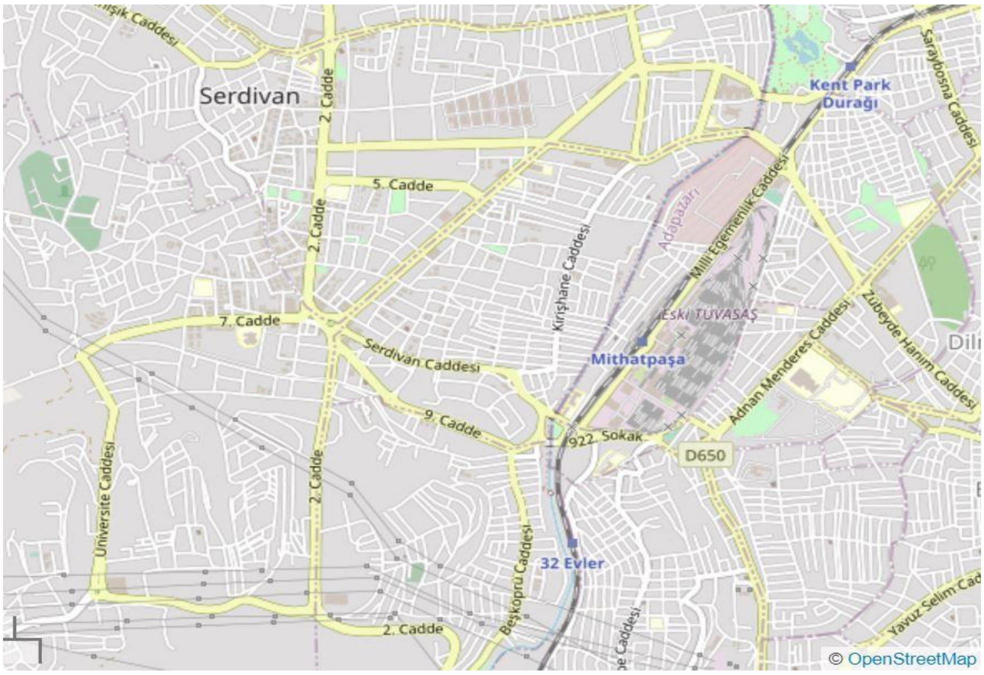


Figure 3.2. Sakarya city (part of map)

3.1.2. Simulation of Urban Mobility (SUMO)

In 2001, the German Aerospace Center (DLR) developed SUMO, an open-source, microscopic traffic simulation program that allows users to create simulations of vehicle movement taking into account road network topology by using digital maps and realistic traffic models [65]. In particular, SUMO network contains details about the edge, shape, and speed limit of each path as well as the logic behind traffic lights provided by crossroads, the junction rules, and the relationship between lanes at the intersection [66]. In this study, SUMO has been selected as the road simulation software because it has real-time capabilities for scenario generation, operation and can be used with other software, such as OSM (OpenStreetMap Editor), which was imported to SUMO to generate simulation maps. A SUMO visualization interface can show all the vehicle movements during the simulation process as well as classify and process the data [67]. SUMO supports a Graphic User Interface (GUI) that provides functions to control and monitor the simulation. This study used SUMO version 1.8.0 and command line as shown in Figure 3.3.

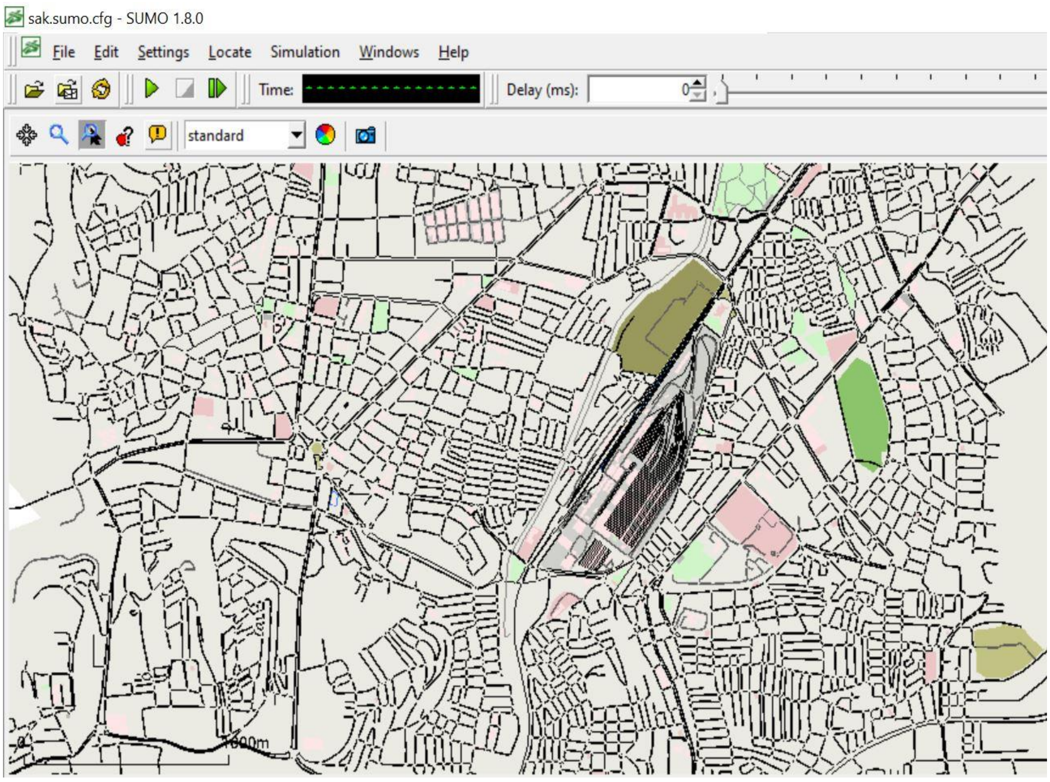


Figure 3.3. Sakarya city (via SUMO Simulator)

As well as a traffic simulation, SUMO has a number of applications that enable the generation and execution of a number of traffic scenarios [22]. Table 3.1. lists SUMO application.

Table 3.1. SUMO application

Application Name	Description
SUMO	Microscopic simulation without visualization; command line application
SUMO – GUI	Microscopic simulation with GUI
NETCONVERT	Network importer and generator; read the network path from a different format and convert it into SUMO format
NETEDIT	Graphical network editor
NETGENERATE	Generate abstract network for SUMO simulation
DUAROUTER	Calculates the fastest route over a network, imports various types of demand descriptions. Do DUA
POLYCONVERT	Importing interesting points and polygons from various formats and translating them into descriptions that can be visualized by SUMO-GUI

3.1.2.1. SUMO mechanism

In order to generate traffic simulation of vehicles, SUMO mechanism has five major parts as follows [22]:

a. Input: SUMO allows importing road networks from different third-party formats like OpenStreetMap databases, PTV VISUM (a macroscopic traffic simulation package), OpenDRIVE networks, MATsim networks, ArcView-data base files, and etc. This study used OpenStreetMap (OSM) file.

b. Network building: Started as soon as data are read (in netconvert) or after an internal description is generated (in netgenerate). This study used netconvert.

c. Demand modeling: After building a network, traffic demand modeling describes the vehicle, generate trip, the edge, the departure time, and give a route for vehicle that contains not only the first and the last edge, but all edges the vehicle will pass. There are several ways to generate routes for SUMO.

The choice depends on your available input data:

a. Using trip definitions: Each trip has at least the source node, destination node and the arrival time. This is important to build demand by hand or by writing scripts to import custom data and use duarouter to convert trips into routes.
b. Using flow definitions: This method is the same approach as using trip definitions, but carry with vehicles having only the same departure and arrival edge.
c. Using Randomization: This is a fast way to get traffic demand modeling. This type of demand modeling has used on this study.
d. Using OD-matrices: It stands for Origin-Destination-Matrices, which is useful when data available from traffic authorities then converted to trips using od2trips.
e. Using detector data (observation points): Induction loops and similar devices are commonly used by authorities to measure traffic and using dfrouter to generate traffic demand.
f. Simulation: The simulation starts at the time given in option begin, which defaults to 0. All vehicles with a departure time lower than the begin time are discarded then the simulation performs step one-by-one and the simulation ends when the final time using option end or no value for option end has been given and all vehicles have been simulated.
g. Output: SUMO allows generating a large number of output files like tripinfo file and output route file. That triggered using command line options.

3.1.3. Python programming language

Python is an open-source programming language, clear, brief, and can be utilized in some operating systems. Software engineers are becoming increasingly interested in Python and some commercial companies use python as well [26]. This study uses Python Version 2.7.

3.1.4. XML programming language

The extensible markup language (XML) is a simple, highly flexible document format derived from SGML (ISO 8879). The original purpose of XML was to support large-scale electronic publishing, but it has since come to be used in a variety of data exchanges online and off [61].

3.1.5. MATLAB

MATrix LABoratory (MATLAB) is the most popular program in the scientific community. It is utilized in most scientific and engineering subjects. MATLAB is a high-performance programming language used to handling technical calculations, computes and demonstration in an easy-to-program environment and does not need much professionalism. This language enables the user to solve several technical problems arithmetically, especially expressed in matrices, which need big effort of other programming languages such as C and FORTRN [69]. MATLAB have many features but the main one which is work with multiple types of programming approaches, such as Functional, Object-Oriented, and array as shown in Figure 3.4.

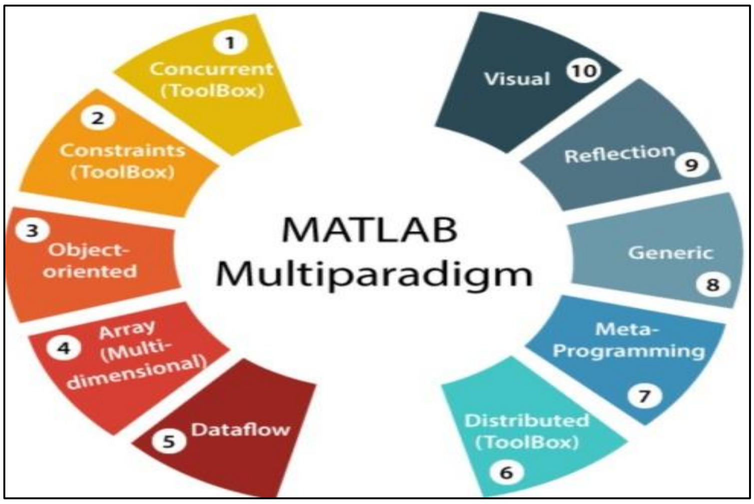


Figure 3.4. MATLAB multi paradigm [70]

3.2. Research Methodology

In this study, a traffic congestion level estimation of part in Sakarya city to get optimal path of vehicle on the road via used microscopic model simulation. This study focused on reducing traffic congestion best on SUMO simulator and MATLAB. From the Figure 3.5. below, the proposed study is distinguished of two frameworks, first framework includes OpenStreetMap and SUMO. OpenStreetMap is building and maintaining real database for the part of Sakarya map and export this data to SUMO, then SUMO used this real data to generate traffic simulation of vehicles that are useful for determining the position and state of traffic jam in limited time. The second framework have Dijkstra algorithm and MATLAB that used to process the data obtained from the SUMO output files and used this real data to enhancement Dijkstra algorithm via calculate and adding congestion level parameter to classic Dijkstra weight to avoid traffic jam by reroute vehicle. After that estimate the effect of improved Dijkstra algorithm in vehicle via sumo simulator. The main functionality of improved Dijkstra algorithm is to alleviate traffic congestion by reducing travel time and also minimize fuel consumption and CO_2 emission.

