FINDING POSSIBLE TRANSIT-ROUTES IN HYPERGRAPH YANGON BUS NETWORK

WAI MAR HLAING

UNIVERSITY OF COMPUTER STUDIES, YANGON

DECEMBER, 2020

Finding Possible Transit-Routes in Hypergraph Yangon Bus Network

Wai Mar Hlaing

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

December, 2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

………………   …………………..

Date   Wai Mar Hlaing

ACKNOWLEDGEMENTS

First and foremost, I would like to say a big thank-you to everyone concerned from the Ministry of Education for giving me a special chance to study for a doctorate in Information Technology at the University of Computer Studies, Yangon.

I would also like to thank our Professor Dr. Mie Mie Thet Thwin, Rector, University of Computer Studies, Yangon (UCSY). She is always full of good ideas not only about research work but also about the domain of computer science.

I sincerely express my deep sense of gratitude to my supervisor, Professor Dr. Myint Myint Sein, Head of Geographic Information System Lab, University of Computer Studies, Yangon, for her invaluable guidance and supervision as well as for her extraordinary cooperation. Moreover, she has always encouraged me to publish valuable research papers and products which provide support for my research.

I wish to express my sincere thanks and appreciation to Dr. Khine Khine Oo, Professor and Course-coordinator of our Ph.D 10th batch, University of Computer Studies, Yangon, for her kindness, leadership, guidance, suggestions and care. She has been able to motivate us PhD candidates to work hard throughout our research work.

I would like to express my special thanks to the external examiner, Dr. Win Zaw, Professor, Department of Computer Engineering and Information Technology, Yangon Technological University, for his excellent advice and comments in the preparation for thesis and for appreciating my research work.

I would also like to describe my thanks to Dr. Ah Nge Htwe, Professor, Department of Computer Science, University of Computer Studies, Yangon, for giving the constructive suggestions in the research work.

My grateful thanks also go to Daw Aye Aye Khine, Associate Professor, and Head of English Department, University of Computer Studies, Yangon, who is willing to give me assistance with any problems that may arise whenever I need some help from her.

I am also thankful to all my teachers and friends for their constructive discussions, perseverance and encouragement during this research work. In addition, I also thank my friends for giving the required help whenever I hold the seminars that I need to do to finish my Ph.D.

I would also like to offer my special thanks to my brilliant husband and my lovely sisters for helping me in every way they can as well as for their humour and lightheartedness during this time-consuming effort of mine. I also want to thank teachers and students from San Myint Maw Private High School for participating in my subjective evaluation.

Above all, my heartfelt thanks are due to my beloved parents for their constant moral support and mellifluous affection which have helped me become what I am today.

Finally, I sincerely acknowledge the efforts of all those who have directly or indirectly helped me in completing my thesis successfully.

ABSTRACT

Finding different possible ways in Public Bus Transportation System are essential for users. Most systems implement to search one shortest route for advising the user. Sometimes, the shortest route is not the best for users because they want to choose the desired route depend on the current condition. Transportation is a main sector for a country. The people in Yangon mainly rely on bus, taxi and private cars. Travelling in train is not popular in Yangon Region because it cannot reach to any place within the Yangon area. The system develops for two networks: Road network and Public bus network. Road network is developed for Yangon downtown area and Bus network is applied for Public transportation system. In Yangon downtown network, junction points of the road are assigned as nodes. Latha, Lanmadaw, Pabeldan, Kyauktada, Pazundaung and Botahtaung townships are contained Yangon downtown among of the 33 townships in Yangon. Simulated graph for Yangon downtown network is implemented using Java 2 Standard Edition (J2SE) technology. When the user enters start and destination points, two shortest results are described on the graph. For public bus transportation network, there are 33 townships coverage in Yangon and it contains about 2000 bus stops and many connections over 84 bus lines type and these are stored in oracle database. Keyword search technique can be used for inserting start bus stop and end bus stop. In addition, two Languages: Myanmar and English can be used for optimal routes finding. Client Server architecture is used for public transportation network. Java 2 Enterprise Edition (J2EE) Tomcat server and Android studio is used for implementing the public bus system. When the user enters start and end points, the system checks these points and then it searches the shortest and possible routes using proposed heuristic based search named Combine Forward and Backward Heuristic Search (CFBHS) algorithm. Each Route can be viewed on Google Map using Google Map Application Interface (API). Furthermore, the system also supports the bus numbers for each route. For unfamiliar users about Yangon Region, the system uses location-based alarm technique to get the alarm message when the destination is reached. According to the experimental results, proposed CFBHS algorithm is faster than Bi-directional Dijkstra and Popular heuristic search algorithm A*. Moreover, the proposed algorithm can produce not only the shortest but also the possible paths while A* can give one shortest result.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **i**

ABSTRACT **iii**

TABLE OF CONTENTS **iv**

LIST OF FIGURES **viii**

LIST OF TABLES **xii**

LIST OF EQUATIONS **xiii**

1. INTRODUCTION

1.1 Transportation Network………………………………………………... 2

1.2 Public Bus Transportation……………………..………………......... 2

1.3 Finding the Possible Transit Routes for Public Bus Transportation System………………………………………………………………. 2

1.4 Motivation of Thesis…………………………………………………… 3

1.5 Objectives of the Thesis………………………………………………... 4

1.6 Problems and Contributions of the Thesis. …..… ……… ……………. 5

1.7 Organization of the Thesis……………………………………………… 5

2. LITERATURE REVIEW AND RELATED WORK

2.1 Popular Environments Based on GIS Technology …… …… …………. 8

2.2 Spatial Data and Coordinate System …………………………………… 9

2.2.1 Geographic Coordinate Systems…………………………………. 10

2.3 Global Positioning System……………………………………………... 14

2.4 Google Earth…..… 17

2.4.1 Google Map and Open Street Map……………………………….. 18

2.4.2 Road Network………... 22

2.5 Partitioning Techniques……………………………………………….. 26

2.6 Public Transportation System and Network Optimization …… ……. .. 29

2.7 Location Based Services in Android……………................................. 32

2.8 Shortest Path Problems……………………...................................... 35

2.9 Summary………………………………………………………………. 37

3. THEORETICAL BACKGROUND OF OPTIMAL ROUTE FINDING

3.1 Finding the Optimal Path on the Road Network …… …… ……. …… 39

3.2 Shortest Path Algorithms………………………………………………. 39

3.2.1 A* Algorithm ………………………………………………… 40

3.2.1.1 Heuristic Function ……………………………………… 43

3.2.2 Dijkstra's Algorithm …………………………………………… 45

3.2.2.1 Working Example of Dijkstra's Algorithm……………… 46

3.2.3 Bidirectional Dijkstra Algorithm……………………………… 48

3.2.4 Problems of Related Shortest Path Algorithms………………….. 49

3.3 Summary………………………………………………………………. 49

4. IMPLEMENTATION OF THE PUBLIC BUS SYSTEM

4.1 Preprocessing Stage for the Proposed System………………………… 52

4.1.1 Yangon Region and Yangon Bus Network……………………… 53

4.1.2 Yangon Bus System…………………………………………… 54

4.1.3 Network Connectivity…………………………………………… 56

4.1.3.1 The Nature of the Bus Stop……………………………… 56

4.1.4 GPS Etrex-10 Handheld Device………………………………… 57

4.2 Data Structure of Public Transportation……………………………… 58

4.2.1 Data File………………………………………………………… 59
4.2.1.1 Extensible Markup Language (Xml)…………………....... 59
4.2.2 JavaScript Object Notation (JSON)……………………………… 60
4.2.3 Spatial Database…………………………………………………. 61
4.3 Combined Forward and Backward Heuristic Search (CFBHS)
Algorithm………………………………………………………………… 62
4.3.1 Proposed Method ……………………………………………….. 63
4.3.2 Subdivision Scheme……………………………………………... 64
4.3.3 Choosing the Next Expanded Node……………………………… 65
4.4 Finding Shortest and Possible Paths Using CFBHS…………………… 65
4.5 Working Example……………………………………………………... 70
4.6 Transit Nodes Information…………………………………………….. 71
4.6.1 Direct Route……………………………………………………... 71
4.6.2 Transit Route…………………………………………………...... 72
4.7 Summary………………………………………………………………. 73

5. EXPERIMENTAL RESULTS AND EVALUATION

5.1 Using Different Threshold Values for Different Components………… 75
5.1.1 Testing Yangon Downtown Map with Different Parameterized
Threshold Values…………………………………………….... 77
5.1.2 Testing Yangon Region with Different Parameterized
Threshold Values…………………………………………….. 78
5.2 Time Complexity for Yangon Region……………………….............. 80
5.2.1 Statistical Analysis of Undirected Road Network and Directed
Hypergraph Network………………………………………... 83

5.2.2 Average Number of Nodes that Need to Compare for Going the Next Level …………………………………………………….. 85
5.3 Performance Analysis………………………………………………... 87
5.3.1 Performance Analysis of CFBHS Algorithm………………….. 87
5.3.2 Performance Analysis of A* Algorithm……………………….. 88
5.3.3 Performance Comparison between CFBHS and A* Algorithm… 89
5.4 Implementation of Yangon Bus System Using Proposed Algorithm … 89
5.4.1 User Input and Bus Stop Information………………………….. 90
5.4.2 Finding the Routes between Start and End Bus Stops…………. 93
5.5 Subjective Evaluation for Proposed System………………………….. 95
5.6 Summary……………………………………………………………… 97
6. CONCLUSION AND FURTHER EXTENSION
6.1 Limitations…………………………………………………………… 100
6.2 Further Extension…………………………………………………... 100
LIST OF ACRONYMS……………………………………………………… 101
AUTHOR'S PUBLICATIONS………………………………………………… 103
BIBLIOGRAPHY………………………………………………………… 105

LIST OF FIGURES

Figure 2.1	Examples of Latitudinal Lines on the Left and Examples of Longitudinal Lines on the Right………………………………….	10
Figure 2.2	Latitude Coordinate……………………………………………	11
Figure 2.3	Longitude Coordinate………………………………………….	11
Figure 2.4	Semi-minor Axis and Semi-major Axis………………………….	12
Figure 2.5	Earth’s Geoid with Gravitational Field in Rainbow Colors………	13
Figure 2.6	Two Different Geographic Coordinate System…………………..	14
Figure 2.7	Global Positioning System with Orbit on the Globe……………..	15
Figure 2.8	Some GPS Devices in Different Applications……………………	16
Figure 2.9	Google Earth…………………………………………………..	17
Figure 2.10	Two Different Views of Google Map…………………………….	18
Figure 2.11	System Overview of Land Transportation Infrastructure………...	19
Figure 2.12	Showing Road on Google Map……………………………………	20
Figure 2.13	Yangon Area on Open Street Map………………………………..	21
Figure 2.14	Bus Stops of Yangon City on Google Earth………………………	22
Figure 2.15	Different Types of Graph…………………………………………	23
Figure 2.16	The Network Structure of Hypergraph…………………………..	24
Figure 2.17	Hypergraph Bus Network………………………………………..	25
Figure 2.18	Four Drawings of the Hypergraph H= (V, E) with V = {1,2,3,4,5,6} and E = {(1,2), (1,2,3), (2,4), (3,4,5,6)} ………	25
Figure 2.19	The Relationship between Simple Graph, Hypergraph and Partial-Order Hypergraph………………………………………………..	26
Figure 2.20	Graph Partitioning for Germany Map……………………………	27
Figure 2.21	Directed Graph…………………………………………………..	28
Figure 2.22	Yangon Bus Network……………………………………………	32

Figure 2.23	Generating Google Map API Key……………………………….	33
Figure 2.24	Android Manifest File……………………………………………	34
Figure 2.25	Interaction between the Mobile Device and the Processing Steps of Location Base Service……………………………………………	35
Figure 3.1	The Heuristic Values between Nodes……………………………	41
Figure 3.2	Comparing Heuristic Distance Values among the Nodes………..	42
Figure 3.3	A Weighted Graph………………………………………………..	42
Figure 3.4	Manhattan Distance………………………………………………	44
Figure 3.5	Diagonal Distance………………………………………………..	44
Figure 3.6	Euclidean Distance……………………………………………….	45
Figure 3.7	Example of Dijkstra Algorithm…………………………………..	46
Figure 3.8	Choosing the Smallest Distance Value using Dijkstra……………	47
Figure 3.9	Bi-directional Dijkstra……………………………………………	48
Figure 3.10	Meeting at the Same Point………………………………………..	48
Figure 4.1	Architecture of the Proposed System……………………………	52
Figure 4.2	Townships in Yangon Region……………………………………	54
Figure 4.3	Different Colors in Bus Numbers………………………………..	55
Figure 4.4	Nature of Yangon Bus Network…………………………………	57
Figure 4.5	GPS Device………………………………………………………	57
Figure 4.6	The Functions of GPS Garmin eTrex 10…………………………	58
Figure 4.7	Xml Tree Structure of Proposed System…………………………	60
Figure 4.8	Sql Database in QGIS……………………………………………	61
Figure 4.9	System Flow Diagram……………………………………………	62
Figure 4.10	Demonstration of Subdivision……………………………………	64
Figure 4.11	Undirected Graph………………………………………………....	71
Figure 4.12	Direct Route between Two Bus stops……………………………	72

Figure 4.13 Two Transactions between Two Bus Stops……………………… 73
Figure 4.14 Three Transactions between Two Bus Stops…………………… 73
Figure 5.1 Constructing Yangon Downtown Spatial Data Network into Graph Data Network…………………………………………….............. 76
Figure 5.2 Using the Different Threshold Values…………………………… 78
Figure 5.3 The Number of Vertices according to Degree…………………… 79
Figure 5.4 Yangon Bus Network with different threshold values…………… 79
Figure 5.5 Finding the Shortest and Possible Routes for Yangon Downtown.. 80
Figure 5.6 Time Complexity of Three Algorithms for Yangon Downtown…. 81
Figure 5.7 The Number of Paths according to rs Value for Yangon Downtown 82
Figure 5.8 Comparison of Time complexity of CFBHS for Six Townships and 33 Townships in Yangon Region…………………………………. 82
Figure 5.9 The Variance of Time Complexity depends on the Number of Results of Each Subtree………………………………………….. 83
Figure 5.10 Average time complexity for A* and proposed algorithm with Different Threshold Values……………………………………… 84
Figure 5.11 Time Complexity among Different Algorithms depends on Execution Time………………………………………………….. 84
Figure 5.12 Time Complexity of the Proposed Algorithm depends on Different Data Sizes……………………………………………………… 85
Figure 5.13 Compared Nodes using A* Algorithm…………………………… 86
Figure 5.14 Compared Nodes using CFBHS Algorithm……………………… 86
Figure 5.15 Compared Nodes between CFBHS and A* during Five Times….. 87
Figure 5.16 Branching the Tree for Finding the Shortest Path between Node A and Node L…………………………………………………….. 88
Figure 5.17 Data Input using Myanmar or English Language………………… 90
Figure 5.18 Bus Stops in Yangon Bus System………………………………… 91

Figure 5.19 Bus Lines and Bus Stop Information of Each Bus Line in Yangon Bus System………………………………………………………. 92
Figure 5.20 Finding Routes and Bus Numbers in Yangon Bus System………. 93
Figure 5.21 Displaying the Route on Google Map…………………………… 94
Figure 5.22 Home Page of System……………………………………………. 95
Figure 5.23 Users' Satisfaction in Different Bus Lines' Color………………. 96
Figure 5.24 Users' Satisfaction about the Processing Time …………………. 96

LIST OF TABLES

Table 3.1 Straight Line Distance Table…………………………………….. 41

Table 4.1 The Relations between Degree, Minute and Seconds…………… .. 53

Table 4.2 Bus Routes……………... 55

Table 5.1 Detail Performance Analysis of CFBHS Algorithm……………… 89

LIST OF EQUATIONS

Equation 3.1 40

Equation 3.2 44

Equation 3.3 45

Equation 3.4 45

Equation 4.1 53

Equation 4.2 63

Equation 4.3 72

Equation 4.4 72

CHAPTER 1
INTRODUCTION

Networks have always been essential in transport geography and telecommunication. They contain as an important sector for all businesses today because of giving the fast service for work done. Network analysis studies the nature of the nodes and their relationships between them. Graph theory is a fundamental approach to solve the problems in transportation networks. Transportation network is constructing the spatial data as the mathematical graph. Google earth, Arc Geographical Information System (ArcGIS) and Quantum Geographical Information System (QGIS) are popular geographical tools to collect the spatial data and spatial data analysis. Depending on the size and nature of the graph, different short finding algorithms are used in transportation networks.