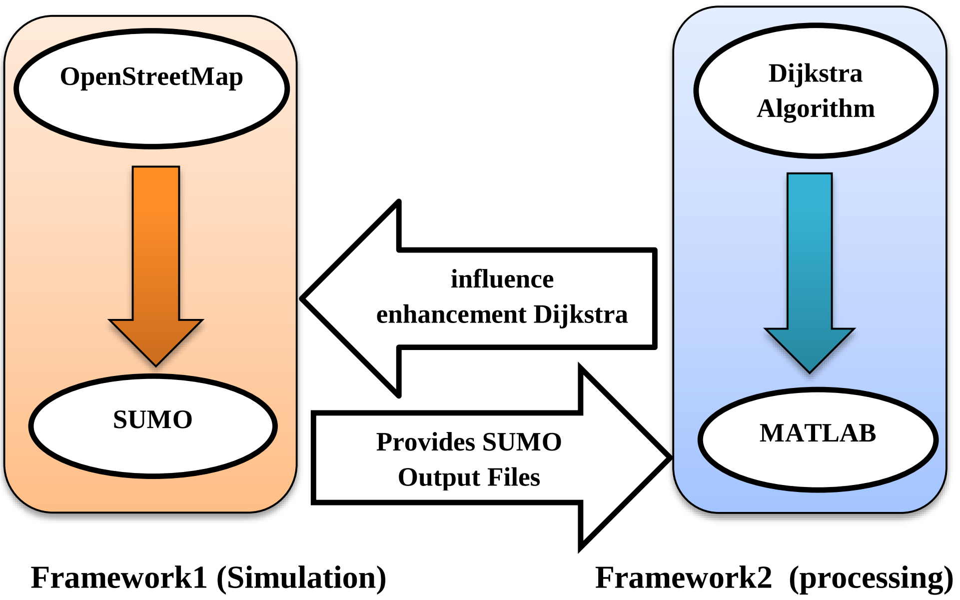


Figure 3.5. Main frameworks of methodology

3.3. The Improved Dijkstra Algorithm

According to the basic two frameworks in Figure 3.5., the traffic congestion level parameter is used to optimize the distribution path of vehicles in road network. The improved Dijkstra algorithm steps can be described as shown in Figure 3.6., as following steps:

- Step 1: Import SUMO output files, then read files as input on MATLAB to generate chain of matrices;

  Set congestion level parameters = 0.
  Set THreshold Value (THV) =0.6.

- Step 2: select vehicle with route from vehicle matrix.

- Step 3: calculate traffic congestion level parameter

  Traffic congestion level = Traffic flow/ Road Capacity

Then, check traffic congestion level against selected vehicle route.

- Step 4: for the initial point, if traffic congestion level for selected vehicle route bigger than (THV), then apply improved Dijkstra model via rerouting vehicle with consideration shortest path when select alternative route; otherwise, apply classic Dijkstra.

- Step 5: move vehicle in the road network depending on the time to arrive goal point.

- Step 6: update traffic congestion level and go to step 2 to select next vehicle, iterations. Until to finds optimal path for all vehicles, output, optimal path (lesser traffic congestion and shortest path) for all vehicles.

- Step 7: End.

Figure 3.6. describes the flowchart for improved Dijkstra Algorithm.

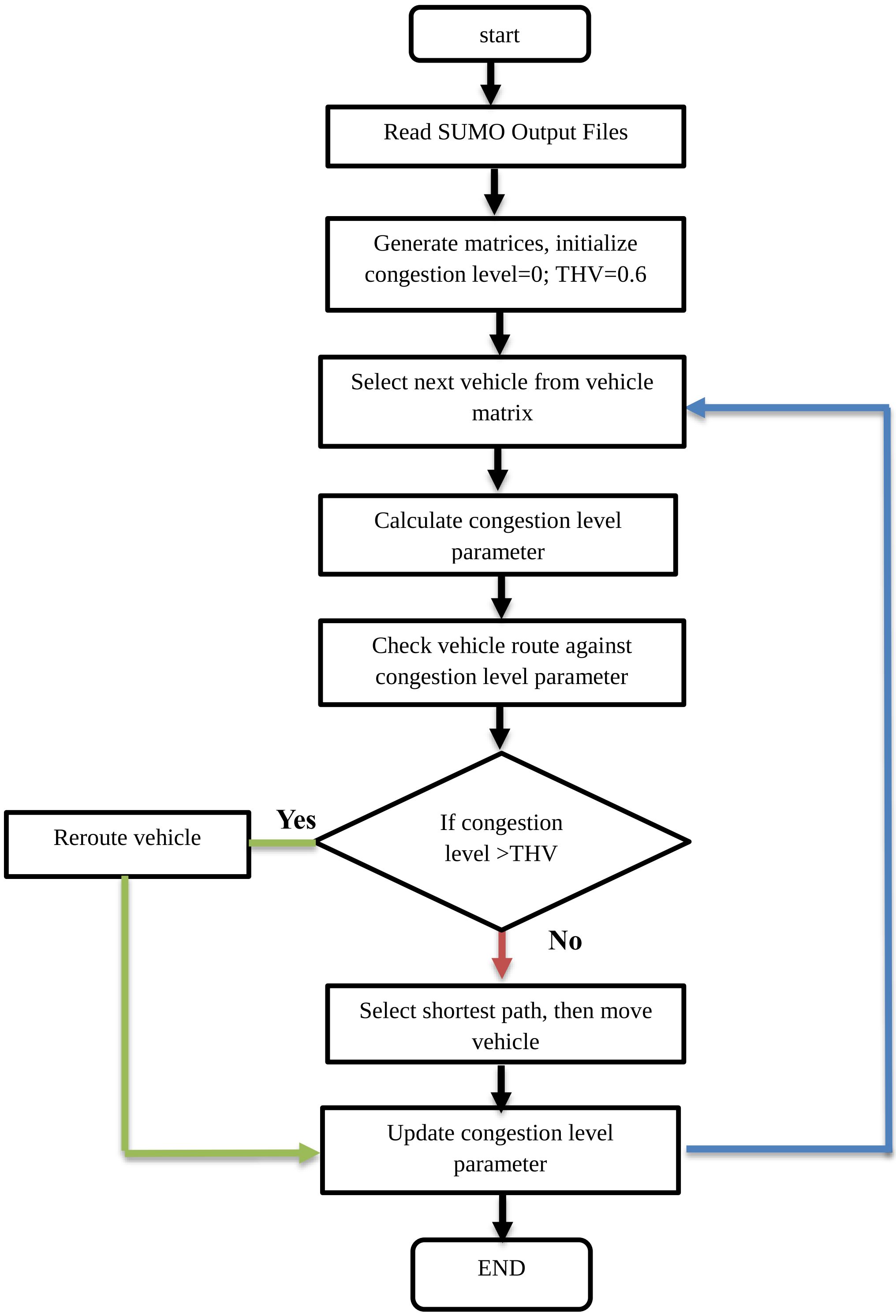


Figure 3.6. Improved Dijkstra Algorithm.

Based on Figure 3.6., the improvement Dijkstra algorithm stage will be explained as following:

- Input

Initialization map and input matrices

As mentioned in Section 3.1.1 and 3.1.2, OpenStreetMap convert part of Sakarya city map to digital map and save as OSM file, then SUMO has used OSM file to generate vehicles and make traffic simulation.

- Generate Matrices

Depending on SUMO output files, MATLAB generate a number of cascading matrices as following:

- Generate the road connected matrix

This matrix represents how roads are connected on the network, where it contains value 0 or 1. The value (0) refers to no connection between the selected roads while the value 1 means there is a physical connection between the selected road intersections (nodes). The horizontal and vertical column represents the road intersection (Node ID). This matrix is generated based on SUMO_edge file as shown in Equation 3.1.

$$\text{Road}_{\text{Connected}}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \quad (3.1)$$

- Generate the Road_ID matrix

This matrix refers to the road ID (road name) on the network. The Road_ID represents the route of the vehicle on the road network. It is generated by reading the content of the SUMO node and edge files of the map as shown in Equation 3.2.

$$\mathrm{Road_{ID}}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 13822 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1399 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2155 & 0 & \dots & 0 & 0 & 0 \\ 1400 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 5518 & 0 & 0 & \dots & 0 & 0 & 0 \\ 4913 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 5436 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \quad (3.2)$$

- Generate the Road_Length matrix

This matrix represents the real length of each road in meters. It is generated by reading the content of the SUMO edge file of the map as shown in Equation 3.3.

$$\mathrm{Road_{lenght}}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 130 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 400 & 0 & \dots & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1000 & 0 & 0 & \dots & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \quad (3.3)$$

- Generate the vehicles _ matrix

This matrix represents vehicles moving on the road network. It is generated by reading the content of the edge file, rout file and trip-file of the map. Figure 3.7. shows the part of vehicles matrix.

8	-1.4E+10	-1.4E+10	-1.4E+10	-1.4E+10	480	0	0	0	0	0	0
11	4.88E+08	-4.9E+08	4.88E+08	8.51E+10	8.51E+10	8.51E+10	-1.4E+08	4.88E+10	4.91E+08	-4.9E+08	1200
14	1.95E+10	1.89E+10	1.95E+10	1.95E+10	1.95E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+08
17	3.26E+10	-4.9E+10	3.26E+10	3.26E+10	3.26E+10	3.26E+10	2.12E+08	3.26E+10	3.26E+10	3.26E+10	3.26E+08
25	4.91E+08	4.91E+10	4.91E+08	6.78E+08	6.78E+08	4.91E+08	4.91E+08	4.91E+10	4.91E+10	4.91E+10	4.91E+10
19	1.4E+10	3.68E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+08	6.78E+08	4.91E+10
15	1.4E+10	4.53E+10	1.4E+10	1.95E+10	1.95E+10	1.95E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+10
12	4.91E+10	3.26E+10	4.91E+10	4.91E+10	4.91E+10	4.49E+10	4.49E+10	4.49E+10	3.26E+10	3.26E+10	3.26E+10
0	4.88E+10	-4.9E+10	4.88E+10	4.88E+10	4.88E+10	4.88E+10	4.88E+10	4.88E+08	8.51E+10	1.4E+08	2.16E+10
6	1.4E+10	5.79E+10	1.4E+10	3.26E+08	-1.4E+10	-1.4E+10	-1.4E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10
16	1.4E+08	1.95E+10	1.4E+08	-1.4E+10	-1.4E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10	1.4E+10
4	3.26E+10	1.4E+08	3.26E+10	3.26E+10	3.26E+10	6.71E+10	6.71E+10	6.71E+10	6.72E+08	3.26E+10	3.26E+10

Figure 3.7. Part of vehicles matrix

First column: represent the vehicle ID.

Second column: represent the start point of vehicle (departure time).

Third column: represent the end point of vehicle (arrival time).

The rest of columns: represent the vehicle route.

Last column: represent the length of trip.

- Traffic Congestion Measurement

As mentioned in Section 2.4.2, there are several metrics that are using to determine the status of traffic congestion on the road network. This study has used the traffic flow and road capacity parameters to find out the level-of-congestion that represents the traffic congestion density for a certain road.

Traffic flow parameter: is a variable value, which refers to the current number of vehicles for a road. It is generated based on information from the Vehicle matrix and the Road_ID matrix as shown in Equation 3.4.

$$TF_i = \sum_{k=1}^{n} V_{ki} \tag{3.4}$$

Where

V: represents vehicles

n: represents number of vehicles on the road network

k: current number value

Equation 3.5 shows part of the Traffic Flow matrix.

$$\text{Traffic Flow}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 36 & 0 & \dots & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 120 & 0 & 0 & \dots & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \quad (3.5)$$

Road capacity parameter: is a static value, which refers to the maximum number of vehicles that the road can accommodate. It is generated by reading the content SUMO_ edge file of the map and can calculated by applying Equation 3.6.

$$R_{ci} = \frac{R_{li}}{L_v} \quad (3.6)$$

Where

Rci: represents the road capacity of road

Rli: represents the route length of road

Lv: represents the length of vehicle with the safety distance

The safety distance is the minimum gap (in meters) between two consecutive vehicles, which is measured from the front end of a vehicle to the rear end of the other. To make it simple, this study has supposed that all vehicles have the same length (6 meters) and the minimum gap (4 meters).

Equation 3.7 shows part of the Road Capacity matrix.

$$\text{Road capcity}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 26 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80 & 0 & \dots & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 200 & 0 & 0 & \dots & 0 & 0 & 0 \\ 20 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \quad (3.7)$$

Level-of-congestion (LOC): Is a pointer that describes the condition of congestion level a given street on the road network, which can be represented as a continuous value between 0 and 1. This weight is based on the information of the Traffic Flow and Road Capacity matrices and can calculate by applying Equation 3.8.

$$\mathrm{TC_i} = \frac{\mathrm{TF_i}}{\mathrm{RC_i}} \tag{3.8}$$

Equation 3.9 shows part of the traffic congestion matrix.

$$\text{Traffic Congstion}(405 \times 405) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0.6 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \tag{3.9}$$

This study has classified the traffic congestion status into four levels: free of congestion, slight congestion, heavy congestion, and locked congestion as shown in Table 3.2. [27].

Table 3.2. Traffic congestion levels

LOC	Description
0	Free of congestion
> 0 – 0.5	Slight congestion
> 0.5 – 0.9	Heavy congestion
> 0.9 – 1	Locked congestion

Level-of-congestion threshold: Is an important key value in identifying the road condition on the network, where the road is considered as congested the traffic flow is equal to or greater than the threshold value. Choosing the correct value for threshold is substantially important and can affect the performance of the proposed algorithm. If the threshold value was too small, it will result in re-routing vehicles that are unaffected by the congestion, which will cause increased travel time for vehicles as well as increasing the computation of the proposed algorithm. On the other hand, when the threshold value is too large, it will result in detecting the congestion too late and the re-routing for vehicles will be useless.

This study used 0.6 as threshold value and is implemented in two fundamental cases: classic Dijkstra model and the improved Dijkstra model.

a. Classic Dijkstra Mode

Depending on threshold value that is less than or equal of (0.6), The vehicle journey used classic Dijkstra mode moving on the road network from start point to destination point as explained on Figure 2.11.

b. The Improved Dijkstra Mode

Depending on threshold value that is bigger than (0.6), that is implemented the improved Dijkstra model via rerouting vehicle based on a number of generated matrices such as Vehicle, Traffic Flow, Traffic Congestion, and Road length for any cause of a traffic jam like road accident, existence of roadworks, bad weather and etc. Thus, the road is partially or completely closed for a period of time. The proposed of Dijkstra algorithm will be performing a number of functions as follows:

- Generate related vehicles (RV) matrix

This study has assumed that the road is considered as congested if the traffic congestion level is greater than the threshold value (0.6). When congestion occurs generate related vehicles (RV) matrix which represented all vehicles route that pass of congested areas. Figure 3.8. displays related vehicles (RV) matrix.

6745	6.78E+08	1.89E+10	1.89E+08	1.89E+10	1.89E+10	1.89E+10	1.89E+10	5100	0	0
6741	6.37E+10	2.16E+10	5.44E+08	8.08E+10	8.08E+10	8.08E+10	8.08E+10	2.16E+10	2.16E+10	5400
6717	2.16E+10	-6.7E+08	3.26E+10	1.4E+08	1.4E+10	1.4E+10	6.72E+08	-6.7E+08	5370	0
6743	2.16E+10	3.68E+10	4.91E+08	4.91E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10	4.91E+10
6736	2.75E+10	4.91E+08	3.26E+08	6.38E+10	6.38E+10	6.38E+10	4.53E+08	1.4E+10	6.78E+08	4.91E+08
6744	2.75E+10	1.85E+10	-1.4E+08	4.88E+10	4.91E+08	-1.9E+10	-1.9E+10	1.85E+10	1.85E+10	5960
6719	4.49E+10	1.4E+10	1.4E+10	1.4E+10	1.4E+08	1.4E+10	1.4E+10	1.4E+10	1.4E+10	1.95E+10
6783	2.16E+10	2.16E+10	1.89E+10	2.16E+08	2.16E+10	2.16E+10	2.16E+10	4490	0	0
6769	8.08E+10	1.89E+10	4.49E+08	1.4E+10	1.4E+08	2.16E+10	1.89E+08	1.89E+10	1.89E+10	4980
6762	4.91E+10	4.88E+10	1.4E+10	-4.9E+08	32485649	4.88E+10	4730	0	0	0
6808	1.4E+10	6.37E+10	4.91E+10	4.91E+10	6.37E+10	6.37E+10	6.37E+10	6.82E+08	6.37E+10	6.37E+10

Figure 3.8. Part of related vehicles (RV) matrix

Fisrt column: represent the vehicle ID.

Second column: represent he start point of vehicle (source node).

Third column: represent the end point of vehicle (destination node).

Fourth column: represent the current location of vehicles on the road network.

The rest of columns: represent the vehicle route.

Last columns: represent the length of trip.

- Generate road ignored (RI) matrix

After identifying congestion routes and generate related vehicle (RV) matrix, then ignore all roads have length less than (100 m) to avoid traffic congestion which be happen and finish at small time period as shown in Equation 3.10.

$$\mathrm{Road}_{\mathrm{ignored}}(10 \times 2) = \begin{bmatrix} \mathrm{Road}_{\mathrm{ID}} & \mathrm{Roadlength} & \mathrm{Congestion}_{\mathrm{Level}} \\ 9823 & 50 & 0.72 \\ 6472 & 70 & 0.66 \\ 6324 & 60 & 0.71 \\ 3154 & 90 & 0.72 \\ 7136 & 70 & 0.69 \\ 1469 & 80 & 0.63 \\ 9321 & 40 & 0.73 \\ 2794 & 50 & 0.81 \\ 3722 & 90 & 0.68 \\ 1175 & 60 & 1 \end{bmatrix} \quad (3.10)$$

- Rerouting vehicles

When traffic congestion happens on the road network, the improved Dijkstra model will look for the vehicles which are pass on the congestion area, to reroute and prevent them from entering in the congested area via give alternative route for vehicle. The rerouting vehicles function as follows steps:

Step 1: Initialize, set congestion route_id
Step 2: generate (RV) and (APR) matrices.
Step 3: check congestion route_id against (APR) matrix.
Step 3: if find congestion route on (APR) matrix, remove this route from (ARP) matrix, else save route on (APR) matrix.
Step 4: arrange (APR) matrix in ascending order depending on the length of trip.
Step 5: select optimal path and give for vehicle route.
Step 6: End.

The APP matrix shows in Figure 3.9.

189421156	21553715723	21553715724	21553715725	14012632620	14012632621	14011975223	14011975224	14011975225
189421156	21553715723	21553715724	21553715725	14012632620	194897795	19489780121	19489779221	19489779222
189421156	21553715723	21553715724	21553715725	14012632620	14012632621	19489780120	19489780121	19489779221
189421156	21553715723	140126324	85060108822	85060108823	85060108820	140126327	21553715725	14012632620
189421156	21553715723	21553715724	21553715725	14012632620	14012632621	14011975223	14011975224	14011975225
189421156	21553715723	140126324	85060108822	-140120733	-48823690211	488236901	85060108820	140126327
189421156	21553715723	140126324	85060108822	85060108823	85060108820	140126327	21553715725	14012632620
189421156	21553715723	21553715724	21553715725	14012632620	194897795	19489780121	19489779221	491292386
189421156	21553715723	21553715724	21553715725	14012632620	14012632621	14011975223	14011975224	13967979520
189421156	21553715723	21553715724	21553715725	14012632620	194897795	19489780121	19489779221	19489779222
189421156	21553715723	21553715724	21553715725	14012632620	14012632621	14011975223	14011975224	14011975225
189421156	21553715723	140126324	85060108822	85060108823	85060108820	140126327	21553715725	14012632620

Figure 3.9. Part of the (APR) matrix

Figure 3.10. describes the flow chart for rerouting vehicle.

This study recommended to implement several scenarios to choose the optimal value. Concerning the threshold distance value, where it can be assigned based on network size and congestion duration. So, the candidate vehicle which is pass to the congested area will have be rerouted. Figure 3.11. demonstrates a simple scenario of a traffic jam in the road in approximate shape of the Sakarya map and how to identify rerouting and chose the optimal path.