Some graph networks are very complex and these are difficult to find the shortest path. The processing time is a considerable factor for short finding mechanisms. Classical short finding algorithms such as Dijkstra are matured algorithms for small area networks. Matured Classical Short Finding Algorithms are combined with various data structures to get the optimal path within the short time. Some algorithms search the shortest paths by working in two directions, forward and backward. For Large networks especially geospatial networks, heuristic search techniques are popular methods. This does not need to calculate the distance values of all nodes in the whole graph. Moreover, dividing the large tree into small subtrees reduce the processing time by working each subtree separately to find the optimal path. For these reasons, finding the shortest path is still occurring as an issue for the complex networks until now.

There are different transportation types such as bus, car, train, ship and airplane. Among them, people in Yangon, Myanmar, uses private car and public bus transportation as the main transportation types. Yangon is the largest city in Myanmar and also the economic center of country and it owns a lot of popular places that are worth to visit. However, it is composed of many roads and streets. The system provides not only the shortest route but also the possible route for networks using efficient heuristic based technique. The proposed technique is tested using two datasets: Yangon downtown network for undirected graph and Public bus network for directed graph.

1.1 Transportation Network

Transportation network is a network that composed of spatial data. It is also called spatial network, this is applied for vehicle movements about communication that travels from one place to another. A transport network describes as a permanent track such as roads, rails and canals or scheduled services such as public transit bus, train and airline. Two types of transportation services exit in this transportation network. These are tangible network (example roads and rails) and non-tangible networks (example air and sea corridors). Transport network is the opposite of non-spatial networks such as social networks, corporate organizational network and other systems such as biological systems, electronic online management systems, these are especially managed by using the other constrained factors. Transportation networks can be constructed by linking the points to move between different places. Some researchers transform these networks into graph structured or tree structured networks to be easy for finding the optimal path between two places of the network. Transportation network terms to the architecture of routes within the locations described as nodes. A route is a connection using a link between two nodes which contain as a part of the network.

1.2 Public Bus Transportation

Public transportation systems belong as an important role in all sections such as communities, education and businesses. Moreover, constructing the public transportation with low cost to travel from one place to another makes to be easy the life of people. There are many transportation systems such as bus, train and private cars within the local region of a country. Among the transportation systems in Myanmar, public bus transportation system is more important than the other transportations. Most of the people rely on the public bus transportation as the main one for travelling within the Yangon Region. Myanmar is composed of Seven States and Seven Regions. Among them, Yangon is the commercial city after the capital city of Myanmar, Nay Pyi Taw and so it is the most popular transportation city in Myanmar. The system develops android mobile application for public bus network using client server architecture.

1.3 Finding the Possible Transit Routes for Public Bus Transportation System

Public bus network is very complicated because it needs to consider from two starting points if the keyword search is used at finding the shortest route whether the user needs to ride the bus from current location or opposite location. Therefore, it is not easy to find the shortest route for using the multiple source and destination by using current popular shortest path algorithms. Among the algorithms, most find all nodes and traverse all edges to reach the destination. Therefore, when the dataset is very large, most short finding algorithms must wait long time to get the shortest results. Moreover, the geographical system of the earth occurs varies with the time because of the natural disaster. For these reasons, short finding problems are still occurring as an important issue until now.

This proposed system considers applying the heuristic based technique that the execution time is less at finding the shortest route and it does not need to traverse all nodes in graph. This technique considers depend only on the neighbor nodes of current going node to reach the destination. Public bus transportation system provides not only finding the shortest routes but also suggesting the suitable bus numbers for passengers. It is implemented on the mobile device for the passengers to use easily. JSON is used to store the data for supporting the keyword search on the mobile device for reducing the data storage size. Both Myanmar and English Languages can be used for searching the route. Android is used for the client side and J2EE Servlet is applied as the tomcat server.

1.4 Motivation of Thesis

Some people want to know more than one route while they travel from one to another place. When they get all routes information for their travel, they can choose the most suitable route for their current condition. For example, although the route is the second shortest, when they want to pick up some people on this way while they travelling, this route is the best for them in current condition. Moreover, some destination can be reached by taking only one bus, but take a long time. Users can go other alternative bus numbers to be faster. For this reason, the system proposes a heuristic based technique that can search different possible routes simultaneously. Especially, the system develops this technique for public bus transportation network.

The people living in urban area use public bus for shorter route. Public bus services provide buses for passengers to carry them when they travel to their desired places. Trains cannot support to reach many places for passengers in Myanmar. The people mainly rely only on the public bus system for their transportation. In 2017, the government changed the public bus services by modifying the bus network with bus lines. They reduced the number of bus lines from over three hundred to over ninety bus lines. Each line is classified by color coded depending on location served. The people are very difficult to know about the information of public bus services in Yangon. Moreover, some routes for passengers cannot reach directly and they must take the transit routes.

This thesis provides not only two shortest routes between the locations such as start point and end point but also the bus information that include transit routes. The next point is that some people are difficult to use English language. For these people, especially intend for providing the Myanmar Language to be easy when they find the routes. Although the citizens that live in Myanmar can use easily it, foreigners are not familiar to use it. For this problem, the system provides two languages when the passengers find the shortest routes. Another problem is a situation that does not know whether the destination is reached. For this issue, the system will provide an alert system when the destination is nearer for the passengers. For each route, the system will advise the suitable buses with showing the distance.

1.5 Objectives of the Thesis

The public bus transportation system is developed on mobile device for Yangon Region. The major objectives of this thesis are as seven objectives. The first objective is to find the two shortest routes for Yangon Bus system. The second objective is to suggest the possible transit bus numbers for each route after finding the shortest routes. The third one is to support the bus traveler by showing the optimal route in the map on the mobile device.

The fourth objective is to assist the mobility of people by taking advantage of the transportation infrastructure. And then, the next objective is to give the alert messages before the destination is reached for the unfamiliar users about Yangon Region. The sixth objective is to reduce the processing time in finding the optimal route and geospatial information and the last one is to propose the efficient short path finding technique that can search shortest and possible routes simultaneously.

1.6 Contributions of the Thesis

This system develops the public bus transportation services of Yangon Region. Yangon Bus Network is composed of the nodes that contains about 2000 and many connections between them. The information for the Yangon Bus is constructed from Yangon Region Transport Authority (YRTA) and it operated for new bus service. Some data of Yangon bus network is constructed by collecting data using GARMIN-EXTRA-10 device and Google Earth. Popular Classical shortest path algorithm such as Dijkstra cannot find the shortest path in short time when the network is big. Although heuristic search A* algorithm can find the shortest path with the suitable time, it still considers and expands unnecessary while the next node is going to reach the target node. For these problems, the thesis proposes a new algorithm based on heuristic search called Combined Forward and Backward Heuristic Search (CFBHS). This algorithm can find the optimal shortest paths simultaneously during the short time by removing the unnecessary nodes. The next step is finding the possible transit routes and buses using proposed transit method for the routes that emerging from CFBHS algorithm. This thesis not only determines the two optimal paths but also presents transit bus routes between start and destination bus stop.

1.7 Organization of the Thesis

The structure of this thesis contains six chapters forming a general introduction, related work, theory background, implementation and conclusion. This thesis is organized as follows:

Chapter 1 describes the different types of networks and introduces the public bus transportation network for Yangon Region and abstract of the system, objectives, motivation, problems and contributions.

Chapter 2 depicts about the literature reviews and theories that have been implemented in Geographical Information System and public transportation network. This chapter also describes some algorithms and methods applied in short path finding for networks.

Chapter 3 is the discussion about some popular short path finding algorithms among of the algorithms and defined problems about these mature and widespread algorithms.

Chapter 4 proposes a short path finding algorithm and describes in detail about processing of this algorithm and also describes the working of transit methods after finding the routes in public bus network.

Chapter 5 describes the experimental results using empirical and statistical approaches for Yangon Downtown Network by comparing other popular short finding algorithms. This chapter also illustrates the proposed algorithm for Yangon Bus Network.

Chapter 6 concludes the summary of thesis and describes limitation and further extension in the future.

.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORK

Geographical information system (GIS) is a computer system to capture, store and apply the information that includes location on Earth's surface. The location data may be described as address, zip code, spatial data such as latitude and longitude. GIS can support each person and organizations to have the better lives using spatial information. This provides geographical information by analyzing and queries spatial data using GIS tools such as Quantum GIS (QGIS). V. Bhanumurthy [10] applied geographic information system for emergency management analyzing the routes. Geospatial services can be used for addressing emergency problems. Remote sensing such as geospatial technologies and geographical information can be used combined to get valuable real time information when the comprehensive platform is provided for emergency situation. GIS can also provide another service for finding shortest route at disaster situation. E. Lepuschitz [35] renders data set on Google Map by applying the GIS software for disaster management and mitigation. Geographical data such as routes and locations can be analyzed and provided for describing on the map using software tools such as Google Earth and QGIS. Unrelated information can also be stored together spatial data using key index for processing and sharing the completed geospatial information. Data visualization is presented in maps to range sophisticated charts and graphs in remote sensing and GIS.

GIS can be applied to compare and contrast different types of information such as income, population and education level. Moreover, the information about landscape such as vegetation and soils for agriculture and flood detection can be captured in GIS. It is related many applications such as planting, weather detection, disaster management, roads, electric power lines, manipulating and analyzing the data that concerned with locations. Short route finding and traffic congestion in Transportation systems frequently used GIS technology. These routes can be presented on map using GIS featured maps such as Vector Map, Raster Map and Google Map in web applications. Google Map services are used not only for the enterprise systems such as Java 2 Enterprise(J2EE) web server, Active Server Pages (ASP.net) but also for mobile applications based on Android Operating System, Windows Mobile System for GSM and iOS.

2.1 Popular Environments Based on GIS Technology

Today, many countries all over the world try to develop into the smart city for many areas such as education, emergency systems, hospital systems, e-government systems, weather forecasting system, security management systems, route planning systems and disaster management systems. Many application areas relate to use geographic information. Today, the population on earth is more and more increased. And there is also increase a significant about the patients' number for the emergency cases. For increasing the number of patients, sending an ambulance from hospital is not easy. In the serious case, the patient may die at sometimes if the ambulance does not reach in time. S. Sarkar [49] proposes a GPS tracker system that manages about the ambulance of the hospitals. This system can track the ambulances' locations and if it receives a call for serious case, it checks the ambulance that exists the nearest location. Then, it sends a message to the ambulance to go the patient. The system can help the patients to reach early to the hospitals using GIS technology. R. J. Jyothi [31] proposed a traffic control system to avoid the delay because of traffic congestion when they send an ambulance to the patient.

Some environments use GIS to analyze the regions that occur the crime. The place that emerges the incidents marks using the spatial point. The number of incidents may differ for different locations. A. G. Akpan [3] used GIS and crime mapping to be feasible for police departments. P. Mubi [40] adopted Inverse Distance Weighted and Getis-Ord GI* methods and geographic tools such as Google Earth, ArcGIS 10.3, Global Mapper 18.1 and Surfer 14 software in identifying the hotspots for crime incidence such as Child Abuse, Robbery, Homicide, Murder and House Break.

Natural disasters occur heavily in many places all over the world because of the changes of weather condition and surface of earth. Many researchers solve disaster problems until now using GIS technology and Remote Sensing. Flood affects many places of the world and they consider it as a major disaster. M. Arseni [5] tried to emerge a flood hazard map from Landsat five's satellite image data and depicted the land cover types using innovative Semi-automated Classification Processing tool. This research described a speedy and prohibitive method to monitor the flood emergency management plan about the downstream of Prut River Romania.

The advanced system of Geospatial Information System (GIS) is WebGIS that can use for the web platform. GIS Server and client works together to share information between them. WebGIS provides a platform to implement Client Server web application for both mobile application and desktop application. WebGIS helps businesses and organizations to manage all their geographic data. WebGIS is a free open source web GIS system and it contains Geo-server referred web GIS server and PostGIS/PostgreSQL. It has many advantages to do dissemination, distribution, and managing about spatial information for implementing time and cost-effective system over the internet. A. Bhandari [9] generated and mapped the wasteland information system in GIS environment using GIS technology and Remote sensing techniques.

GIS technology can be used in the location-based systems, traffic systems and shortest path finding in the Transportation system. P. Jaramillo-Alvarez [28] proposed a transportation model based on operations research techniques. This paper designed and modeled for the development of public transportation plan and its stages. This also developed to minimize transfer costs and time. A* Algorithm is used as the popular heuristic search algorithm for finding the shortest path in different areas such as road networks and nearby search location [50][52]. E. Dere [16] used the A-Star algorithm for finding the shortest part between start and end point on the Google Map that segmented as grid-cells. To specify the grid for active paths on Google Maps, this paper used the Algorithm Map. Moreover, traffic intensity about the different roads was constructed on the Google Maps.

2.2 Spatial Data and Coordinate System

Spatial Data is related about the geographical information on the earth. Different places on the geographical area can be viewed on the map using spatial data. Different tools can be used to get the spatial data about the required places of the earth. Google Earth is a popular tool to analyze and retrieve the geospatial data. Spatial Data can be marked on the surface of the google earth and can be detected whether these spatial data are correct with the real time data. The required portion of the geographical region can be extracted and transformed as the raster map and vector map using geographical information system's software such as Quantum GIS and ArcGIS. These tools can be used to study the data structure on the map about the earth. Spatial data can be viewed in different shapes such as circles, rectangles, lines and polygons.

Coordinate systems provide geographical data about the locations. It can also represent the locations as geographic features and imagery. Implied with any geographic data is a spatial reference system. The spatial features within geographic reference system are mapped to an earth-based reference system. The center of this area is on earth reference systems which can be created on a Geographic Coordinate System (GCS) or Map Coordinate System and a Project Coordinate System (PCS).

2.2.1 Geographic Coordinate Systems

The places on the curved surface of the earth can be identified as coordinates in geographic coordinate system. This is measuring in angular units from the center of the earth relative to two planes by the equator and by the prime meridian. A position of a place is therefore defined by two values: latitude and longitude. Figure 2.1 describes latitudinal lines and longitudinal lines on the earth's surface. The equator describes in 0° degree reference lines on the left and prime meridian for longitudinal measurements. A GCS is specified by a sphere, ellipsoid, geoid and datum.

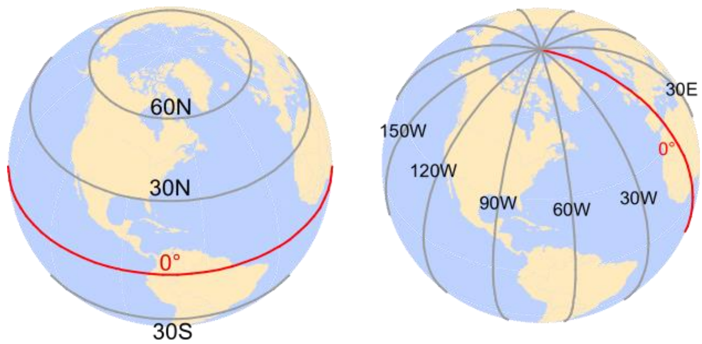


Figure 2. 1 Examples of Latitudinal Lines on the Left and Examples of Longitudinal Lines on the Right

(1) Geodetic Latitude

Latitude of a point can express the angle between equator and a line fleeting over the point perpendicular on earth's surface. Latitude can also refer to the geodetic latitude. Latitudes are specified with north (N) and south (S) of the equator for locations. Positive values for northern hemisphere and negative values for southern hemisphere are used in latitude angles. And North Pole and South Pole of latitude value is 90°N and -90° or 90°S. For example, north latitude 40° means angle of 40° between a line that connects the center of earth and plane of equator. All points having the same latitudinal angle connect as a line on the map is assigned latitudinal line. Figure 2.2 describes lines of latitudes have Y-values that are between -90° and +90° degrees.