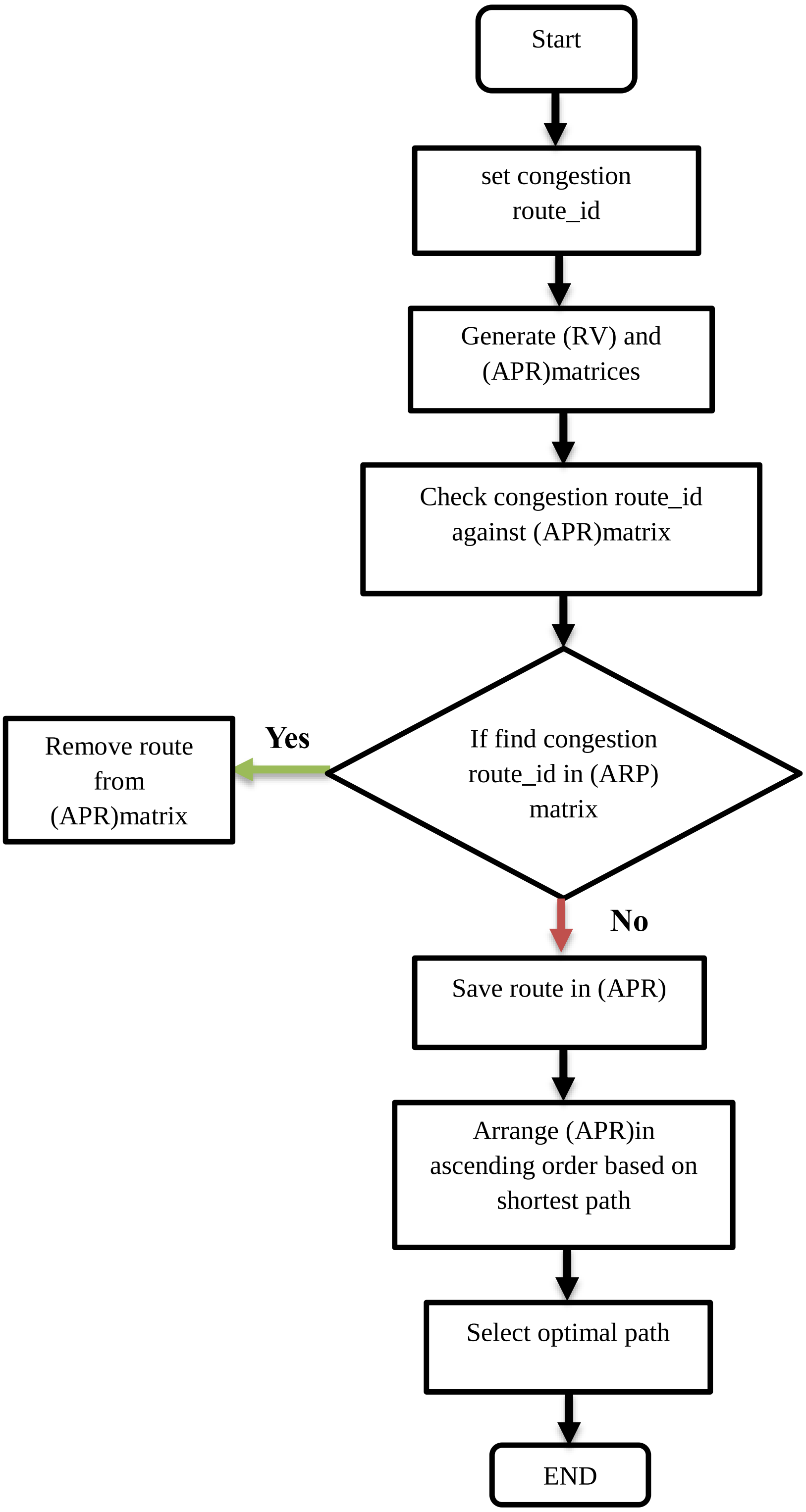


Figure 3.10. The flow chart for rerouting vehicle

Figure 3.11. A simple scenario, which describes how the optimal path has been computed

The Figure 3.11. displays a part of Sakarya road network that consists of 11 intersections (represented from node R1 to node R11) and 15 roads of different length. Assuming that a vehicle is planning to make a journey from node R1 (source point) to node R11 (destination point) through the many routes Table 3.3. explains how calculate optimal path based on Figure 3.11.

Based on a vehicle color on Table 3.3. has three scenarios as follows:

Scenario 1: (green vehicle), the road network commences in node R1, the vehicle can arrive at nodes R2, R4 and R6, these three routes have free congested route and the path between R1 and R6 is shortest (408m) therefore going R1 to R6 then path R6 to R5 is (1200m) but this path has congested route, so ignored this route and the alternative route is R6 to R7 to R5 and this path length is long (1823m), then from node

R5 to node R10 to arrive destination node R1. The trip has free congestion and the trip length is (4477m).

Table 3.3. Optimal path calculation

Scenario	Source node	Next node	All possible path	Shortest path	Trip length	Traffic congested status
Green vehicle	R1	R2, R4, R6	[R1,R2][R1,R4][R1,R6]	[R1, R6]	408m	free
Green vehicle	R6	R7, R5	[R6,R7][R6,R5]	[R6,R5]	1608m	blocked
Green vehicle	R6	R7	[R6,R7,R5]	[R6,R7,R5]	2231m	free
Green vehicle	R5	R10	[R5,R10]	[R5,R10]	3531m	free
Green vehicle	R10	R11	[R10,R11]	[R10,R11]	4477m	free
Red vehicle	R1	R2	[R1,R2]	[R1,R2]	667m	free
Red vehicle	R2	R3	[R2,R3]	[R2,R3]	1406m	free
Red vehicle	R3	R4	[R3,R4]	[R3,R4]	2031m	free
Bule vehicle	R1	R4	[R1,R4]	[R1,R4]	943m	free
Bule vehicle	R4	R9	[R4,R9]	[R4,49]	1943m	free
Bule vehicle	R9	R10	[R9,R10]	[R9,R10]	2772m	free
Bule vehicle	R10	R11	[R10,R11]	[R10,R11]	3718m	free

Scenario 2: (Red vehicle), the red vehicle start move from node R1 to node R2 because the path length between R1 and R2 less length between R1 to R4 then node R2 to node R3 and node R3 to node R4, the trip length for arrive node R4 is (2031m) in scenario 2, but in scenario 3 bule vehicle arrive to node R4 in trip length is (943m), So going to scenario 3.

Scenario 3: (Bule vehicle), the bule vehicle can start from node R1 to node R4, then node R4 to R9 to R10 to arrive destination point R11 without traffic jam and the trip length is (3718m), so the bule vehicle has optimal path because have shortest path with free congestion to arrive the destination node. The optimal path is R1, R4, R5, R9, R10.

CHAPTER 4. EXPERIMENTS, RESULTS AND DISCUSSION

Improved Dijkstra algorithm is implemented on the part of Sakarya city roadmap. This chapter includes two main topics related to the improved Dijkstra algorithm, which are the roadmap topology and experiments. The roadmap topology section describes the types of roadmap topology, characteristics of case study (part of Sakarya city roadmap) and the options that have been used and added on SUMO simulator. The experiment section presented three experiments scenarios and to make the road network environment more realistic display the experiment determinants that utilized in this study then evaluate the experiments results. Finally, the comparison between improved Dijkstra and classic Dijkstra has been evaluated based on the performance metrics; related vehicles on the traffic congestion, average delay time, fuel consumption and CO2 emission.

4.1. Roadmap Topology

The roadmap describes map information such as number of roads, number of intersections, road length, number of lanes, maximum speed allowed and etc. Roadmap network classified based on road intersection shape into three categories [62].

4.1.1. Grid roadmap

This type of roadmap is characterized by right angles at each crossroad which create a grid shape. The infrastructure cost for a normal grid is much more expensive and have delay time for vehicles compared with the other roadmap topologies according to the traffic incidents and other traffic determinants.

4.1.2. Roundabout road map

A modern road intersection that is used in many countries and have several features such as yield control of all entering traffic and the delay time is less than other roadmap topology.

4.1.3. Hybrid roadmap

This kind of roadmap is combination from grid and roundabout roadmap. This study used this type of roadmap to represent Sakarya map.

4.2. Case Study

In order to simulate and evaluate the improved Dijkstra algorithm, this study used hybrid road map topology (represented in part of Sakarya traffic map inTurkey) as case study). Table 4.1. presents a brief information about the simulation experiments. Figure 4.1. shows the case study roadmap via google satellite.

Table 4.1. Case study information

Parameters	Value
Case study	Part of Sakarya city in Turkey
Map size	25 km^2
Number of roads	622
Number of intersections	405
Type of roadmap	Hybrid roadmap
Maximum speed allowed	70km/h
Driving side	Right

Figure 4.1. Sakarya map (case study)

4.3. Simulation Experiments

SUMO is used to simulated and evaluated improved Dijkstra algorithm. The traffic map is extracted from the OpenStreetMap website as OSM file. Then, it filtered out unnecessary information such as building names, road signs, rivers, and background decorations because this information may affect the execution of the simulation. This study has used SUMO version 1.8.0 as a simulator platform to apply the scenarios on the part of Sakarya real roadmaps. The most important characteristic of SUMO is able to simulate a different of vehicle types and multi-lane road. Furthermore, it supports different kinds of map formats such as OpenStreetMap (OSM), Vissim, ArcView, and XML-Description as mentioned in Section 3.1.2.1. Although SUMO provides all these features, there are still some drawbacks in the design and implementation that need to be developing for example the matter of generating the same number of vehicles on the map at a specific period. Figure 4.2. presents the SUMO architecture that is utilized in experiments.

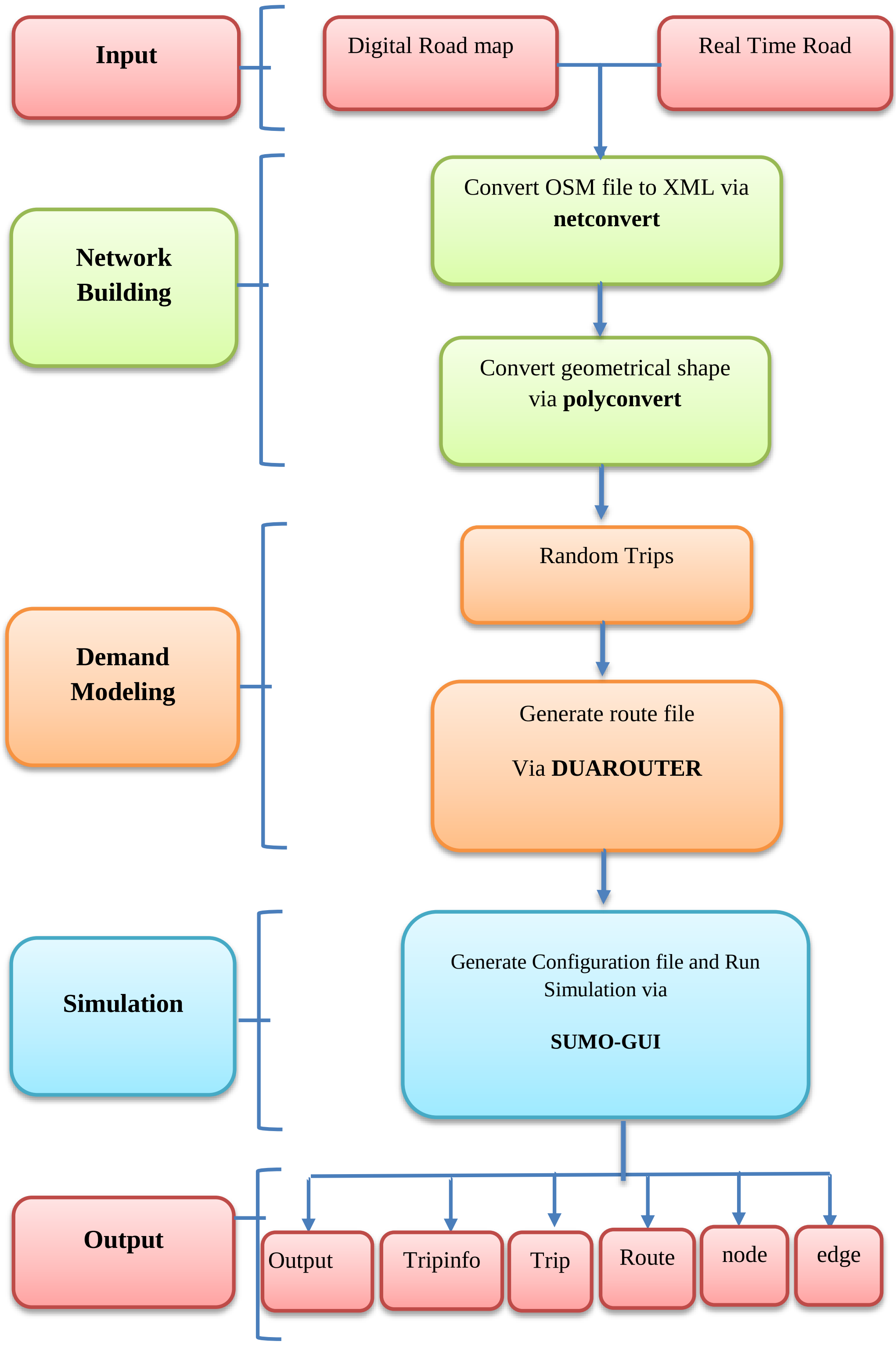


Figure 4.2. The SUMO architecture

Input: the simulation has two inputs

Digital road map: which is imported from the OpenStreetMap (OSM) database. In this study, a part of Sakarya city is used as indicated on Figure 3.2. This map contains information about the roads e.g., road length, the maximum speed allowed, and the number of lanes per road.

Real-time traffic information: is an important data that helps to know traffic jam and incidents. This information includes vehicle location that is gathered from vehicles movement on the roads using floating car data method. In this study, it is assumed the simulation total time to 3 hours.

Building network: building network via run the following command line:

Netconvert: imports digital road networks from different sources such as (OpenStreetMap) and generates road networks that can be utilized by other tools from the package. The "Netconvert" instruction as below.

netconvert --osm my_osm_net.xml

polyconvert: it is imports geometrical shapes (polygons or points of interest) from different format then turned them to a representation that can visualized using sumo-gui.the "Ployconvert" instruction as below.

polyconvert --net-file filename.net.xml --osm-files filename. Osm --type-file typemap.xml -o filename.poly.xml

Demand modeling:

Random trips: it used to generate random routes for a given road network with the options as shown in Table 4.2., by help a python programming that generate random trip script which can also produce a set of routes employing the duarouter as following commend line.

Python tools/randomTrips.py-n<net-file>

Table 4.2. The random trip options

Begin time (b)	0
End time (e)	10800s
Period (p)/interval	120s
Number of period (P)/slot	90
Peak time (m)	45m
Sigma value (s)/normal distribution	26
Number of vehicles (N)	300 vehicle/km^2
Route generation method	Random
Routing algorithm (R)	Dijkstra

DUAROUTER: imports different demand definitions, computer vehicle routes that using shortest path computation via the following commend line:

tools\randomTrips.py" -n filename.net.xml -r filename.rou.xml

Simulation: generates configuration file and run simulation on sumo_gui via the following command line:

sumo-gui -c filename.sumo.cfg

Output: SUMO simulation provides several output files. The list of SUMO output files that utilized in this study as follow:

Edge file: it has data for each edge (road) in the map like road id, start point, end point, speed limited, number of lanes and etc. The number roads in this study (part of Sakarya city map) are (622) edges. Figure 4.3. shows the information for one edge.

```
    <edge id="-139962358#4" from="2216485648" to="1542086064"
priority="10" type="highway.tertiary" numLanes="1"
speed="22.22" shape="10440.97,7745.92 10432.31,7743.95
10411.06,7734.68 10378.80,7717.49 10359.94,7706.14"
disallow="tram rail_urban rail rail_electric rail_fast ship"/>
```

Figure 4.3. The information for one edge

Node file: it has data for each node (intersection)in the map like node id, x value, y value and etc. The number intersection in this study (part of Sakarya city map) are (405) nodes. Figure 4.4. shows the information for one node.

```
    <node id="1040766682" x="9141.42" y="5933.82"
type="priority"/>
```

Figure 4.4. The information for one node

Route file: it has data for each vehicle like vehicle id, depart time and vehicle routing**.** Figure 4.5. shows the information for one vehicle.

```
    <vehicle id="2" depart="2.00">
        <route edges="215282166 140828915#4 140828915#5
140828915#6 140828915#7 140828915#8 -140868490#5 -140868490#
4 -140868490#3 -140868490#2 -140868490#1 -140868490#
0 -140868429#2 -140868429#1 -140868429#0 -140868461 140868433#
0 378603511#0 378603511#1 -491330765#4 -491330765#3 -491330765
#2 -491330765#1 -491330765#0 491292368#6 491292368#7 491292368
#0 491292368#1 491292368#2 491292368#3 491292368#4 491330763#0
215255683#0 215255683#1 215255683#2 215255572#8 215255572#9"/>
    </vehicle>
```

Figure 4.5. The information for one vehicle.

Trip file: it has data for each trip of vehicle like trip id, departure time, start point and end point. Figure 4.6. shows the information for one trip.

```
    <trip id="16" depart="16.00" from="212147747#0"
to="660833696#2"/>
```

Figure 4.6. The information for one trip

Tripinfo file**:** it has data for each trip of vehicle like tripinfo id, arrival time, departure time and etc. Figure 4.7. shows the information for one trip.

```
    <tripinfo id="56" depart="56.00" departLane="140817829#2_
0" departPos="5.10" departSpeed="0.00" departDelay="0.00"
arrival="181.00" arrivalLane="-215255404#14_0"
arrivalPos="16.54" arrivalSpeed="9.62" duration="125.00"
routeLength="1493.93" waitingTime="0.00" waitingCount="0"
stopTime="0.00" timeLoss="36.03" rerouteNo="0"
devices="vehroute_56 tripinfo_56" vType="DEFAULT_VEHTYPE"
speedFactor="1.00" vaporized=""/>
```

Figure 4.7. The information for one tripinfo

Output route file: it has data for each vehicle like vehicle id, arrival time, and departure time and vehicle route. Figure 4.8. shows the information for one trip.

```
    <vehicle id="98" depart="98.00" arrival="146.00">
        <route edges="212147983#4 -212148818#1 -212148818#0
212147975#1 212147975#2 212147975#3 212147975#4 212147975#5
212147975#6 -212147975#6"/>
    </vehicle>
```

Figure 4.8. The information for one vehicle

Some of output files is generate before run SUMO via instruction "sumo-gui -c filename.sumo.cfg" such as node.xml, edge.xml, rou.xml and poly.xml (used for visualizing geometrically shapes such as buildings), and other output files building in after simulation on SUMO_GUI is tripinfo.xml and outputroute.xml.

4.4. Experiment Determinants

The experiment scenario requires two essential things: provides a mobility realistic map suitable with SUMO and generates a route for all vehicles. And these two things have described in detail in the previous subsection. SUMO depends on many mobility information to run a scenario such as a road information, priority, the maximum speed allowed, the route of vehicles, and simulation time. To make all scenarios more realistic, the simulation time for all scenarios was assigned to three hours and the experiment run-time was divided into 90 intervals and each 2 minutes updated the vehicle position as shown in Figure 4.9. On the other hand, the process of generating new vehicles on the map is random and process of vehicles distribution over each interval (90) is normal distribution by applying Equation 4.1.

$$f_x(x) = \frac{1}{\sigma\sqrt{2\,\pi}}\, e^{\left[-\frac{(x-\mu)^2}{2\,\sigma^2}\right]} \quad (4.1)$$

Where:

μ =45 is the mean of the distribution.

σ = 26 is the standard deviation of the distribution.

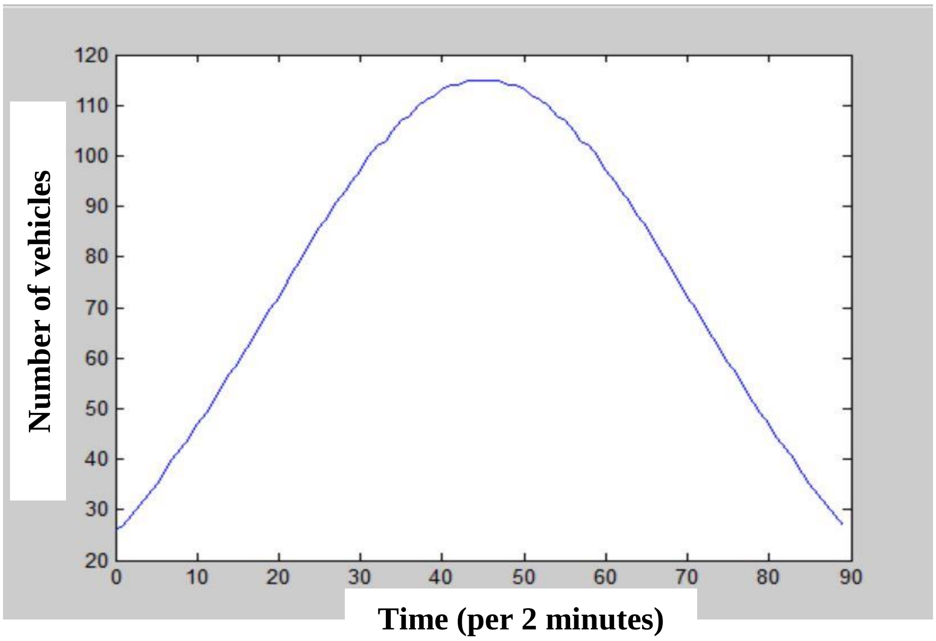
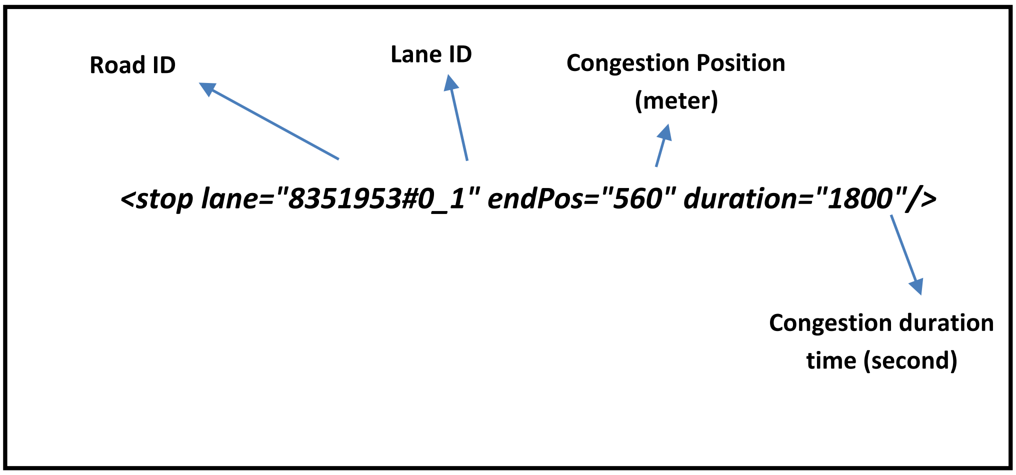


Figure 4.9. Normal distribution of vehicles

The traffic congestion happens for many reasons such as weather condition, roadwork, drivers' behavior and etc. SUMO can help to simulate traffic congestion as following instruction:



<u>Where</u>

Lane: road ID + lane ID.