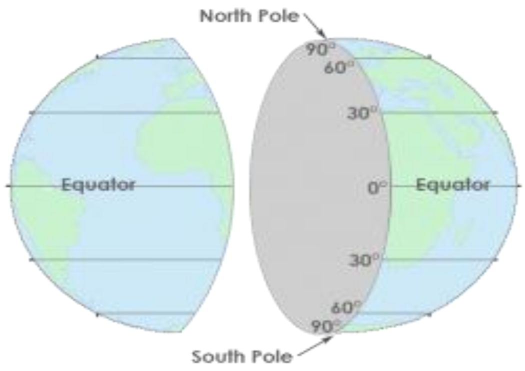


Figure 2. 2 Latitude Coordinate

(2) Geodetic Longitude

The longitudinal point on earth's surface is the angular distance west or east from prime meridian assigned as a reference line occurring from North Pole to South Pole. An arc on the map expressing from north to south joining North Pole to South Pole and over the points with same longitudinal angel is expressed a longitudinal line. Imaginary grid that covers the sphere is formed from lines of latitude and longitude. Geographic coordinates of a point for a location on the earth's surface are comprised by combination of latitude and longitude angles. Figure 2.3 describes lines of longitude which have X-coordinates between -180°and +180° degrees.

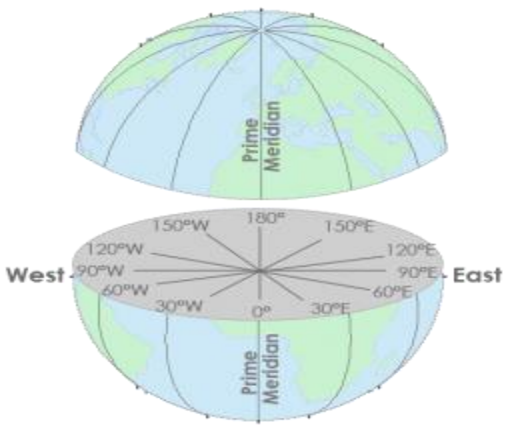


Figure 2. 3 Longitude Coordinate

(3) Geographic Coordinate's Formats

Degrees of latitude and longitude values are subdivided into smaller sectors. Each degree (°) can be divided into sixty minutes of arc (′), and each minute can also be divided into sixty seconds of arc (″). A latitude or longitude coordinate may be written in one of three formats. Degrees, minutes and seconds (ddd° mm′ ss.s″) can be used to describe the times. The range of degree for latitude is from 0° to 90° and 0 to -90° and longitude is from 0° to 180° and 0 to -180°. 1° is equal to 60′, it can carry over and add 1 degree to the part of degree in its place. Similarly, in 60 seconds, in its case it can carry over and add 1′ to the minute part instead. Minutes and seconds can always be assigned less than 60. For example, it can be described as 45° 36′ 23″ N 122° 46′ 18.4″ W. Degree values of latitude and longitude can be denoted as decimal numbers. Positive latitude is assigned in north and negative longitude is west. For example, 46.3678° is north latitude and -122.28453° is negative longitude. The formats of latitude and longitude can be converted defined by the following:

(i) Converting from minutes to degrees : degrees= minutes/60

(ii) Converting from degrees to minutes : minutes=degrees x 60

(iii) Converting seconds to degrees : degrees=seconds /3600

(iv) Converting seconds to minutes : minutes=seconds / 60

(v) Converting minutes to seconds : seconds=minutes x 60

(1) Sphere and Ellipsoid

The earth can be assumed as a sphere for small scale maps to simplify the mathematical calculations. An ellipsoid of earth describes more accurate measurements. An ellipsoid works well when large-scale map is used. An ellipsoid is assigned using semi-major axis or semi-minor axis. Figure 2.4 describes the designation of Semi-minor axis and Semi-major axis of ellipsoid.

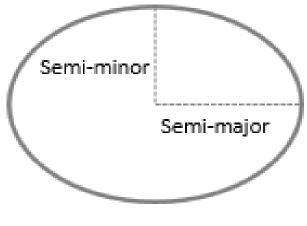


Figure 2. 4 Semi-minor Axis and Semi-major Axis

(2) Geoid

The earth's true shape, the geoid is represented as a mathematical model for a GIS environment. The earth's shape is not a smooth surface and it results undulations from changes by a crossing the surface in gravitational pull. It can influence locational measurements. The visualization of earth's gravitational potential can be focused by envisaging its surface entirely immersed in water and quantifying the distance from the center of earth to the water surface above the entire earth surface. The earth's geoid with gravitational field can be viewed in rainbow colors in the following Figure 2.5.

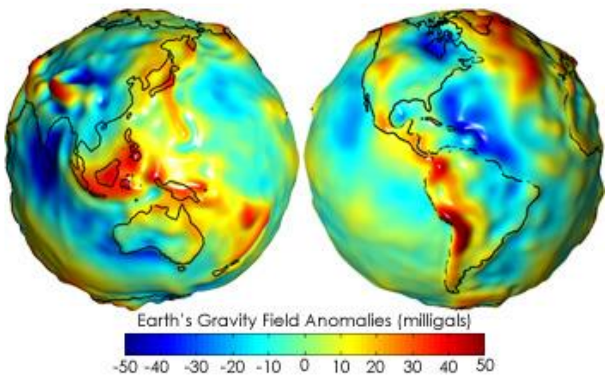


Figure 2. 5 Earth's Geoid with Gravitational Field in Rainbow Colors

(3) Datum

Working with a mathematical model about the earth's shape represents to align the geoid with the sphere or ellipsoid depiction and to record the features of earth's surface onto the sphere. This align process can be resident where the ellipsoid surface is thoroughly fit to the geoid at a point on the surface of earth. Choosing the best one when alignment is made about the ellipsoid to the geoid describes a datum.

(4) Constructing the Geographic Coordinate System

Ellipsoid model is used to define the Geographic Coordinate System (GCS) and it is aligned with the geoid. Moreover, to define the association which GCS is suitable for map document reference system. Overlapping layers can be constructed with different datum and this can produce different coordinate values. Locations with different coordinate values depend on the GCS are described in below.

(i) A location about NAD 1927 GCS with a coordinate values of 44.56698° north and 69.65939° west

(ii) A location about NAD83 GCS with a coordinate values of 44.56704° north and 69.65888° west

(iii) A location about WGS84 GCS with a coordinate value of 44.37465° north and -69.65888° west

If the suitable coordinate system cannot be chosen to define the points, then these coordinate values can be erroneous on a map. Defining temperature depends on different measuring units such as Fahrenheit, degrees Celsius and Kelvin can also produce a different numerical result for mapping on the earth's surface. Figure 2.6 describes two different coordinate systems GCS NAD 1983 on the left and GCS NAD 1927 on the right.

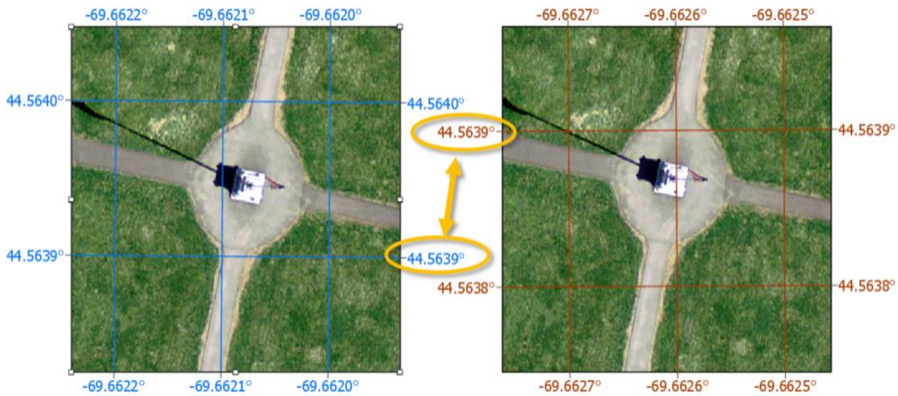


Figure 2. 6 Two Different Geographic Coordinate System

2.3 Global Positioning System

Global positioning system (GPS) is well known as the satellite-based navigation system that built by the U.S. Department of Defense (NAVSTAR) and it is composed of 24 satellites and it works using algorithms and receiver to provide place, speed and time management for sea, air and land travel. And it also contains six earth-centered orbital planes (ECOP) that composed of four satellites for ECOP in the satellite system. The Soviet Union constructed Sputnik I satellite in 1957 for creating more powerful geo-location technology. The U.S. Navy started tracking submarines using satellite navigation which run about the TRANSIT system. GPS based application can be used for getting the more reliable system about the location not only the window operating system but also the mobile operating system starting around the year of 2000. Especially, GPS is used to determine a position, to get the navigation when from one place to another is travelled, to monitor object motion or personal movement, to construct the geographical maps on the earth and to get the specific time measurement of the globe. Figure 2.7 shows the orbit of GPS.

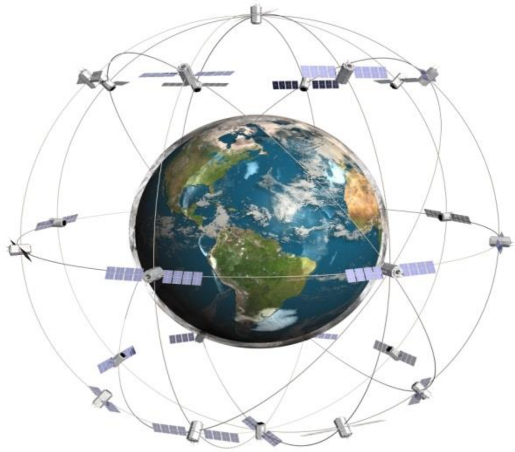


Figure 2. 7 Global Positioning System with Orbit on the Globe

Some research papers solve the weakness of the existing system implementing the automatic detection systems getting data from sensors provided in the vehicle. GPS and GSM module from Global Positioning System (GPS) in the vehicle can send the place that accident occurs automatically to the server. B.V. Sowmya Shree [55] implemented the automatic assigning ambulance from hospitals using GPS and GSM technology. Moreover, this paper controls traffic light signals to minimize the time along the path when the ambulance carries the patients to the hospital using Radio Frequency (RF) communication. GPS can be used in different applications not only the land but also the sea and in the air. However, to receive the signals from GPS are difficult in the deep forest, caves and under the water. Navigation Systems and the systems that need to use remote sensing technology also apply the GPS such as car navigation system, shortest path finding on the road, commercial fishermen and mariners. More modern systems can view the user's position on a Google map and call a drive and he can track the user's path and the system provides the optimal route on a map to go the user's location. To collect the spatial data that provides from Global Positioning System, GPS devices can be used. In addition, GPS tracking devices also provide to track the path of people, vehicles and pets using GIS technology. There are many GPS devices such as Kid smart Watch, Trax Play Kid, Garmin GPS Device and Spot Gen3 Tracker. Figure 2.8 describes some GPS devices in different applications. Figure 2.8(a) serves many functions such as phone book, voice chat, remote shutdown and it also traces for the safety of kids. Parents can control the setting of smart watch. Trax Play Kid's GPS in Figure 2.8(b) can also be used for the safety of children. In addition, GPS tracking device is widely used for tracking the roads. Garmin device in Figure 2.8(c) is evident as a popular GPS tracking device for collecting the spatial data (latitude and longitude) on the surface of the earth. The Spot Gen3 tracker that described in Figure 2.8(d) is a best suitable tracker for hikers because it can resist severe environments and still work with the same stream.

(a) Kid Smartwatch

(b)Trax Play Kid

(c) Garmin GPS Device

(d) Spot Gen3 Tracker

Figure 2. 8 Some GPS Devices in Different Applications

2.4 Google Earth

Google Earth is a popular tool for tracking the geospatial data accessing satellite and aerial image and it represents to a three-dimensional spherical shape as described in Figure 2.9. Google Earth can be used for free on the web for applying the basic geospatial features but it needs to give a cost if the user requests for more advanced features. ESRI shape files can be created and printed clearly using Google Earth Service. There are three versions especially in Google Earth: Free version, Pro version and Enterprise version. For a personal use and schools for educational sites, the services that include in free version are enough. Pro-version is used for importing the ERSI shape files. Earth Enterprise provides the services to build custom 3D spheres and 2D maps. Earth Enterprise does not offer a private version of Google imagery that's currently supported in Google Maps or Earth. Each Google Earth’s versions can be used to access and construct data in KML (Keyhole Markup Language) format. Google Earth can also give the search abilities to pan, zoom, and rotate on the view of the globe. It also supports the tools for creating new data, new marker and layers on Google’s servers. Open Street Map files can also be imported to view the geospatial information in Google Earth.

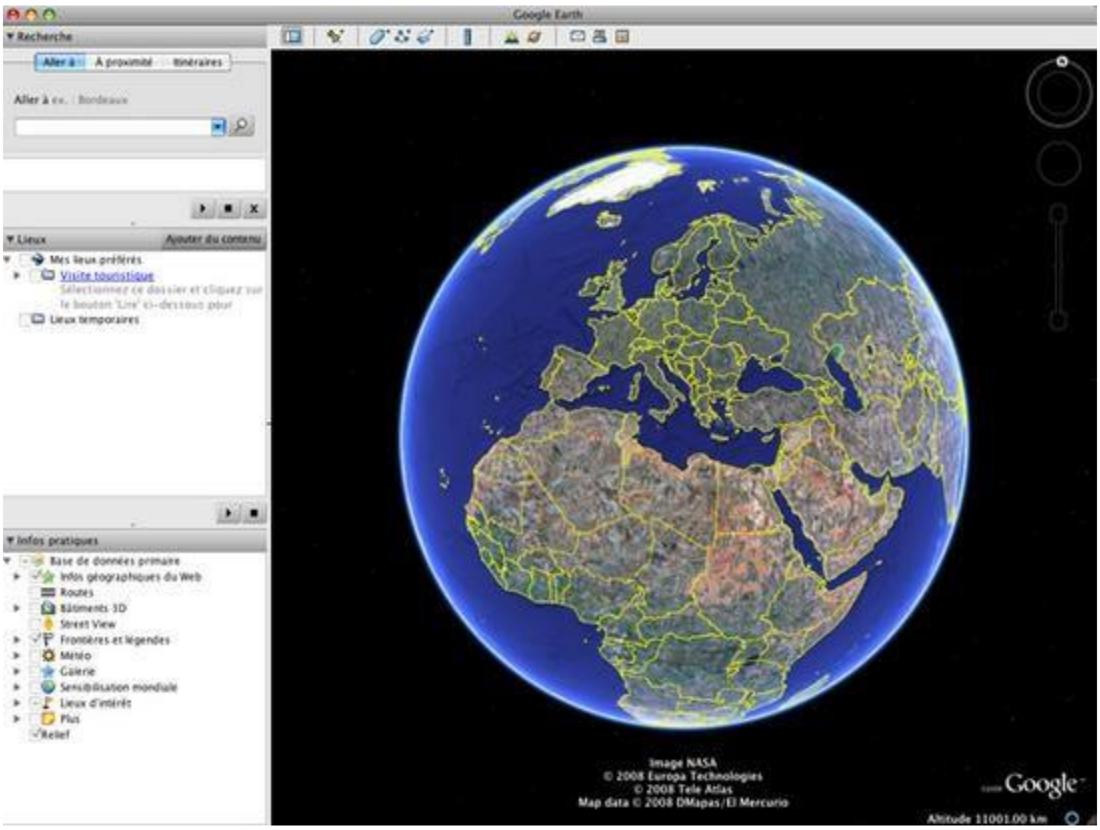


Figure 2. 9 Google Earth

2.4.1 Google Map and Open Street Map

Google Map is a service that produces from Google. It is launched to people in the year of 2005. It includes many features of Application Interface (API) services such as viewing the direction, searching the route and route traffic, giving to customize the map titles design, setting the overlays, and using the advanced layers and so on. Google Maps are especially used in the websites to show the locations of the customers by using the markers to set the latitude and longitude point and the users can see the marker on the map. Users can find their current location and pass this information to the others and display it on the map. Travelers who use the private car and public taxi service can choose the routes with little traffic by viewing traffic conditions on the map. In addition, the markers can be created as the users' desire on the map and they can also draw the route on Google map. Users can view Google map in two different views as shown in Figure 2.10. Map view can be applied to consider the custom routes on the Google map clearly and spatial data analysis in geographical system as delineated in Figure 2.10 (a). Figure 2.10 (b) can give the actual view for the people using satellite view of Google map.