End pos: congestion position on the road in meter.

Duration: congestion duration time in second, congestion time start on (5400 s) and finish on (7200 s) from simulation time.

4.5. Experiments Scenario

Based on the congestion level parameter and threshold value have three experiments scenario implement with consideration of experiments determinants that discussed in previous section as follows:

<u>Scenario 1:</u> in this scenario, the value of traffic congestion level for roads on the map less than or equal threshold value (0.6) this meaning the roads of the map in this scenario have free traffic congestion therefore applied classic Dijkstra algorithm.

<u>Scenario 2:</u> in this scenario, the value of traffic congestion level for roads on the map greater than threshold value (0.6) this meaning the roads of the map in the scenario

have traffic congestion. Although the traffic congestion is occurred but applied classic Dijkstra algorithm to evaluate its performance on the traffic jam.

Scenario 3: in the last scenario, the value of traffic congestion level for roads on the map also is greater than threshold value (0.6) also and the traffic congestion happened but, in this scenario, apply proposed improved Dijkstra via reroute vehicles and give alternative route for vehicles to evaluate its performance.

Table 4.3. shows the simulation parameters for all scenarios of the selected road in the sakarya map.

Table 4.3. Simulation parameters

Parameter	Value
Case study	Part of Sakarya map
Map size	25 km^2 (5*5)
Number of scenarios	3 scenarios
The maximum speed allowed	70km/h
Simulation time	10800 s
Number of intervals	90 (120 second per interval)
Congestion time	1800 s
Number of generated vehicles	300vehicle/km^2
Vehicle length + safety distance	10 Meters
Route generating method	Random
Threshold	0.6

4.6. Experiments Results

After three scenarios had simulated and obtained the results. the performance for both classic Dijkstra and improved Dijkstra algorithm have evaluated in case of traffic congestion based on performance metrics: related vehicles in traffic congestion, average delay time, average fuel consumption and average CO2 via using SUMO simulator and the results have been analyzed using Excel. Table 4.4. shows the comparison between performance classic Dijkstra and improved Dijkstra algorithm in traffic congestion case.

Table 4.4. The comparison between classic Dijkstra and improved Dijkstra algorithm in traffic congestion case

Parameters	Classic Dijkstra	Improved Dijkstra
Related vehicles	229 vehicles	177 vehicles
Average delay time	1075 s	828 s
Average fuel consumption	1644 l/km	1398 l/km
Average CO2	382.595kg/km	325.211kg/km

4.6.1. Related vehicles

The number of affected vehicles on the traffic congestion that has been measured by selecting the vehicles which have a delay time is greater than 270 second their destination. Figures 4.10. show the results of the number of related vehicles for both of classic Dijkstra and improved Dijkstra over the selected route network traffic congestion.

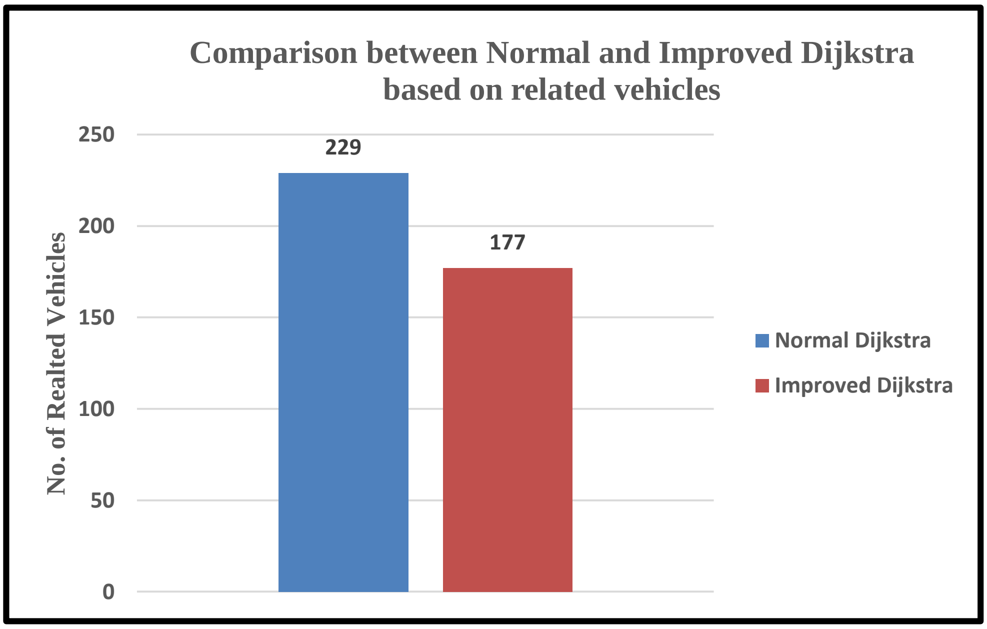


Figure 4.10. Comparison between normal and improved Dijkstra based on related vehicles

From Figure 4.10., it shows the proposed improved Dijkstra algorithm efficiency to reduced number of related vehicles in traffic congestion that directly influences to other parameters metrics such as (delay time, fuel consumption and CO2 emission) that it will be shown later. Where the improved Dijkstra have (177) related vehicles, whereas the classic Dijkstra have (229) related vehicles.

4.6.2. Average Delay Time

The average delay time for vehicles is computed by applying Equation 4.2.

$$\text{A.V. Delay time} = \frac{\sum_{i=1}^{n} V_i}{n} \tag{4.2}$$

<u>Where</u>

Vi: vehicle delay time

n: the number of related vehicles

Figure 4.11. show the results of the average delay time for both of classic Dijkstra and improved Dijkstra over the selected route network traffic congestion.

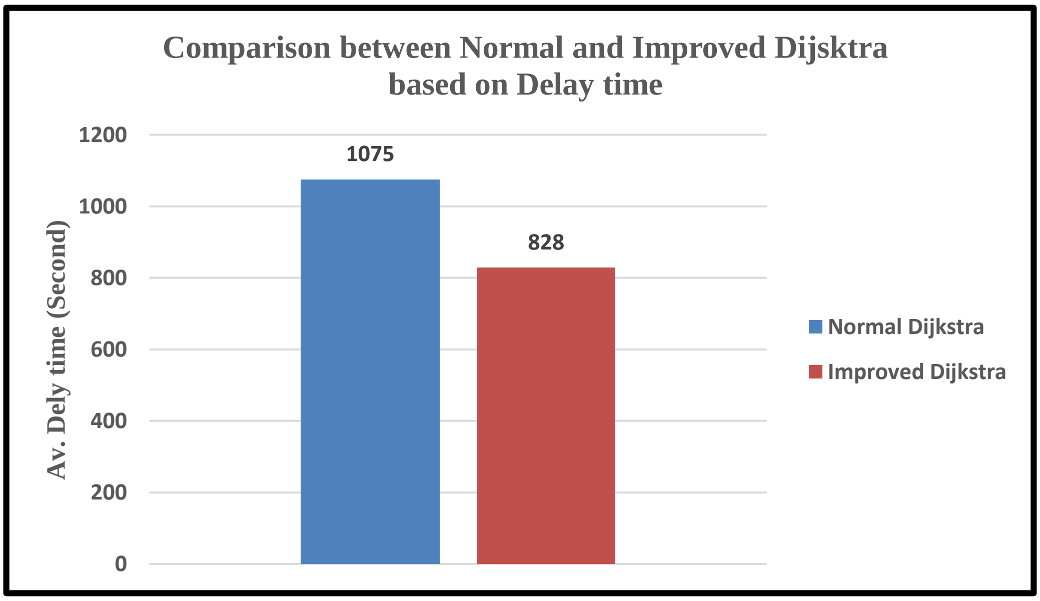


Figure 4.11. Comparison between normal and improved Dijkstra based on delay time

It is interesting to see from Figure 4.11., improved Dijkstra algorithm reduced the average delay time which is 325,211 second while classic Dijkstra have 382.595 second. It is normal for the vehicle delay time decreasing in traffic congestion with decreasing the number of related vehicles in the congested routes.

4.6.3. Average fuel consumption

It can be defined that the rate at which an engine spend fuel, expressed in units such as miles per gallon or liters per kilometer. This study used liters per kilometer. Figure 4.12. shows the results of the fuel consumption for both of classic Dijkstra and improved Dijkstra over the selected route network traffic congestion.

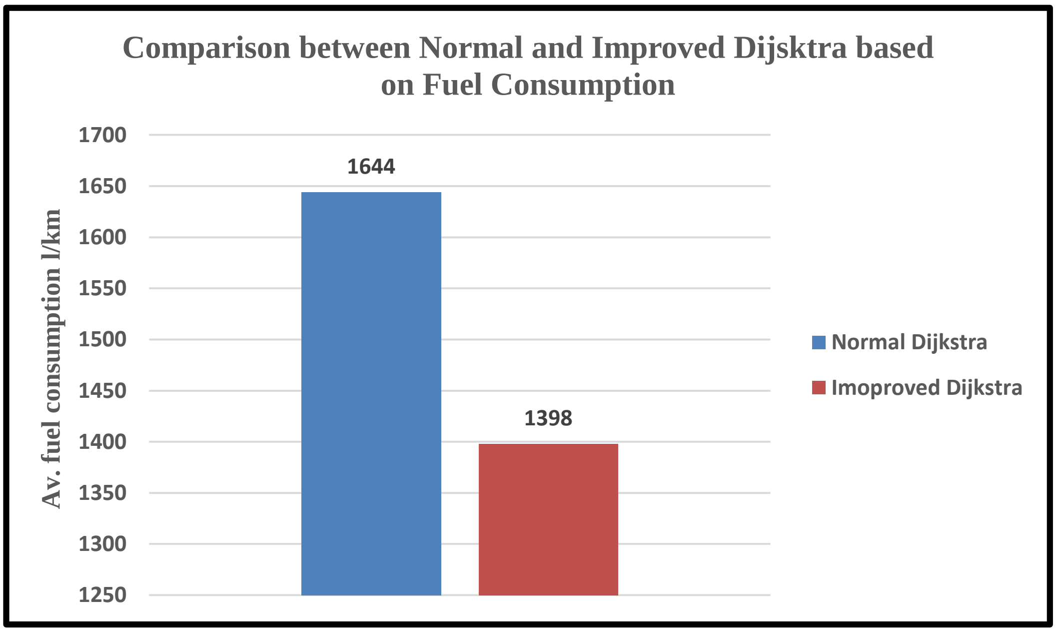


Figure 4.12. Comparison between normal and improved Dijkstra based on fuel consumption

The results are directly related to the status of vehicle if is moving or stopped, velocity and the stopped time for related vehicles during the simulation time. It can be clearly seen from Figure 4.12., that improved Dijkstra has reduced fuel consumption compared to classic Dijkstra for traffic congestion cases. where the average fuel consumption in classic Dijkstra which is 1644 l/km and in improved Dijkstra algorithm which is 1398 l/km.