(a)Map View

(b)Satellite View

Figure 2. 10 Two Different Views of Google Map

Today, mobile technology is more and more developing and it is widely used in many places because it provides ubiquitous features. Some people use mobile technology and web service together to get many features that give from web technology. They constructed road showing the respective route on the map because there are many routes exits for different province. It is not easy to collect the length of the road information manually. I. N. Piarsa [43] conducted a land transportation infrastructure using GPS technology and Google map service on smartphone. They discussed a system to act the road data collection that supports the information such as the name, length and the type of the road surface. I. N. Piarsa described the system overview as the following Figure 2.11. When the user requests to construct a road on the map, he or she enters the start address and end address as the user input. Then, the system will collect the required road data especially spatial data (latitude and longitude) using GPS technology. After collecting the data, the system sends these data to the server using web service. The server constructs a road and it uses the Google map service to display this road on the map. Finally, the user sees the requested road data input as the road map as shown in Figure 2.12.

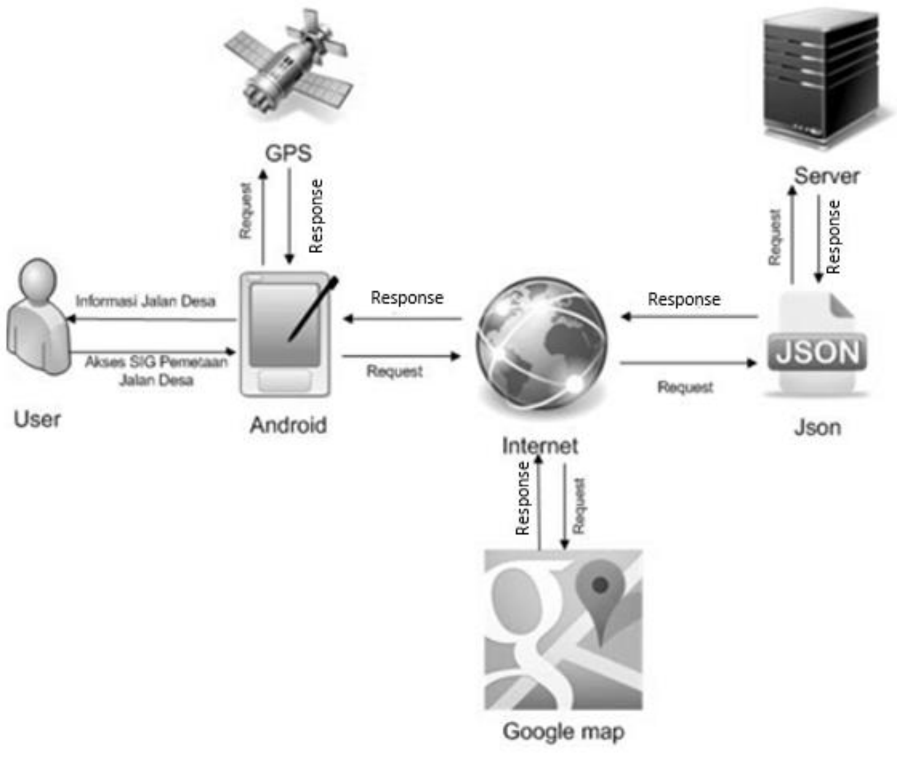


Figure 2. 11 System Overview of Land Transportation Infrastructure

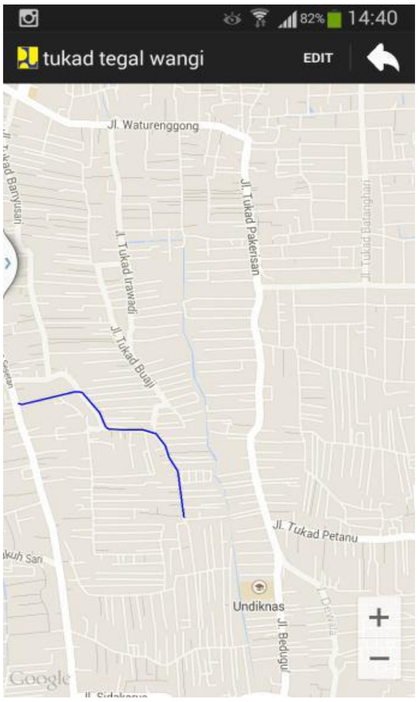


Figure 2. 12 Showing Road on Google Map

Open Street Map (OSM) is an editable map on the surface all over the world with an open-content license. The spatial data and related information about the map can be used by extracting from Open Street Map. It is also referred as the geo-database. The required map portion can be retrieved from Open Street Map using GIS software such as Quantum Gis (QGIS) and Aeronautical Reconnaissance Coverage Geographic Information System (ArcGis). Figure 2.13 shows the Open Street Map for Yangon in Myanmar using Qgis. Points, polygons, rectangles and lines can be used while the expression of spatial data on the open street map. Red circles in Figure 2.13 represent the bus stops in Yangon. The connection of spatial bus data can be analyzed and edited by geospatial analysis. QGIS can be used for big data analytics.

After uploading the OSM map, QGIS tool can support to get the data by automatically assigning the region of the map canvas. The downloaded data is automatically saved in spatial database of QGIS. This database can be used to store not only the geo-spatial data but also the user created attribute data. Furthermore, required spatial data can be filtered from database using QGIS supported functions and SQL language. These filtered set of records can be added as a layer on Google map of QGIS and data analysis can be operated.

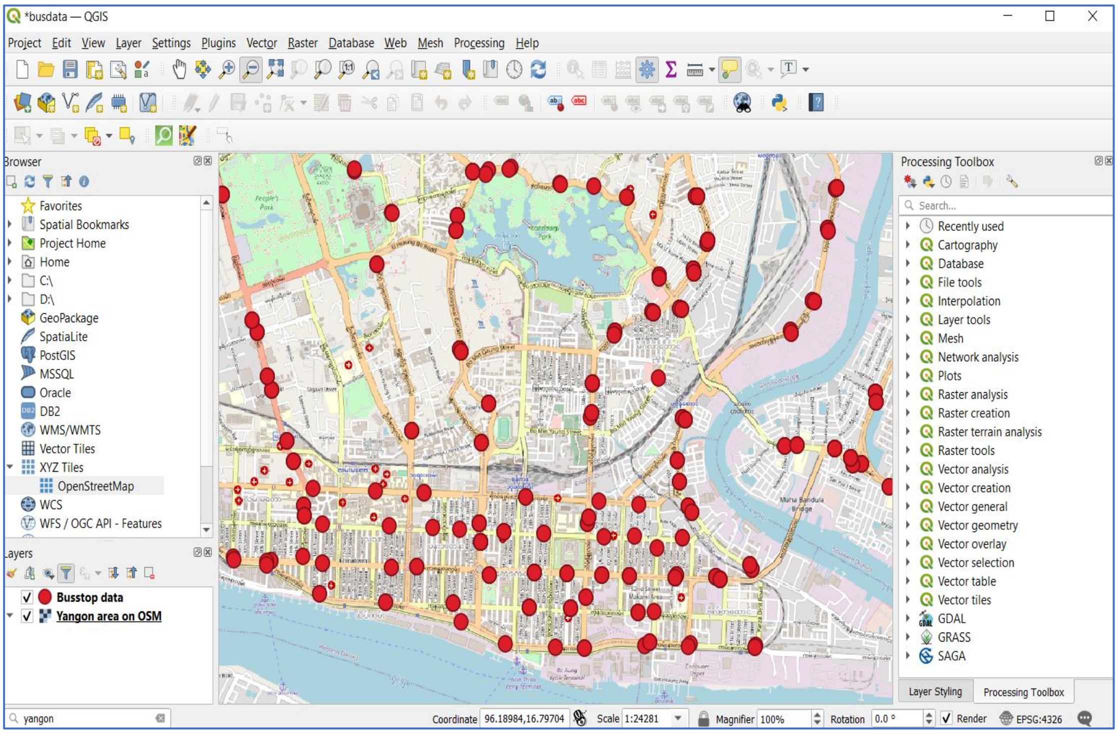


Figure 2. 13 Yangon Area on Open Street Map

S. Ahmed [1] described the network analysis about road network of the Open Street Map for Greater Cairo Area using GIS technology. This paper proposed GIS best route algorithm by road network analysis better than the shortest path algorithm for emergency systems that need to get the fast response time. The base map of Greater Cario was downloaded from Open Street Map (OSM) using web service and it could be accessed and analyzed using ArcGIS tool. The GPS signal receiving becomes important because many applications are more increasing dependency on Ubiquitous and mobile technology. Most route finding system uses to manage and navigate the routes by considering the coverage of cellular network and GPS technology. T. Hultman [27] presented threshold based modified Dijkstra's algorithm to find the shortest path. This paper also applied a route planner tool to search the connectivity aware shortest paths on the crowd sourced data that downloaded from OSM. M. A. Brovelli [11] worked on measuring the quality of OSM buildings. OpenStreetMap (OSM) can provide a lot of geographic information for different parts of the world. Many useful geographic datasets can be used freely without paying the costs. J. Jokar Arsanjani [30] described different popular research about Open Street Map.

2.4.2 Road Network

Road network can be viewed in Google earth to analyze the road network data and to find the desired routes on the Google Map. Road network is the graph structured network. It is composed of the edges and vertices that assign for different signatures according to the purposes. For example, some uses the junction points as the vertices. For public bus network, bus-stops are used as the vertices. Many bus-lines can pass for the same bus-stop. There public bus network is more complex than the road network. Figure 2.14 describes the bus-stops of Yangon City on the surface of the earth.

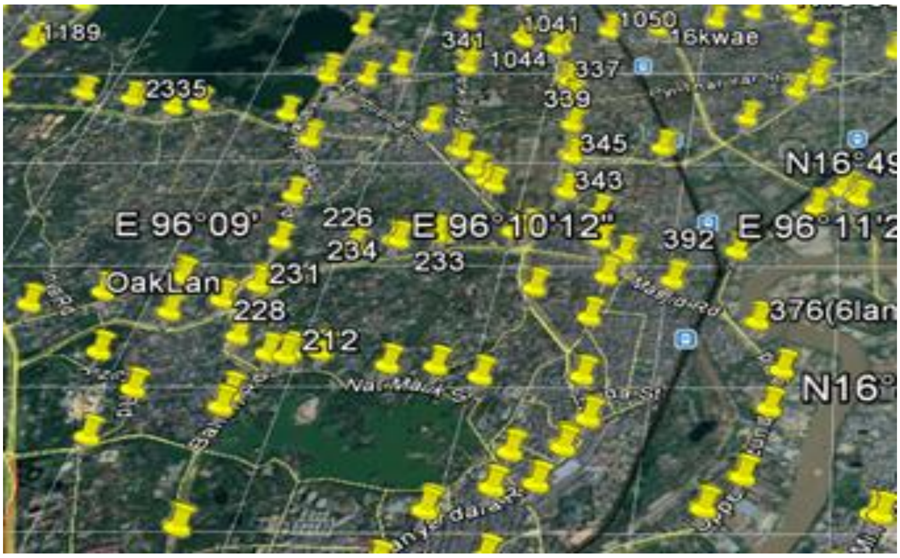
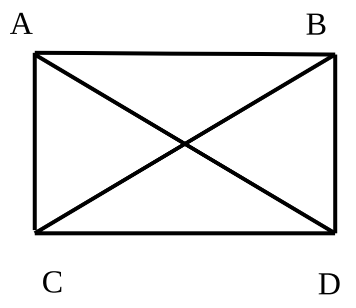


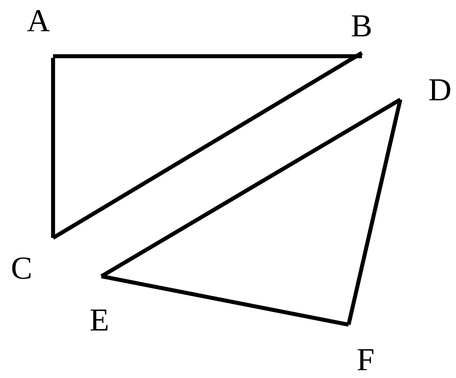
Figure 2. 14 Bus Stops of Yangon City on Google Earth

(1) Graph Network

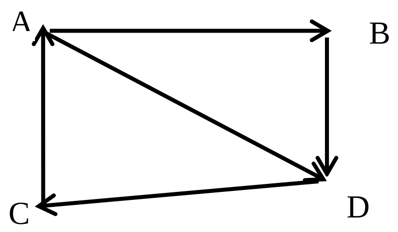
Graph is a network that composed of a set of vertices *V* and a set of edges *E*. Vertex *V* is a set of vertices $\{v_1, v_2, v_3, \ldots, v_n\}$. An edge e_1 is a connection between two vertices that contains in a set $\{v_1, v_2, v_3, \ldots, v_n\}$. A set of edges *E* contains $\{e_1, e_2, e_3, \ldots, e_n\}$. There are different types of graph structure but the graph is especially considered whether it is connected graph or not and directed graph or undirected graph. Figure 2.15 describes the different types of mathematical graphs. Figure 2.15 (a) mentions the connected graph and it can go any vertices from one vertex to another within the graph. Unconnected graph cannot go to all vertices of a graph as described in Figure 2.15 (b). Directed graph need to consider the direction while trying to go another vertex from current vertex. In Figure 2.15 (c), vertex ‘A’ to vertex ‘C’ cannot go directly although the connected line exists because it does not feasible according to direction. Undirected graph can go both directions between two vertices within the graph as expressed in Figure 2.15 (d). For example, if an edge exits from node A to node B, then there also exists an edge from node B to node A.



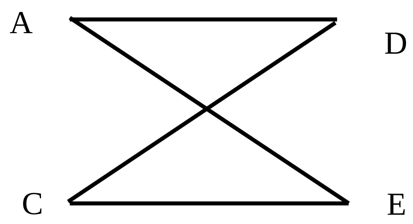
(a) Connected Graph



(b) Disconnected Graph



(c) Directed Graph



(d) Undirected Graph

Figure 2. 15 Different Types of Graph

S. Guze [25] presented theory parameters and algorithms to do analytic and find the optimal result and minimal spanning tree in graphs for transportation system. Traffic congestion of the city streets and roads is one of the most disputed ones in the largest cites over the world. Many technologies become develop to solve the traffic congestion and they try to plan transport network using GIS and Information Technology. They devote the problems the organization of different things of transport flow of an urban transport transit network for interconnection [15]. They proposed an approach that aims for roads and streets network planning and traffic management rules using telecommunications information technology and components of graph theory.

(2) Hypergraph Network

Hypergraph contains vertices and edges as ordinary graph. It is also a type of graph in which an edge can join any number of vertices of a graph. These edges that contain in hypergraph are called hyperedges. An edge or vertices may occur as a subset of another edge in hypergraph. Therefore, hypergraph is a very complex network because a vertex or an edge may occur repeatedly in another edge. In addition, when a route is searched between two locations, it is not easy if the network structure is constituted as the hypergraph. Many researchers firstly prepared the hypergraph network into the simplex network to get the routes in the short time. Others try to construct more powerful shortest finding algorithms rather than modifying the network structure from hypergraph to simple graph. Figure 2.16 describes the network structure of hypergraph that compose of the set of vertices {A, B, C, E, F} and edges {Bus1, Bus2, Bus3}. An edge Bus2 contains as a subset of the edge Bus1.