4.6.4. Average CO2 emission

It is the amount of CO2 that emission from natural and human sources. The unit of measurement is kg/km. Figure 4.13. shows the results of the amount CO2 emissions for both of classic Dijkstra and improved Dijkstra over the selected road in traffic congestion

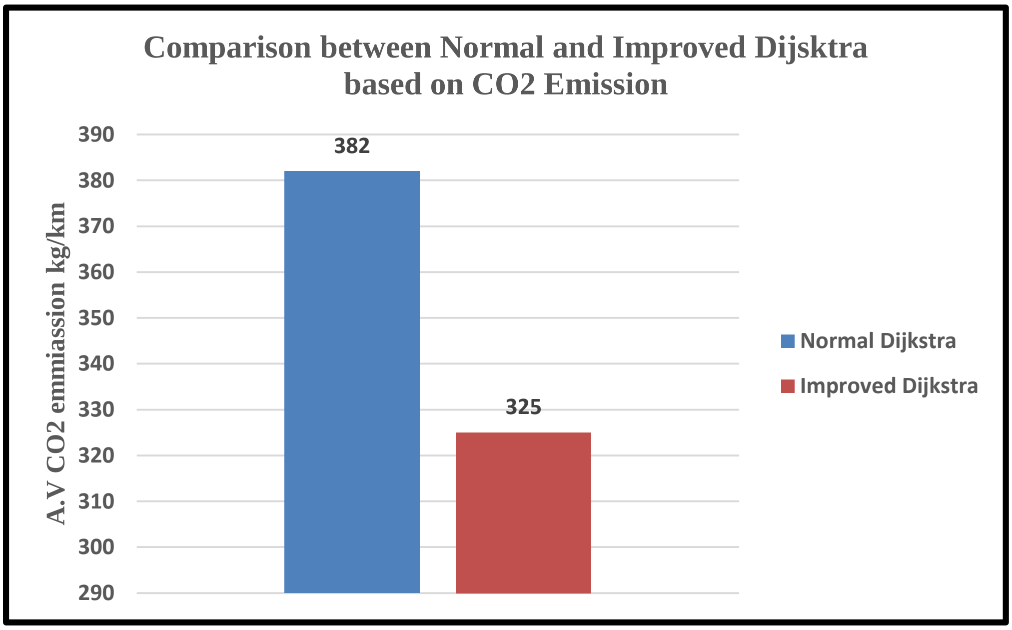


Figure 4.13. Comparison between normal and improved Dijkstra based on CO2 emission

The CO2 emissions are positively associated with fuel consumption. The increase in fuel consumption results in more CO2 emissions. It can obviously be seen from the Figure 4.13., that the amount of CO2 emissions from vehicles have been reduced by applying improved Dijkstra on the related vehicles for the traffic congestion route in compared with the classic Dijkstra.

4.7. Evaluated results

Dijkstra's is one of the most popular shortest path algorithms which can handle the shortest path but the shortest path may not be the optimal path. Therefore, this study selected Dijkstra algorithm to improved it and find optimal path. To evaluated the results and comparison between classic Dijkstra and improved Dijkstra, two traffic congestion scenarios have been simulated with the classic Dijkstra and improved Dijkstra over the part of Sakarya map. As a result, the performance of improved Dijkstra is better than classic Dijkstra because the results approved of Figure 4.10. clarifies that the improved Dijkstra had decreases the related vehicles in traffic congestion by approximately 23%. The reason for this is with the classic Dijkstra in traffic congestion has given all related vehicles one path is shortest path but the improved Dijkstra is rerouting and giving alternative routes for all related vehicles to reach same aim. Reducing number of related vehicles in traffic congestion is real affect to another parameter metrics for example average delay time.

From Figure 4.11, it shows the improved Dijkstra minimize average delay time by approximately 23% also because the average delay time on classic Dijkstra is 1075 second and on Improved Dijkstra is 828 second.

On the environmental side, the improved Dijkstra reduce average fuel consumption by approximately 15% for related vehicles in traffic congestion and this ratio certainly has a relationship with amount of CO_2 emission that diminished by approximately 14% as shown in Figure 4.12. and 4.13.

CHAPTER 5. CONCLUSIONS

With the increasing number of vehicles on the road network in the world, various problems have emerged, one of the most serious is traffic jams. This thesis provides efficient solution for traffic congestion that can effectively and practically reduce traffic congestion on the road network via improvement of Dijkstra algorithm. The improved Dijkstra algorithm has added traffic congestion level to the classic Dijkstra for the purpose of increasing the performance of vehicles and reducing average delay time. To simulate vehicles in a real road traffic environment, this study has added four features to the SUMO 1.8.0 simulator package which are: time period, rush hour, number of vehicles, and routing algorithm, as discussed of details in Section 4.3. In addition, it has used the normal distribution method for generating vehicles on the road network. To evaluate the performance of the proposed improved Dijkstra algorithm, part of Sakarya map have been selected as case study and two traffic congestion experiments scenarios for both classic Dijkstra and improved Dijkstra. The traffic congestion scenarios are built with consideration experiments determinants. Overall, the proposed Dijkstra has improved performance is 23% for both related vehicles in traffic congestion and average delay time compared with the classic Dijkstra (with traffic congestion). on the environmental side, the improved Dijkstra has a great impact in reducing expenses and pollution where reduce average fuel consumption by approximately 15% for related vehicles in traffic congestion and amount of CO_2 emission that diminished by approximately 14%.

In this study, the Dijkstra algorithm was improve using SUMO simulator, and MATLAB software. The next step is to:

1. Examine the performance of improved Dijkstra algorithm on the real road traffic environment.

2. Develop the proposed improved Dijkstra algorithm via integrate another traffic congestion metrics such as speed, throughput and Level-Of-Service (LOS).

3. Employ (VANTE) technology in vehicle route guidance systems, which will play a vital role in alleviate traffic congestion by exchanging traffic information between vehicle to vehicle and the vehicle to infrastructure.

REFERENCES

[1] M. A. Mondal and z. Rehena, “Identifying Traffic Congestion Pattern using K-means Clustering Technique,” Proc. - 2019 4th Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2019, pp. 1–5, 2019, doi: 10.1109/IoT-SIU.2019.8777729.

[2] N. McCarthy, “The Cities With The Worst Traffic Congestion,” 2021.

[3] A. Bhandari, V. Patel, and M. Patel, “A Survey on Traffic Congestion Detection and Rerouting Strategies,” Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018, no. Icoei, pp. 42–44, 2018, doi: 10.1109/ICOEI.2018.8553965.

[4] C. Ding, Y. Chen, and Y. Chen, “Construction of urban intelligent traffic sharing information platform based on GIS,” Proc. - 2020 5th Int. Conf. Electromechanical Control Technol. Transp. ICECTT 2020, pp. 579–584, 2020, doi: 10.1109/ICECTT50890.2020.00132.

[5] J. Jin, H. Guo, J. Xu, X. Wang, and F. Y. Wang, “An End-to-End Recommendation System for Urban Traffic Controls and Management Under a Parallel Learning Framework,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 3, pp. 1616–1626, 2021, doi: 10.1109/TITS.2020.2973736.

[6] P. Roychowdhury and S. Das, “Smart Urban Traffic Management System,” no. May, pp. 1–2, 2015, doi: 10.13140/RG.2.1.3414.6324.

[7] Q. Tan, Z. Wang, Y. S. Ong, and K. H. Low, “Evolutionary optimization-based mission planning for UAS traffic management (UTM),” 2019 Int. Conf. Unmanned Aircr. Syst. ICUAS 2019, pp. 952–958, 2019, doi: 10.1109/ICUAS.2019.8798078.

[8] N. B. Hounsell, B. P. Shrestha, J. Piao, and M. McDonald, “Review of urban traffic management and the impacts of new vehicle technologies,” IET Intell. Transp. Syst., vol. 3, no. 4, pp. 419–428, 2009, doi: 10.1049/iet-its.2009.0046.

[9] A. Zear, P. K. Singh, and Y. Singh, “Intelligent transport system: A progressive review,” Indian J. Sci. Technol., vol. 9, no. 32, 2016, doi: 10.17485/ijst/2016/v9i32/100713.

[10] M. Veres and M. Moussa, “Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey of Emerging Trends,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 8, pp. 3152–3168, 2020, doi: 10.1109/TITS.2019.2929020.

[11] M. Chowdhury and K. Dey, "Intelligent transportation systems-a frontier for breaking boundaries of traditional academic engineering disciplines [Education]," IEEE Intell. Transp. Syst. Mag., vol. 8, no. 1, pp. 4–8, 2016, doi: 10.1109/MITS.2015.2503199.

[12] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-Reliability and Low-Latency Wireless Communication for Internet of Things: Challenges, Fundamentals, and Enabling Technologies," IEEE Internet Things J., vol. 6, no. 5, pp. 7946–7970, 2019, doi: 10.1109/JIOT.2019.2907245.

[13] K. Padmanabhan and P. Kamalakkannan, "A study on energy efficient routing protocols in wireless sensor networks," Eur. J. Sci. Res., vol. 60, no. 4, pp. 517–529, 2011, doi: 10.5121/ijdps.2012.3326.

[14] S. G.C, S. Shirabadagi, and R. Hegadi, "High Density Traffic Management using Image background subtraction Algorithm," Int. J. Comput. Appl., vol. 4, no. February, pp. 10–15, 2014, [Online]. Available: https://pdfs.semanticscholar.org/ad21/3b4e211ccbce785d218c7ef95bdb5c0cab5c.pdf.

[15] H. Zhu, J. Wang, K. Xie, and J. Ye, "Detection of Vehicle Flow in Video Surveillance," 2018 3rd IEEE Int. Conf. Image, Vis. Comput. ICIVC 2018, pp. 528–532, 2018, doi: 10.1109/ICIVC.2018.8492794.

[16] Sightcorp, "What is Vehicle Detection Software," Deepsite Solution, 2020.

[17] A. Mishra and A. Priya, "A Comprehensive Study on Intelligent Transportation Systems," Smart Moves J. Ijoscience, vol. 4, no. 10, p. 10, 2018, doi: 10.24113/ijoscience.v4i10.167.

[18] Y. Liu, "Big Data Technology and Its Analysis of Application in Urban Intelligent Transportation System," Proc. - 3rd Int. Conf. Intell. Transp. Big Data Smart City, ICITBS 2018, vol. 2018-Janua, pp. 17–19, 2018, doi: 10.1109/ICITBS.2018.00012.

[19] E. Ben Hamida, H. Noura, and W. Znaidi, "Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures," Electron. , vol. 4, no. 3, pp. 380–423, 2015, doi: 10.3390/electronics4030380.