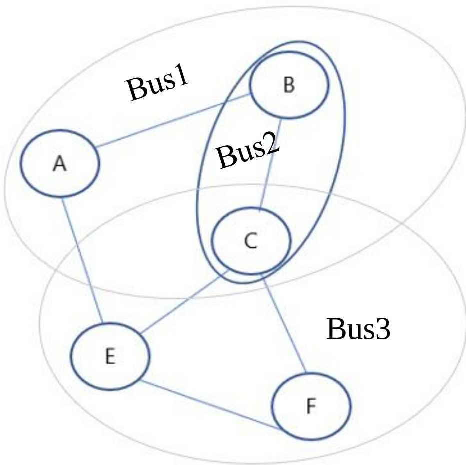


Figure 2. 16 The Network Structure of Hypergraph

The hypergraph can be re-designated into simple graph structure network and each vertex have many edges [24]. Formally, a hypergraph H is a pair H= (X, E) where X is a set of elements called nodes or vertices, and E is a set of non-empty subsets of X called hyperedges or edges. This thesis uses the Yangon bus network dataset that constitutes with 84 bus lines, it contains many vertices and edges. In Figure 2.17, hypergraph network is built with four bus lines, 301 vertices and many incoming edges and outgoing edges and this graph is described using SocNetV.

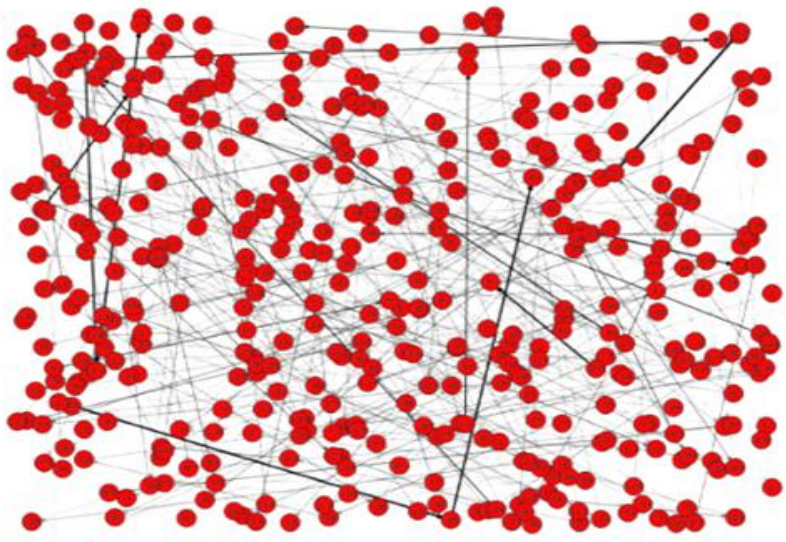
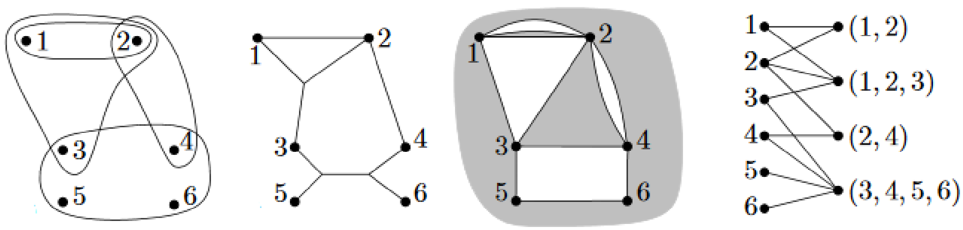


Figure 2. 17 Hypergraph Bus Network

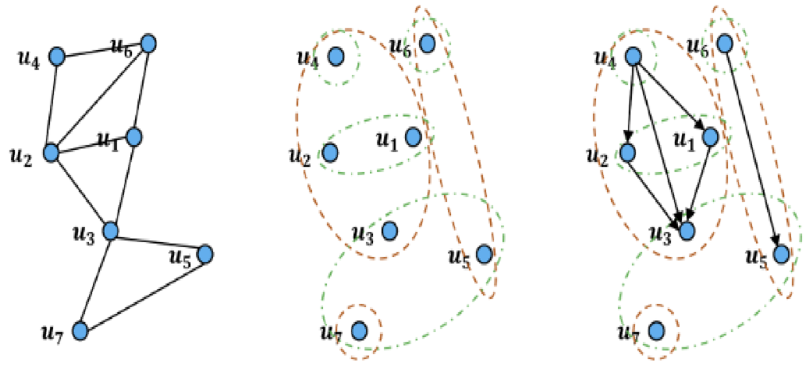
R. van Bevern [53] discussed different subdivision drawings of hypergraphs. Formally, the standard of constructing the hypergraph does not describe in detail but it is used in many application areas for circuit visualization, computational biology and social networks. Figure 2.18 describes four drawings about hypergraph. Among these, subset based and edge-based subdivisions are the most common methods to draw hypergraph. R. van Bevern also discussed the subdivision schemes which are more general than concrete Euler diagrams and more constrained than vertex-based Venn diagrams. F. Feng [20] proposed a novel partial-order hypergraph to denote the partial-order relations among vertices due to the ordering relation of data is lost if conventional hypergraph is used.



(a)Subset-based drawing | (b) Edge-based drawing | (c) Zykov representation | (d) Incidence Representation (bipartite graph)

Figure 2. 18 Four Drawings of the Hypergraph H= (V,E) with V = {1,2,3,4,5,6} and E = { (1,2), (1,2,3), (2,4), (3,4,5,6)}

Uni	City	Sal
u_1	c_1	s_2
u_2	c_1	s_2
u_3	c_1	s_1
u_4	c_1	s_4
u_5	c_2	s_1
u_6	c_2	s_3
u_7	c_3	s_1



(a) Data (b) Simple Graph (c) Hypergraph (d) Partial-Order Hypergraph

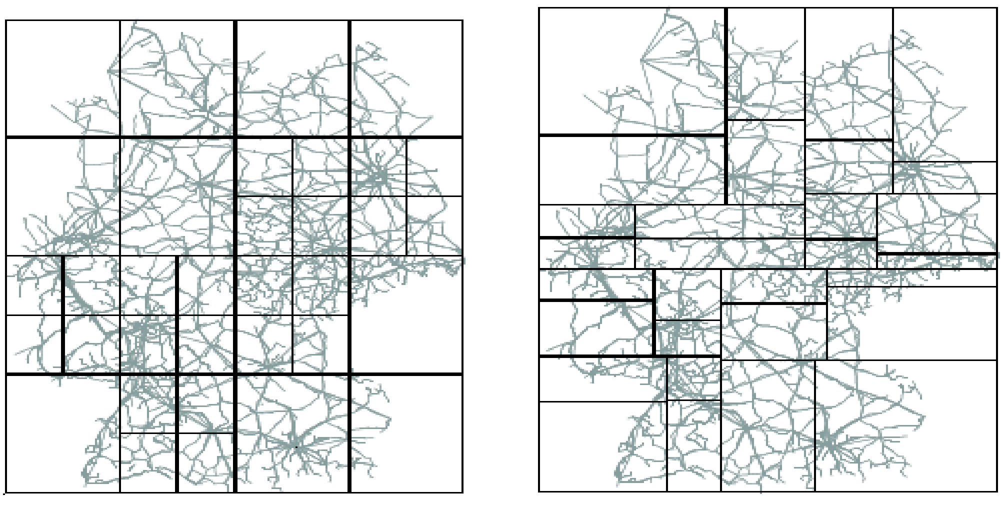
Figure 2. 19 The Relationship between Simple Graph, Hypergraph and Partial-Order Hypergraph

In Figure 2.19(a), each row in data table denotes a university, city and salary. The highest number of salary level is highest index number and the order of salary's level is s_1, s_2, s_3, s_4 according to priority. Universities (u_1, u_2, u_3, u_4) exist in the same city c_1 and it is also the connected subgraph due for this situation. Figure 2.19(b) describes a simple graph, that connected two nearest vertices into an edge. Figure 2.19(c) depicts the hypergraph, where each edge is constructed depend on the same city or same salary. Because of the partial ordering method, Figure 2.19(d) considers the order between the edges depends on salary. For example, university 'u4' will go to the university (u1, u2, u3).

2.5 Partitioning Techniques

Graph network is used in many areas such as social pathological, biological and transport networks. Most graph data contain a lot of vertices and edges and the network structure is complex. If the resulted area requires only a portion of the graph, to analyze the whole graph is not required as essential. Retrieving the required part from the graph, for doing to analyze and find the shortest result is easier and it helps to increase the speed for network analysis. H. Bast [7] describes different partitioning techniques to get the fast response for finding the shortest path route and these partitioning algorithms are compared for various network to decide which one is the most suitable for current layout. H. Bast depicts mainly popular graph's partitioning techniques, Quadtree and KD tree.

To apply the graph partitioning techniques, firstly the graph is constructed into grid structure and it contains the points for this plane. Figure 2.20 (a) describes the quadtree. Quadtrees are especially applied for computing the geographical data in image processing and geographic information systems. After splitting the raster map by row by row and column by column, each node can be divided exactly as four internal nodes. The structure of internal node may be rectangle, circle or different shapes. In quadtree, a region cannot be divided depend on the desired range because a region can be repeatedly divided into only four equally sizes sub-regions. For this problem, KD Tree in Figure 2.20 (b) can be used and the map can be divided depend on data. The construction of KD Tree is nearly equal to the quadtree. It is also divided recursively on the map into sub-maps and started by dividing two parts on it with drawings two parallel straight lines about the vertical and horizontal axis. For example, let 'C' is a country, 'R' is a region and 'T' is a town. A country contains seven regions and each region is composed of ten towns. If KD Tree firstly will divide the map until the seven parts are reached by start dividing from two lines. This can store the partition data according to the desired data points or desired region and town. Figure 2.20 describes the partitioning of Germany's raster map by M. Rolf H using quadtree and KD Tree. When the shortest path is searched using graph partitioning techniques, the searching time is fast but the preprocessing time is long for training the map data to do partitioning of map layout.



(a) Quadtree (b) KD Tree

Figure 2. 20 Graph Partitioning for Germany Map

Quadtree and KD Tree maintains the fragment map layout into tree structure after graph partitioning is made. However, some researchers use the map data instead of the graph structure network into tree structure network for finding the shortest path because of taking long time in graph partitioning technique at the preprocessing step. Instead of drawing the grid on the map layout, the graph network is constructed into the tree structure network. Divide and Conquer technique is used in some popular research area at finding the shortest path during the short time. In divide and conquer, each segment of a network is split from large network into many small sub-networks. Then, the local optimal result is searched from each sub-network and finally all sub-networks' results are re-combined to get the global optimal result. A. Krause [33] described details in bounded-treewidth graphs, how to define subclasses about it and how to decompose the trees into subtrees. A graph network can be converted into tree network intending to decompose and traverse it. For example, the graph that depicts in Figure 2.21 contains the vertices {A, B, C, D}. let the start node is 'A' in graph, this node will be root node in tree and the tree can be divided into three parts {AD}, {ABCD}, {ABD}.

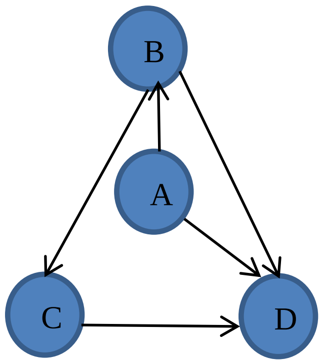


Figure 2. 21 Directed Graph

There are three important parts in tree such as edge, path, depth and level. Fundamentally, an edge is a connected line between two vertices, or a node and a leaf. A path begins from a vertex and ends at another vertex. A path is a collection of sequence of edges along the path. The height of a vertex is the number of edges on the longest path while it is going to downward between a vertex and a leaf. The depth of a vertex is the number of edges from this vertex to the tree's root node. The level is similar finding the depth of a vertex in tree but it needs to add one from the depth. R. Zhou [61] used a proposed algorithm combining the strengths of breadth-first search.

U. Lauther [34] proposed a new algorithm to get the shortest paths with a fast speed in graphs for road networks with nodes that use latitude and longitude data. The method is especially focused on pre-processing step, road map is divided into small segments as regions and it determines the region that will contain the shortest path by calculating edge flags. Dijkstra algorithm is used to find the shortest path for network that contains 500000 road segments.

2.6 Public Transportation System and Network Optimization

P. Jaramillo-Alvarez [28] showed the optimal public transportation routes' findings for urban using operations research techniques. This paper examined the development of transportation planning, its stages and models. P. Jaramillo-Alvarez proposed optimal public bus transportation's model that minimizes transfer than the previous models. However, it suggested using genetic algorithms for large networks in the future. Two types of spatial data: raster data and vector data exit in geospatial information system (GIS). Vector data is made up of coordinates about latitude and longitude instead of pixels that use in raster map. Mostly, vector map is used to find the shortest route because it provides the points, lines and polygons features for map. S. Behzadi [8] used the genetic algorithm to find the optimal route on the urban map with raster data. It is more feasible than using the other algorithms to find the route on raster map. In it, color attributes are used as the parameters for genetic algorithm and each road is depicted by a specific color. A. Jeyaraj [29] analyzed the geographical network in searching the optimal route, finding the nearest place and providing the service area. Searching optimal route and nearby places is a popular service in geospatial information system (GIS). This paper proposed a method based on genetic algorithm to satisfy the optimized route in global sampling. It did not use the genetic algorithm individually due to the low rate in optimal route searching time. It compared the results of three short finding algorithms with and without genetic algorithm. According to results, A* based on genetic algorithm get better performance in searching time than the other optimal route-finding algorithms. Some research used forward changing and backward changing approach to reach the target of an application or to find the route in geospatial information system. Forward changing is finding the result or route from facts. Backward changing tries to get the route by tracing the goal node until the start node is reached. A. Al-Ajlan [4] compared useful area and algorithms that relate the basic idea of forward changing and backward changing strategies and he also described the Breadth First search, Depth First search, Best-First Search, Uniform Cost Search, Hill Climbing Search and A*. Breadth First Search is firstly expanding the same level and then it searches whether the goal node contains in this level. If it does not see the goal node in current level, it will go to next level. Although the breadth search can occur least depth cost, it will be many path costs when the tree is huge. Uniform cost search is similar to the working of breadth-first search but it less the path expand cost because it stops expanding the nodes of current level when the target node is found. Hill Climbing Search is a kind of DFS and always moves towards the goal. Hill Climbing Search is similar to the depth-first search but it has constraints when the end node is reached, it cannot go back to backward traversing. A* algorithm uses the two concepts of both breadth first search and depth first search.

Nowadays, population is more and more increasing and the transport network is more complicated and larger than the previous time. Therefore, many people are difficult to find a route for an unfamiliar place easily. A. Dabhade [14] presented how to analyze the spatial data to get the shortest path for hospital information systems using popular classical shortest path algorithm called Dijkstra and ArcGIS software. S. Sadhu [47] analyzed the network based on ArcGIS software for different management spatial systems about finding the effective travel information, travel bearings and searching the nearest governmental office, schools and hospitals. The spatial information that gets from ArcGIS is used for finding the shortest route between the railway station in Jaipur city and popular places. For examination the experimental results, Google earth image is used and then it does geo-referencing and digitization to get the shape document.

R. S. Parmar [41] aimed to find the shortest route solving traffic congestion problem and focuses to reduce the redundant nodes of the graph network by considering the next neighboring node of a bounding box continuously. Two prototypes are made for it, on both the desktop and mobile. E. Lepuschitz [36] proposed an intelligent web-based route planning application using advanced data structures, heuristics short finding algorithms and network optimization techniques for emergency response system such as natural disasters, health care systems that intend for efficient service delivery. Network optimization also contains as the important sector for solving the shortest path problems. A. Garcia [21] apparatus an intelligent routing system of local tourist organization to support recommendation, route generation and personalized route customization about the local attraction places for travelers. Intelligent routing system implements a time dependent Dijkstra's shortest path algorithm for solving the problems in transportation system. It also applies an advanced OR algorithm to generate public transit routes. Q. Wu [59] used the k-shortest path algorithms to find three shortest paths in Nottingham City and suggest the most suitable route according to three user's preferences. This can reduce traffic congestion and the most effective way in public transportation for private vehicle drivers due to manage the routes according to users' preferences. C. N. Yahia [60] tried to solve traffic assignment problems using decomposition approach (DSTAP) for large and complex networks. This paper examined two different partitioning methods: agglomerative and weighted spectral clustering algorithms. The first one is to decompose a network intending to minimize sub networks' boundary nodes. The next one is to partition the network using normalized graph Laplacian.

The construction of network architecture is important for getting the fast response time in short path finding and operational costs are greatly depended on it. V. Guihaire [23] depicted design and scheduling of the network for transit planning. It also described a classification way for 69 approaches that solved design and transit routes and it analyzed for each work. M. Petrelli [42] proposed new heuristic bus design model to solve the network design's problem and to generate the transit routes. Moreover, it found the optimal sub-set of routes using genetic algorithm. The system produces the bus routes, the frequencies and the vehicle sizes. R. van Nes [56] used the features of an optimization model and describes the details about it to design the public transport network. The optimization model for transport network is implemented on desktop computer.

Our proposed system uses the Public Yangon Bus Network. The main difference between road network and bus network is that road network passes only one time for each node, it means there is only one edge between two vertices. However, there are one or more edges between two vertices in bus network. Figure 2.22 described the nature of Yangon Bus Network that is composed of many edges between two vertices because many bus lines may pass a bus-stop. Each bus line was depicted using different colors to see evidently on the map.

Figure 2. 22 Yangon Bus Network

2.7 Location Based Services in Android

Nowadays, many people use smart phone due to ubiquitous instead of carrying the laptop when they go from one place to another. Android smartphone can give many services such as animations and transitions, audio and video, images and graphics, sensors, connectivity, navigation and user location and so on. Many services including GPS can be provided on android phone's users easily by adding a little configuration in manifest file or app's configuration file from developers. Location based services in android need to get the help of Google Play Services. Location awareness of an android app can give the facilities with geocoding, automated location tracking and activity recognition. A location object of geographic system signifies a latitude, longitude, time stamp, altitude and velocity. There are important methods about location object that support on Smart Phone using Google API key. The altitude value in meters above sea level can be searched using "getAltitude" method. The latitude value and longitude value in degree can be searched using "getLatitude" and "getLongitude" methods. In addition, the approximate distances in meters between two locations: current location and given parameterized location can be searched using "distanceTo" method.

Moreover, android platform can give the service to get current location of a mobile user by creating a locationClient object as the location object. This method can give the latitude and longitude value with coordinates of user. The address can be known from the location object. To use the methods of Google play service, Google

Map API key is needed to use and register via Google Cloud Platform Console as shown in Figure 2.23.

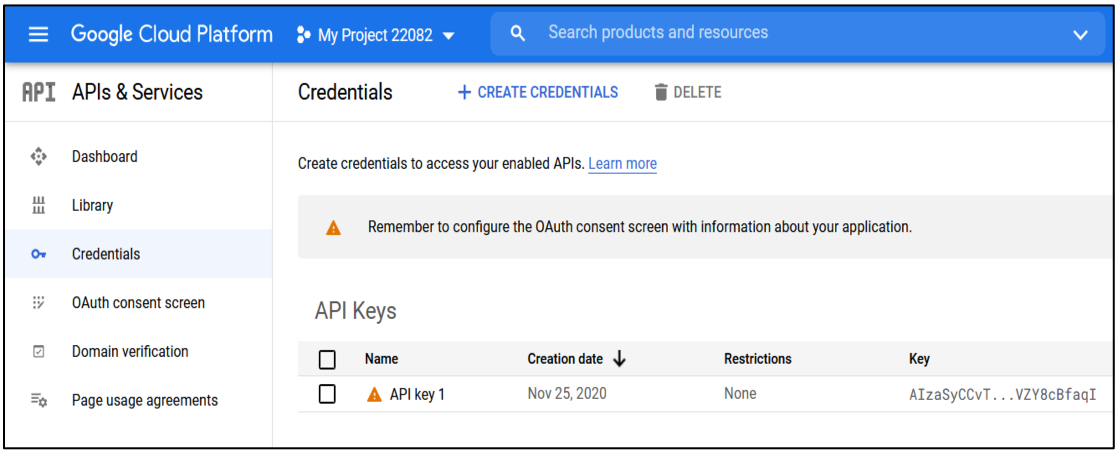


Figure 2. 23 Generating Google Map API Key

The next important one to draw customized routes and points on Google Map is adding the constraints in Android Manifest file. The Android Manifest contains automatically when the android application is created using Android Software Development Toolkit (SDK). The required permission can be assigned using <uses-permission> tag under <permissions> tag. The permissions that need to assign in Android Manifest File for GPS service are six permissions. The first permission: ACCESS_COARSE LOCATION is used to get the approximate location that equivalent to city block of android mobile device based on network location sources such as Wi-Fi and cell towers. The second permission: ACCESS_FINE LOCATION is assigned in Manifest File to know the precise location that support from location providers with GPS as well as Wi-Fi and cell data. The next permission: ACCESS_LOCATION_EXTRA_COMMANDS can allow the application about other operation of the GPS that does not include in ACCESS_COARSE LOCATION and ACCESS_FINE LOCATION.

The fourth permission: ACCESS_MOCK_LOCATION is used to get the location services in developer mode. And then, the fifth permission: CONTROL_LOCATION_UPDATES can be used to get the notification of updated information when the device location is changed. The last permission: INTERNET is needed to assign when the internet service is used. The tag structure in Android Manifest File can be seen in Figure 2.24.

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.example.snazzyapp">

    <uses-permission android:name="android.permission.INTERNET"/>
    <!-- other permissions go here -->

    <application ...>
        ...
    </application>
</manifest>
```

Figure 2. 24 Android Manifest File

M. Singhal [51] proposed a location based model using Google Web Services and Walk Score Transit Application Interface (API) on Android Smart Phones. This model advised clients for current traffic conditions, finding routing information and providing them find nearby hotels with the help of GPS in phones and Location based Services. M. P. Uddin [54] implemented a location tracking system for a group of people on Android mobile phone due to cost effective and it can be used multidimensional purposes that contain many built-in services about GPS technology. Md. Palash Uddin developed a location tracking system with an alert using technological tools such as Jason like xml file for data saving, Java Programming Language, Android Virtual Device (ADV) and LAMP. LAMP stands for the first letters of Linux operating System is ‘L’, Apache Web Server, MySQL database and web programming language or script such as PHP, Perl, Python. J. Samaul [48] developed a monitoring or tracking system based on Android mobile with a GPS system to know the current location and previous location at each specified interval. The system can trace back the previous location for the current tracked route using Google Maps API. A. Pinandito [44] proposed a framework of mobile application for outdoor mobile navigation application at finding different routes. It uses Abstract Factory design pattern, Dijkstra and A-Star algorithms for implementing the system and the routes are represented on a digital map. M. S. Minal Mahure [38] implemented a system using Location based Services on android mobile phone for remind the place that user want to visit. Mobile phone supports the reminder services depend on the time. D. Ankita [17] intended to help the people who trapped in the area that occur disaster by monitoring the location on mobile phones. S. Vanjire [57] used client and server technology for location-based information system. This technology is evident in information technology field. Client gave all required information about location from Android mobile phone to Java 2 Enterprise Server (J2EE). The server executes information carrying from android mobile client and finally passes the result to android. This system advises mobile users about current traffic conditions, routing information and it also supports nearby services for the users. It presents the interaction between the mobile device and the processing steps of Location base service as the below Figure 2.25. In the first step, user requests a service that obtained from GPS using mobile device. This service request is sent to the service server via the mobile communication network in step 2. And then it is sent from service server to geographic database and the required information is send back to server in step 3 and 4. Finally, the requested information is sent back to mobile user via the mobile communication network. This paper also describes the architecture of GPS system and the components of Location based Services.

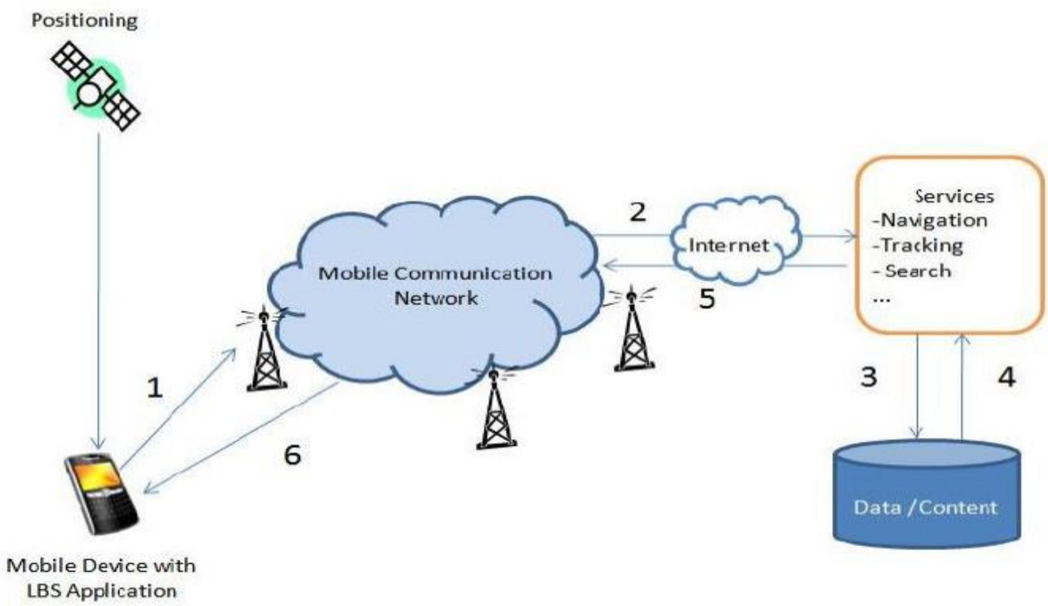


Figure 2. 25 Interaction between the Mobile Device and the Processing Steps of Location Base Service

2.8 Shortest Path Problems

Short finding problems are an important issue until now because the transportation network becomes more and more emerge and complex in Nowadays. Researchers use popular classical short finding algorithm such as Dijkstra Algorithm.

A. Dabhade [14] used the Dijkstra to find the hospital information because people don't know the detailed information about the places within the hospital. However, Dijkstra Algorithm is suitable when the network size is small. Otherwise, if it is so massive, processing time will be very long. Therefore, Dijkstra is used together with different data structures such as priority queues, fibonacci heap and fusion, etc. E. W. Dijkstra [18] described the detail about the time complexity cost as $O(n^2+m)$ for linear search, where the number of vertices and edges are assigned as n and m respectively. B. V. Cherkassky [13] examined using various data structure together with Dijkstra Algorithm and compared to get the more efficient short finding algorithms. R. K. Ahuja [2] proposed an efficient short finding algorithm that get the less response time for the desired route and it costs the complexity to O (m + nlog C), where C describes for the cost of weightiest edge. Y. Huang [26] introduces a short finding algorithm by assigning the constraints to omit the useless paths and to reduce the transverse time while finding the short route associated with relative direction. And, the weighted value is flexibly changed to get the adaptable complexity for different networks. Although the researcher solves the short finding problems using various data structures and constraints depend on Dijkstra's algorithm, it also still needs to get better efficiency.

Another technique is to execute Dijkstra Algorithm in two directions, forward from the source and backward from the destination simultaneously. This bi-directional Dijkstra takes the complexity in O(m) for the worst case, where 'm' represents the number of edges of a network graph. Bi-directional Dijkstra is more efficient than the previous classical Dijkstra because it does not need to traverse the whole graph to find the shortest route. Ramadiani [46] produced the shortest path between two or more positions on a digital map of Android Mobile using a proposed method based on Floyd-Warshall algorithm. Floyd-Warshall algorithm does not know the sequence of nodes for the shortest path although it produces the shortest distance between the nodes. C.Kinyua [32] studied the effectiveness of PRIMs algorithm for the LAN networks design of Chuka University and this algorithm is applied for the design of the Campus Network of Chuka. Prim's algorithm finds minimum cost spanning tree and it is similar with the shortest path algorithm. Although it is simple for finding the shortest path, the processing time is slow when the graph contains larger number of nodes.

Graph data is partitioned and popular heuristic short finding algorithms are applied at finding the optimal route with less time complexity by reducing unnecessary search area [53]. KD tree is one of the popular graphs partitioning technique in data mining and it is a well-known partitioning approach of geographic data [58]. R. H. Möhring [39] Graph partitioning approach achieves the optimal partitions which easy to approximate while finding the goal. Although it occurs at preprocessing stage, it needs to consider the execution time which dominates the shortest-route computing.

The popular heuristic search A* algorithm is used for different environments [6][12][37]. At each stage of the tree or graph, it needs to examine which partial path is suitable to expand to reach the goal. This algorithm runs based on the estimated costs that calculate using the weighted value of different paths while it is going to the goal node. A* choses the optimal path that minimizes using f(n) = g(n) + h(n), where h(n) represents the cheapest heuristic value that go from current node 'n' to the goal node and g(n) is the weighted value from start node to node 'n'. Goyal et al. A. Goyal [22] compared the processing time between heuristic search A* and classical approach Dijkstra. And then this research proved that A* owns better performance than the linear search Dijkstra by comparing the time complexity. But, A* remains nodes' considering problems in the open lists repetitively.

2.9 Summary

This chapter describes three main review collections: The first review is about Spatial Data and Coordinate System, Location based Services and Global Positioning System of GIS Technology. This described how to collect spatial data from Google Earth and GPS devices and the appliance in many areas. It also explains Google Map API, Open Street Map and their works and the relation between local based services and android manifest file. In GIS applications, the collection and data analysis of spatial data is important for getting the accurate data about earth.

The next one is studying the transportation network: Road Network and Bus Network. Moreover, this describes the network structure such as simple graph network and hypergraph network. When the network is complex, to find the required information such as shortest path from network graph is not easy in the short time. This chapter discusses the shortest path problems and their performance. The choice of method is the mainly important factor to get the shortest results. This chapter describes many shortest path algorithms. These are Genetic Algorithm, Classical Mature Shortest

Path Algorithms, Heuristic Search Algorithms, Forward Changing and Backward Changing Strategies, Breadth First search, Depth First search, Best-First Search, Uniform Cost Search, Hill Climbing Search and so on.

Different types of partitioning techniques are described to reduce the searching time when the shortest path is searched. These are Quadtree and KD tree algorithms. Some researches applied Divide and Conquer technique to divide the large tree into small subtrees. This chapter describes the appliance and collection of spatial data from google earth and GPS devices, shortest path problems and their time complexities, graph partitioning techniques, location-based services and network analysis using QGIS and SocNetV. And theorical methods of background of thesis will be discussed in next chapter.

CHAPTER 3

THEORY BACKGROUND OF OPTIMAL ROUTE FINDING

Public transportation relates to all sectors for daily life of people such as education, business and markets and so on. Myanmar is composed of Fifteen Divisions. These are Ayeyarwady, Bago, Chin, Kachin, Kayah, Kayin, Magway, Mandalay, Mon, Naypyidaw, Rakhine, Sagaing, Shan, Tanintharyi and Yangon. Among them, it is evident that Yangon is the transportation city in Myanmar although the area is not so large. Although there are many public transportation services such as train, taxi and buses in Yangon, most of the people use public bus transportation services especially because there are many bus-lines and buses to travel easily from one place to another. However, passengers are difficult to decide which one is the most suitable bus if they are unfamiliar about Yangon Region and public bus dataset is enormous. Sometimes, passengers need to take the bus more than one time and they need to transfer from one bus stop to another while travelling to the target bus stop. The system proposes the optimal and possible transit routes between two bus stops.

3.1 Finding the Optimal Path on the Road Network

Road Network can be seen in two types of graph, undirected and directed. The vehicles can pass both in two directions for some routes. Undirected Concept can be used for this case because undirected path does not need to consider for direction. In directed graphs, graph network needs considering the directions. The spatial data about locations are important to exact, complete and to know the relations between the location points. Most of the research construct the road network into the graph structure and if they need the graph network is transformed into tree structure network and some make partitioning the graph to be easy at finding the optimal path or shortest path of the road network. There are many popular short finding algorithms for finding the optimal route on the map.

3.2 Shortest Path Algorithms

There are many optimal path finding algorithms for dataset that contains vertices and edges. Matured classical short finding algorithm such as Dijkstra and heuristic search algorithm A* are the popular methods to get the optimal route during the short time. Although there are many short path finding algorithms, short finding problems occur until now. Therefore, researchers use some short finding algorithms with bi-directional concept. M. Dramski [19] searched the shortest path by working in two directions. This algorithm works from start to end and end to start simultaneously and finally they meet at a point by going forward and backward.

3.2.1 A * Algorithm

A* algorithm is one of the popular short path-finding and graph traversals algorithms and it is evident as a smart algorithm among of the other classical algorithms. A* uses heuristic estimated values while traversing the graph to reach the goal with high efficiency. It expands the partial path at each step until the goal is reached. A* choose the partial path that exits minimize value as the followings

$$\mathrm{f}(n)= g(n)+h(n) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad (3.1)$$

In Equation (3.1), n is the current last node on the path, (n) is the cost value of the path from the start node to current node n and $h(n)$ is a heuristic value from n to the target. A* can avoid the expensive paths while it is trying to reach the goal using estimated values. It is a good performance and high accuracy one among of the shortest path algorithms. However, it needs to pre-store the estimated values between two nodes in graph using distance-based methods such as Euclidean distance and Haversine distance method. Heuristic Search A* calculates the straight-line distance (heuristic values) between each node and others of a graph. Pre-processing time may be long when the graph is large because it needs to calculate heuristic values for all nodes in a graph. OPEN table and CLOSED table are used in heuristic search A* algorithm. OPEN table is used to store the extended nodes. To store the selected partial paths, CLOSED table is used.

In Figure 3.1, the shortest path will be found between node A and node B. Before finding the shortest path, firstly straight-line distance table is needed to construct as the preprocessing step as described in Table 3.1. This heuristic values between each node and others in a graph are calculated according to the heuristic methods. Table 3.1 describes the straight line distance values between node B and other nodes in the weighted graph as mentioned in Figure 3.1.

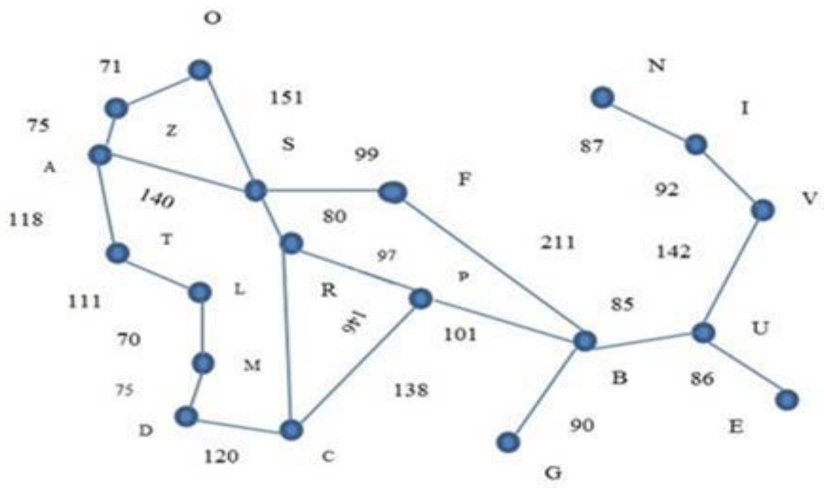


Figure 3. 1 A Weighted Graph

Table 3.1 Straight Line Distance Table

Node	Distance
A	366
B	0
C	160
D	242
E	161
F	176
G	77
H	151
I	226
L	244
M	241
N	234
O	380
P	10
R	193
S	253
T	329
U	80
V	199
Z	374

When node A is expanded, node S, T and Z will be stretched. Among them, node S will be selected because it is the nearest node to reach the target according to the Equation (3.1). The nodes in closed table are described in blue color and the nodes that put in open table can be seen in white color as described in Figure 3.2.

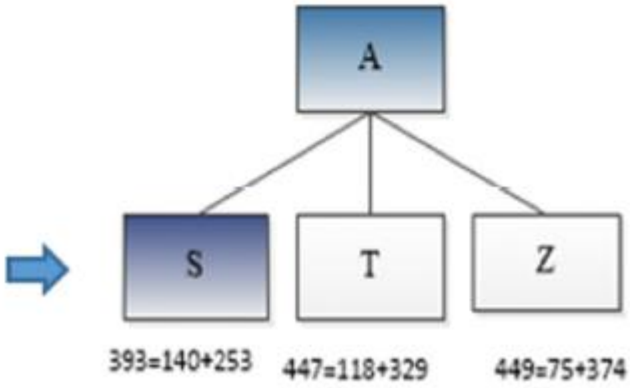


Figure 3. 2 The Heuristic Values between Nodes

Then, node S will expand node F, O and R for next step. In current, the choosing path is node A and node S. These two nodes will exist in the closed table. Node T, Z, F, O and R is in the open table now. Therefore, A* repeatedly calculate nodes in the Open table. Node T and Z will be calculated and compared again for considering the next step and described in Figure 3.3.

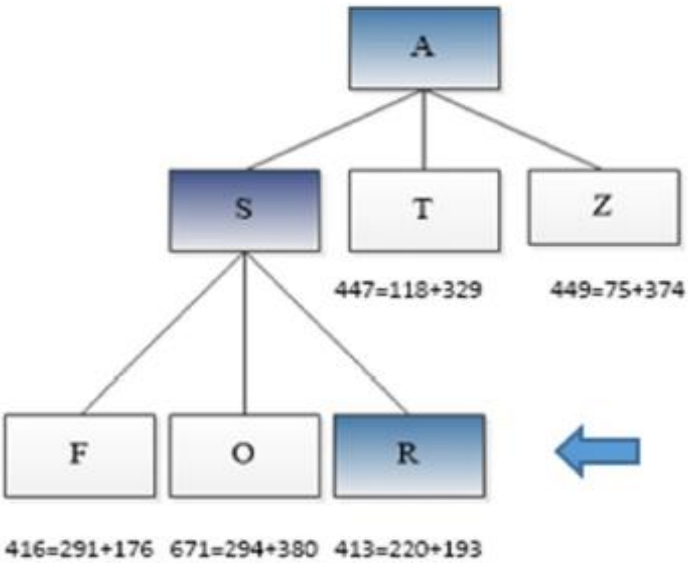


Figure 3. 3 Comparing Heuristic Distance Values among the Nodes

Combined Forward and Backward Heuristic Search and proposed heuristic search algorithms inspired about open node repeatedly considering problems mentioned above.

A* Search algorithm