[20] G. Yan and Q. Qin, "The Application of Edge Computing Technology in the Collaborative Optimization of Intelligent Transportation System Based on Information Physical Fusion," IEEE Access, vol. 8, pp. 153264–153272, 2020, doi: 10.1109/ACCESS.2020.3008780.

[21] ARM, "Accelerating Autonomous Vehicle Technology," IEEE Spectrum, 2020.

[22] M. A. Population, “Sakarya, Turkey Metro Area Population 1950-2021,” Macrotrend.

[23] Statista Research Department, “Annual number of maritime accidents in Turkey from 2001 to 2019,” 2021.

[24] S. Şener, “Road safety in Turkey,” World Health Organization, 2020.

[25] C. Rhodes and S. Djahel, “TRADER:Traffic light phases aware driving for reduced traffic congestion in smart cities,” 2017 Int. Smart Cities Conf. ISC2 2017, 2017, doi: 10.1109/ISC2.2017.8090783.

[26] G. Wen, N. Huang, and C. Wang, “Network Congestion Diffusion Model Considering Congestion Distribution Information,” IEEE Access, vol. 7, pp. 102064–102072, 2019, doi: 10.1109/ACCESS.2019.2931354.

[27] T. Afrin and N. Yodo, “A survey of road traffic congestion measures towards a sustainable and resilient transportation system,” Sustain., vol. 12, no. 11, pp. 1–23, 2020, doi: 10.3390/su12114660.

[28] M. Pi, H. Yeon, H. Son, and Y. Jang, “Visual Cause Analytics for Traffic Congestion,” IEEE Trans. Vis. Comput. Graph., vol. 27, no. 3, pp. 2186–2201, 2021, doi: 10.1109/TVCG.2019.2940580.

[29] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, “Traffic speed prediction under non-recurrent congestion: based on lstm method and beidou navigation satellite system data,” IEEE Intell. Transp. Syst. Mag., vol. 11, no. 2, pp. 70–81, 2019, doi: 10.1109/MITS.2019.2903431.

[30] R. Hu, Y. Xia, C. Y. Hsu, H. Chen, and W. Xu, “Traffic Intersection Detection Using Floating Car Data,” 2020 5th IEEE Int. Conf. Big Data Anal. ICBDA 2020, pp. 116–120, 2020, doi: 10.1109/ICBDA49040.2020.9101259.

[31] C. T. Lam, H. Gao, and B. Ng, “A real-time traffic congestion detection system using on-line images,” Int. Conf. Commun. Technol. Proceedings, ICCT, vol. 2017-Octob, pp. 1548–1552, 2018, doi: 10.1109/ICCT.2017.8359891.

[32] D. Jin, S. Zhang, X. Huo, and F. Yang, “A Corner Detection Method for Conventional Light Field Camera by Jointly Using Line-Features,” IEEE Access, vol. 8, pp. 75884–75893, 2020, doi: 10.1109/ACCESS.2020.2989640.

[33] E. D’Andrea and F. Marcelloni, “Detection of traffic congestion and incidents from GPS trace analysis,” Expert Syst. Appl., vol. 73, pp. 43–56, 2017, doi: 10.1016/j.eswa.2016.12.018.

[34] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, “The Cellular Network as a Sensor: From Mobile Phone Data to Real-Time Road Traffic Monitoring,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 5, pp. 2551–2572, 2015, doi: 10.1109/TITS.2015.2413215.

[35] A. Devangavi and R. Gupta, “Routing Protocols in VANET- A Survey,” Int. Conf. Smart Technol. Smart Nation, vol. 1, pp. 163–167, 2017.

[36] H. Kaur and Meenakshi, “Analysis of VANET geographic routing protocols on real city map,” RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc., vol. 2018-Janua, pp. 895–899, 2017, doi: 10.1109/RTEICT.2017.8256727.

[37] R. Baweja, R. Gupta, and N. K. Bhagat, “Improved congestion avoidance and resource allocation algorithm,” Proc. 2014 2nd Int. Conf. “Emerging Technol. Trends Electron. Commun. Networking”, ET2ECN 2014, pp. 1–5, 2015, doi: 10.1109/ET2ECN.2014.7044991.

[38] S. M. Deshmukh and B. N. Savant, “Designing an optimized smart device in vehicle for detection and avoidance of traffic congestion,” Conf. Adv. Signal Process. CASP 2016, pp. 33–36, 2016, doi: 10.1109/CASP.2016.7746133.

[39] C. Backfrieder, G. Ostermayer, and C. F. Mecklenbrauker, “Increased traffic flow through node-based bottleneck prediction and V2X communication,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 2, pp. 349–363, 2017, doi: 10.1109/TITS.2016.2573292.

[40] J. Pan, M. A. Khan, I. S. Popa, K. Zeitouni, and C. Borcea, “Proactive vehicle re-routing strategies for congestion avoidance,” Proc. - IEEE Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2012, no. 1, pp. 265–272, 2012, doi: 10.1109/DCOSS.2012.29.

[41] D. L. Schrank, S. M. Turner, and T. J. Lomax, “Estimates of urban roadway congestion-1990,” Res. Rep., vol. 1, no. 2, pp. 1131–5, 1993, [Online]. Available: http://tti.tamu.edu/documents/1131-5.pdf.

[42] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, “Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 11, pp. 3204–3217, 2017, doi: 10.1109/TITS.2017.2687620.

[43] R.-P. Schäfer, K.-U. Thiessenhusen, and P. Wagner, “A Traffic Information System By Means of Real-Time Floating-Car Data,” ITS world Congr., vol. m, no. January 2002, pp. 1–8, 2002, [Online]. Available: http://elib.dlr.de/6499/01/chicago_final.pdf.

[44] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta, "A cooperative approach to traffic congestion detection with Complex Event processing and VANET," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 2, pp. 914–929, 2012, doi: 10.1109/TITS.2012.2186127.

[45] Y. Xu, Y. Wu, J. Xu, and L. Sun, "Efficient detection scheme for urban traffic congestion using buses," Proc. - 26th IEEE Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2012, pp. 287–293, 2012, doi: 10.1109/WAINA.2012.62.

[46] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative Vehicle-to-Vehicle communications," Proc. - Conf. Local Comput. Networks, LCN, pp. 606–612, 2010, doi: 10.1109/LCN.2010.5735780.

[47] J. W. Wedel, B. Schünemann, and I. Radusch, "V2X-based traffic congestion recognition and avoidance," I-SPAN 2009 - 10th Int. Symp. Pervasive Syst. Algorithms, Networks, pp. 637–641, 2009, doi: 10.1109/I-SPAN.2009.71.

[48] I. Leontiadis, G. Marfia, D. MacK, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1537–1548, 2011, doi: 10.1109/TITS.2011.2161469.

[49] K. Wei, Y. Gao, W. Zhang, and S. Lin, "A Modified Dijkstra's Algorithm for Solving the Problem of Finding the Maximum Load Path," 2019 IEEE 2nd Int. Conf. Inf. Comput. Technol. ICICT 2019, pp. 10–13, 2019, doi: 10.1109/INFOCT.2019.8711024.

[50] R. J. Venkat, "PATH FINDING - Dijkstra ' s Algorithm," Appl. Math. Model., pp. 2454–2462, 2014.

[51] K. Magzhan and H. Jani, "A Review And Evaluations Of Shortest Path Algorithms," Int. J. Sci. Technol. Res., vol. 2, no. 6, pp. 99–104, 2013, [Online]. Available: http://www.ijstr.org/final-print/june2013/A-Review-And-Evaluations-Of-Shortest-Path-Algorithms.pdf.

[52] A. Alyasin, E. I. Abbas, and S. D. Hasan, "An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method," 4th Sci. Int. Conf. Najaf, SICN 2019, pp. 11–14, 2019, doi: 10.1109/SICN47020.2019.9019345.

[53] L. Wenzheng, L. Junjun, and Y. Shunli, "An Improved Dijkstra's Algorithm for Shortest Path Planning on 2D Grid Maps," ICEIEC 2019 - Proc. 2019 IEEE 9th Int. Conf. Electron. Inf. Emerg. Commun., pp. 438–441, 2019, doi: 10.1109/ICEIEC.2019.8784487.

[54] S. Risald, Antonio E., "Best Routes Section Using Dijkstra and Floyd-Warshall Algorithm," in 2017 International Conference on Information & Communication Technology and System (ICTS), 2017, pp. 155–158.

[55] N. Makariye, "Towards Shortest Path Computation using Dijkstra Algorithm," 2017 Int. Conf. IoT Appl. (pp. 1-3). IEEE., pp. 1–3.

[56] G. Qing, Z. Zheng, and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," Proc. 29th Chinese Control Decis. Conf. CCDC 2017, pp. 7138–7143, 2017, doi: 10.1109/CCDC.2017.7978471.

[57] I. Chahbi, D. Ben Amara, and A. Belghith, "A Novel Route Guidance Algorithm Using Beamforming Techniques for Vehicular Networks," 7th IEEE LCN Work. User Mobil. Veh. Networks (ON-MOVE 2013) A, pp. 168–174, 2013.

[58] O. A. Gbadamosi and D. R. Aremu, "Design of a Modified Dijkstra's Algorithm for finding alternate routes for shortest-path problems with huge costs.," 2020 Int. Conf. Math. Comput. Eng. Comput. Sci. ICMCECS 2020, pp. 13–18, 2020, doi: 10.1109/ICMCECS47690.2020.240873.

[59] M. Wei and Y. Meng, "Research on the optimal route choice based on improved Dijkstra," Proc. - 2014 IEEE Work. Adv. Res. Technol. Ind. Appl. WARTIA 2014, pp. 303–306, 2014, doi: 10.1109/WARTIA.2014.6976257.

[60] Yujin and G. Xiaoxue, "Optimal Route Planning of Parking Lot Based on Dijkstra Algorithm," Proc. - 2017 Int. Conf. Robot. Intell. Syst. ICRIS 2017, pp. 221–224, 2017, doi: 10.1109/ICRIS.2017.62.

[61] R. Data, "Introduction to XML," W3Schools, 2019.

[62] J. A. Sanguesa et al., "Vehicle density and roadmap topology issues when characterizing vehicular communications," Proc. - 2015 IEEE 14th Int. Symp. Netw. Comput. Appl. NCA 2015, pp. 200–203, 2016, doi: 10.1109/NCA.2015.23.

RESUME

Name / Surname : **Amenah Sufyan Mhmood THABIT**

EDUCATION

Level	School	Graduation Year
Undergrad	University of Mosul / College of computer sciences and mathmtical /software	2009
Secondary School	Quriba High School	2005

EXPERIENCE

Year	Location	Task
2012-present	Iraq Parliament	Programmer
2014-2018	BQc company at UK	Marketing manager
2010-2012	Mosul collage	Programmer

LANGUAGE

English

PUBLICATIONS

1. Performance Evaluation of Manet Routing Protocols Using Network Simulator NS2.

HOBIES

Travel, sport and hand made