```
1: create an opened list that contains start node
2: create an empty closed list
3: while (current node != destination node) do
4:    choose the node with the lowest f score in the open list for the next level
5:     if this node reaches destination node then
6:          print the shortest path
7:     else
8:          insert this node into the closed list and find all neighbor nodes of current node
9:          find the f score values of these all neighbor nodes
10:         compare the f score values among these neighbor nodes of current node and
            nodes that already exists in the open list (it does not consider the nodes that
            already saves in the closed list although these exists in open list).
11:         choose the node with lowest f score value and save this node in the closed list
12:         put all remaining neighbor nodes in the open list
13:       go to step 5
14:    end if
15: end while
```

3.2.1.1 Heuristic Function

When the shortest route is found, the performance is an important factor for giving the information during the short time. To satisfy this purpose, shortcut method can be used. Heuristic Method can solve the problem and advice the shortest route to the user faster than classical shortest path algorithms. When the branch is expanded at each step, it considers the distance information and decides which branch is more suitable to follow. Three main types of heuristic functions are (1) Manhattan Distance (2) Diagonal Distance and (3) Euclidean Distance.

(1) Manhattan distance

Finding the distance between two points of a grid that composed of horizontal lines and vertical lines can be applied by using Manhattan distance. It is a popular method among of the distance measured methods on a grid. Manhattan distance calculates the diagonal distance by using the Pythagorean Theorem. The Manhattan distance between point1(x1, y1) and point2 (x2, y2) is given by |x1 - x2| + |y1 - y2|. Distance (d) of a path can be calculated using an absolute sum of the difference between the Cartesian coordinates of a grid using the following Equation (3.2).

$$d = \sum_{i=1}^{n} |xi - yi| \quad \text{………………………….} \quad (3.2)$$

where, n is the number of variables, x_i and y_i are the variables for x and y vectors and the points may be (x_1, x_2, x_3,…) for x and (y_1, y_2, y_3,…). Figure 3.4 describes the Manhattan distance by adding the difference between the points of x and y vectors such as (x_1 — y_1) + (x_2 — y_2) + (x_3 — y_3) + … + (x_i — y_i).

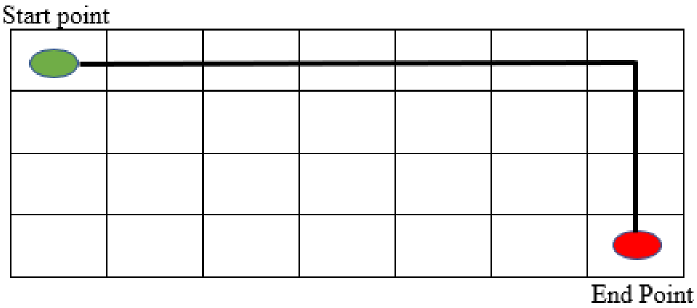


Figure 3. 4 Manhattan Distance

(2) Diagonal Distance

If the map uses diagonal distance that includes one of the heuristic methods, it uses Manhattan distance for four east and four north and this will be 8 * D. The heuristic value should be 4*D2 when the movement is four north and four east. D2 represents the movement cost diagonally. However, for simply move (4 northeast) instead, so the heuristic should be 4**D2*, where *D2* is the cost of moving diagonally. Figure 3.5 shows the example of Diagonal distance.

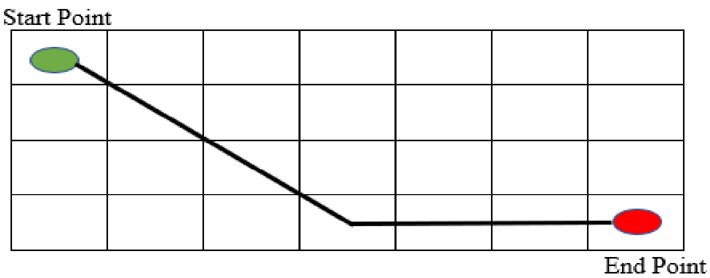


Figure 3. 5 Diagonal Distance

Function heuristic (node)

$$h(n) = c_d\, d\_min + c_n\, (d\_max - d\_min) \quad \ldots\ldots\ldots.. \quad (3.3)$$

where, $d\_max = max(|n.x - goal.x|,|n.y - goal.y|)$ and $d\_min = min(|n.x - goal.x|,|n.y - goal.y|)$. c_n is the cost of non-diagonal movement, c_d is the cost of diagonal movement, n.x is the x value and *n.y* is the y value of current node.

(3) Euclidean Distance

When the heuristic value is calculated, straight line distance function referred to Euclidean Distance can also be used. It can be used when the movement occurs at any angle instead of grid direction. It can also be considered as a less expensive function. Euclidean Distance function is described as follows:

Function heuristic (node)

$$h(n) = \sqrt{(n.x - goal.x)^2 + (n.y - goal.y)^2} \quad \ldots\ldots\ldots... \quad (3.4)$$

where, n.x and n.y is the value of x and y of current node. The example of Euclidean distance is described in Figure 3.6.

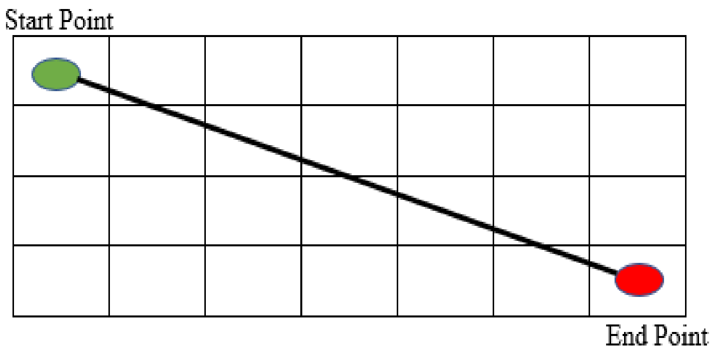


Figure 3. 6 Euclidean Distance

3.2.2 Dijkstra's Algorithm

Dijkstra algorithm is a popular classical approach and it can be used for solving the single source short path finding problem. This algorithm is considered by Edsger Dijkstra who was a Dutch computer scientist in 1956 and this was published in 1959 as a graph search algorithm for finding the single-source shortest path with non-negative edge path costs. Dijkstra algorithm can produce a shortest path tree. Moreover, it can be used for both undirected and directed graphs but graph must be connected. The time complexity for the worst case is *O (E + V log V)*. *E* is the number of edges and *V* is the total number of vertices.

The steps of the Dijkstra's Algorithms are described as follows.

step (1) Assign to every node a tentative distance value: set it to zero for initial node and to infinity for all other nodes.

step (2) Keep a set of visited nodes. This set starts with just the initial node.

step (3) For the current node, consider all of its unvisited neighbors of the current node, mark the current node as visited and remove it from the unvisited set.

step (4) If the destination node has been marked visited, the algorithm has finished.

step (5) Set the unvisited node marked with the smallest tentative distance as the next "current node" and go back to step 3.

3.2.2.1 Working Example of Dijkstra's Algorithm

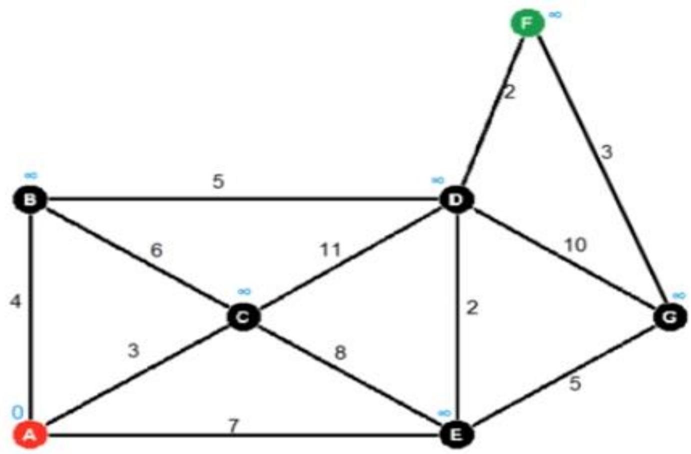


Figure 3. 7 Example of Dijkstra Algorithm

Firstly**,** Dijkstra Algorithm assigns the infinity value for all nodes that contains in graph. In Figure 3.7, there are seven nodes such as A, B, C, D, E, F, G. Starting node is called initial node and let node 'A' be the starting node and node 'F' be the end node. After assigning the infinity value for all nodes, this algorithm will be starting consider the neighbor nodes of start node 'A'. Then, the value of the node B, C and E will become '4, 3, 7' instead of infinity value because these values are smaller than the infinity value. The visited nodes are marked in red color. Among of the values of node B, C and E, node C owns the smallest distance value. Therefore, node C is chosen and it is checked whether this node is equal to the target node. If the target node does not reach in current, this node C expands their neighbor nodes. Then, it is marked as the red color because it is current visited node. Node C owns four neighbor nodes such as A, B, D and E. Node ‘A’ is already visited node and it does not consider repeat again. For Node ‘B’, the value of path ‘A, C, B’ is 9 and it is greater than current distance value ‘4’ for node ‘B’. For this situation new value ‘9’ is skipped. ‘A, C, E’ is ’11’ and it is greater than the previous value ‘7’ and then, it is also skipped. And, ‘A, C, D’ is ‘14’ and this value is assigned for node ‘D’ because it is less than infinity value.

Now, node A and node C will be represented using the red color because it exits in visited phenomenon. In the current situation, among the values of the nodes that are not visited, node ‘B’ owns the smallest distance. Therefore, node ‘B’ is chosen as another node for traversing. The neighbor nodes of node ‘B’ are ‘A, C and D’. Node ‘A and C’ is already checked. Therefore node ‘D’ is calculated to get the distance value for path ‘A, B, C’ and the value is ‘9’ and it is chosen because this distance value is smaller than the current distance value ‘14’. Now, Node ‘B’ is inserted into the visited node. Then, Node ‘A, B, C’ will exist in the visited array. The remaining node ‘D, E, F, G’, have ‘9, 7, ∞, ∞’. Among of these nodes, node ‘E’ is the smallest minimum distance value and this node is selected to expand the neighbor nodes except the visited nodes. The path ‘E, D’ is ‘9’ and this new value ‘9’ is assigned instead of ‘14’. Node ‘G’ is assigned the value ‘5’ instead of ‘∞’. For the next time, node ‘D’ is selected as the current node because it owns the smallest distance values in current situations among of the remaining nodes. In this way, it will do for traversing the following graph in Figure 3.8 until it reaches the target node ‘F’. Finally, the shortest path between two nodes ‘A’ and ‘F’ will produce as ‘A, B, C, D and F’.

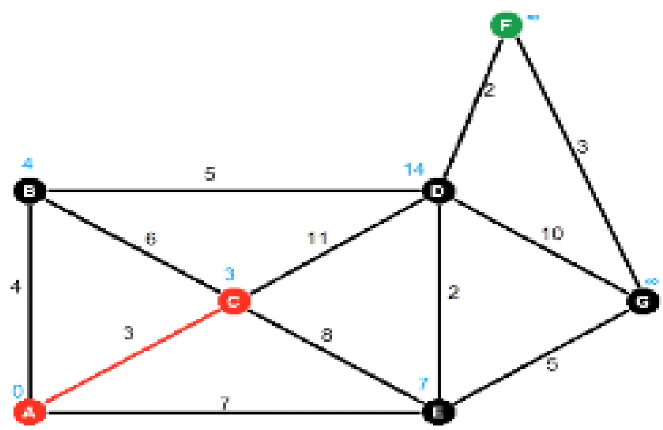


Figure 3. 8 Choosing the Smallest Distance Value using Dijkstra

3.2.3 Bidirectional Dijkstra Algorithm

Bidirectional search finds the shortest path using forward and backward search simultaneously based on Dijkstra Algorithm. While the source node is going to the target node in forward direction, the target node is traversing to the source node in backward direction, and they meet at a same point and finally this path is advised as the shortest path between two points in the graph. M. A. Qureshi [45] described the basic idea of working about the Bi-directional Dijkstra. In Figure 3.9, starting point and target points are represented as the symbols such as ‘s’ and ‘t’. Starting point and target point extends their edges to meet a middle point while they travelling simultaneously in both forward and backward directions.

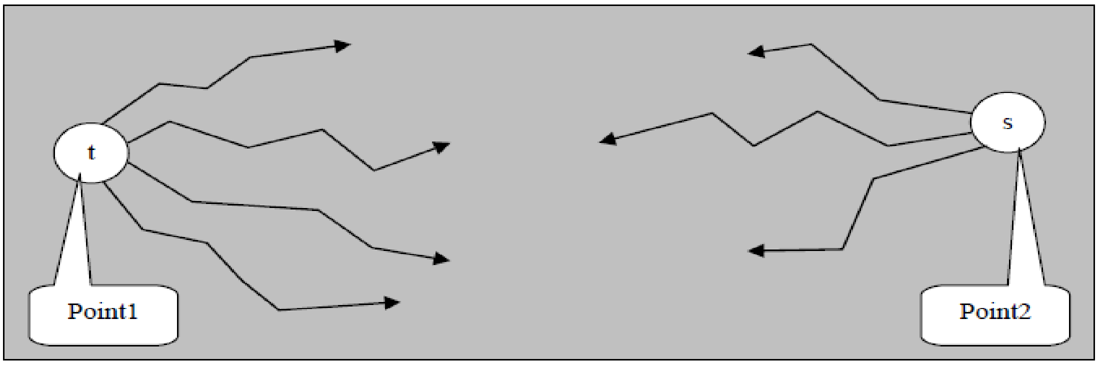


Figure 3. 9 Bi-directional Dijkstra

Finally, a point that comes from forward direction and another point that traverses from backward direction can meet as shown in Figure 3.10. While travelling, each point checks the conditions of other sides. For example, ‘Point1’ checks the conditions of ‘Point2’ and Point2 also checks the current information of ‘Point1’. When they meet at a same point, they exchange their travelling information and this information are combined as a single route. And then, this result is saved in routing database and this information is displayed to the customers.

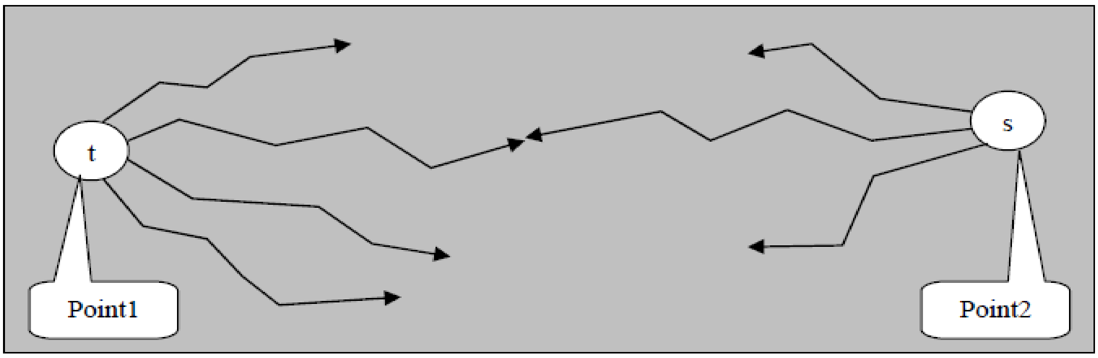


Figure 3. 10 Meeting at the Same Point

3.2.4 Problems of Related Shortest Path Algorithms

There are many shortest path finding algorithms and the working of some popular short finding algorithms have already mentioned above. Matured classical short path finding algorithm called Dijkstra is the popular approach at finding the optimal path but it needs to traverse the whole graph and it is more suitable for small network or graph that constitutes small spatial dataset. When the network is large, it needs to wait a long time for getting the shortest path between starting node and goal node. Some researches execute Dijkstra Algorithm bi-directionally. The time complexity of Bidirectional Dijkstra is less than the popular Dijkstra because of the working in both directions. However, it still needs to traverse all vertices in the whole graph. Heuristic search algorithms can be used at finding the optimal path and it does not require traversing all nodes in the graph by using estimated heuristic methods. Heuristic search A* is faster than the classical short finding algorithms if they compare the time complexity among of the algorithms. A* algorithm uses open table and closed table. Closed table is used to save the nodes that will include in result list. The neighbor nodes and all nodes in the open table are considered at choosing the node for the next level. Sometimes, it considers the nodes in the Open table repeatedly and it can enhance the computation time while the algorithm executes the graph for getting the shortest result. Moreover, A* algorithm needs to give the time for preprocessing time to save the estimated heuristic values between each node and other nodes in a graph. The proposed system will inspire an algorithm to reduce the time complexity by avoiding the computation time of unnecessary node. This can solve comparing the values of the nodes repeatedly in the open list. In addition, the system will give two shortest paths in the short time and it is faster than popular heuristic method A* algorithm and Bi-directional Dijkstra.

3.3 Summary

This chapter describes two main parts: the detailed work of shortest path algorithms and the comparisons among these algorithms. This chapter describes heuristic search algorithm: A star algorithm. It uses the heuristic method to estimate the shortest result in the short time. This chapter also presents different heuristic methods: Manhattan Distance, Diagonal Distance and Euclidean Distance. A star algorithm does not need to traverse the whole graph to get the shortest route. It uses open table and close table at finding the shortest route. Moreover, this chapter describes classical mature shortest path algorithm: Dijkstra algorithm and it is suitable for small network. It traverses the whole graph to get the shortest result and the shortest result is accurate but it is not suitable for large network because it can take more longer time to generate the optimal result. The next one is using bi-directional search in classical shortest path algorithm. This algorithm is faster than simple classical shortest path algorithm: Dijkstra.

The second part is comparing the work of these three algorithms: A* Algorithm, Classical Shortest Path Dijkstra Algorithm and Bi-directional Dijkstra. Although Bi-directional Dijkstra is faster than Dijkstra, it can still traverse all edges of the whole graph. A star algorithm is faster than other two algorithms: Dijkstra and Bi-directional Dijkstra. However, A* algorithm is still considering the unnecessary nodes when the current node is expanded for going to next level. For these problems, next chapter will present step by step working of the proposed algorithm together with working example.

CHAPTER 4

IMPLEMENTATION OF THE PUBLIC BUS SYSTEM

To develop a smart city, the transportation is very important because it is related with all important sectors such as health, education, travelling and especially in business. It also saves costs and provides better service about safety and security management to people. Smart city transport applies wireless and Internet technologies to provide transport information between two points for large city and control the traffic flow from city's authorities. Today, internet technology has more and more developed and many systems are implemented on the mobile phone due to the ubiquitous. For a smart city, the construction of good transport infrastructure is the main step among of the ways. There are many transportation services such as private cars, taxi, trains, motorcycles, bicycles and public buses. Among them, most of the people who live in developed country especially rely on the public bus system.

In this system, public bus transportation for Yangon Region is developed using proposed short finding algorithm based on subtree dividing scheme. There are mainly three parts: the first part is finding two shortest paths; the second part contains searching transit nodes from the shortest path route using transit nodes calculating method; and the third part displays these routes on the Google Map of Yangon Region. As the other services, the proposed system can set the timer for the desired location for unfamiliar users about the Yangon Region. In addition, the customer can study the bus lines and detailed information about bus management system easily. To be efficient at finding the shortest paths, the customers can use the keyword search technique for filling the information about starting bus stop and destination bus stop. Two languages (Myanmar, English) can be used for not only Myanmar citizens but also foreigners.

Figure 4.1 describes the architecture of the proposed system. The system uses client server architecture at finding the optimal paths for public bus transportation network. This architecture mainly contains two roles. The first one is working by using android mobile phone from client side. When the user wants to know about the shortest path, he or she inputs the source and destination bus-stops via mobile phone and then these data are passed to the server side using hypertext transfer protocol. When the data that pass from the client side reaches to the server side, Java 2 Enterprise (J2EE) Tomcat server accepts the data and calculate two shortest paths and finally it sends the result as JSON format to the client side. Before calculating the shortest paths of user's inputs, the system firstly constructs the graph database by transforming the Yangon Bus Network. After sending the result, the user can see the paths on the map of android phone and it can also view the detailed bus's information that include transit bus-stop.

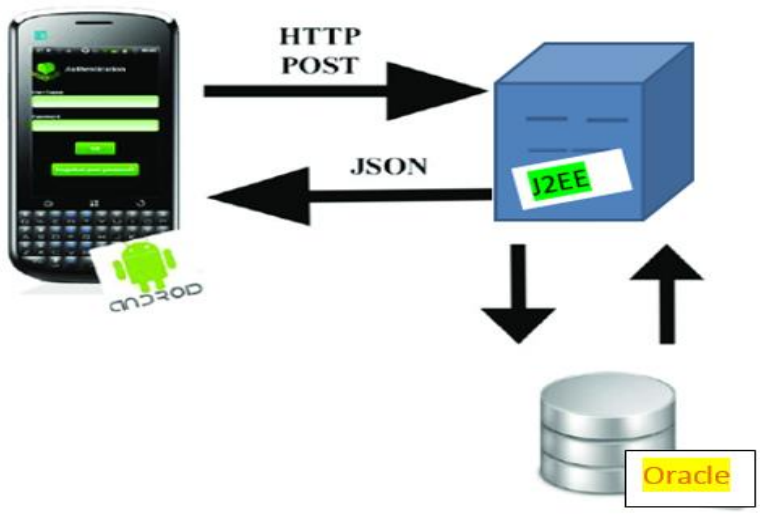


Figure 4. 1 Architecture of the Proposed System

The proposed system especially contains four main parts: constructing the graph database at the pre-processing time; giving the current bus stop and destination bus stop; calculating the two optimal paths; finding transit bus stops, displaying each optimal path on the google map; and setting the alarm for the targeted bus stop. When the transit routes are described on the google map, users can see the information about each bus stop of these routes by clicking bus stop icons on google map. When two optimal paths are searched, the system uses Proposed Combined Forward and Backward Heuristic Search (CFBHS) Algorithm. The detailed working of CFBHS algorithm will be described in 4.3 of current section.

4.1 Preprocessing Stage for the Proposed System

The system contains two parts at the preprocessing stage. The first part is the collection of latitude and longitude data, and the second part is the construction of the graph database using spatial data and distance calculation method. This system collects these latitude and longitude data into three ways: using Google Earth, applying GPS

Etrex-10 handheld device, and getting the spatial data from the government that operates under The Yangon Region Transport Authority (YRTA) of Yangon Region, Myanmar.

Firstly, the latitude and longitude data of each bus stop from Google Earth is converted to decimal value. Google Earth gives the value of each bus stop as degrees, minutes and seconds format. After converting these spatial data to decimal value, the distance value between two bus stops can be computed using Haversine distance formula. It takes longer time than Euclidean distance formula but it is more accurate for spherical shape. Euclidean distance is more suitable for the grid view of a map. Equation (4.1) describes how to convert degrees, minutes and seconds to decimal values. The relation among of the degree, minute and seconds can be described in the following Table 4.1. One degree is equivalent to 60 minutes or 3600 seconds. One minute can be described as 1/60 degrees and one second is equal to 1/3600 degrees.

Table 4.1 The Relations between Degree, Minute and Seconds

One degree	60 minutes or 3600 seconds
One minute	1/60 degrees
One second	1/3600 degrees

The decimal degree from degrees, minutes and seconds is equal to

$$\text{Decimal Degrees} = \text{Degrees} + \text{minutes}/60 + \text{seconds}/3600 \quad \ldots\ldots\ldots.. \quad (4.1)$$

The next one is using GPS Etrex-10 handheld device. This device is also applied while the data is collecting. GPS receiver can provide information about the locations, tracks, waypoints and map. This device can run more than 24 hours per charge. It needs a view of sky to mark the locations using device but it can even work for the dense forest.

4.1.1 Yangon Region and Yangon Bus Network

Yangon is the popular city related with transportation in Myanmar. Many people use public transport for everyday life. However, there are many traffic and accidents occur at the Public Bus system. The government, therefore, reduced many bus lines from over 300 to over 100 at January, 2017. Therefore, traffic problems may lessen due to the reduced number of bus lines. However, there are many transit bus-lines than the previous bus system. The proposed system will solve these problems. Yangon Bus Network is served under The Yangon Region Transport Authority (YRTA) that operates transit agency for Yangon Region, Myanmar. Yangon Region is composed of four districts that constitute in 33 townships and Yangon Bus Network is run over the Yangon Region. It can be deliberated as a graph with positive weighted values. The node represents a connection and each edge of the graph network is linked with a road fragment between two connections. Figure 4.2 describes 33 townships that include in four districts of Yangon Region.

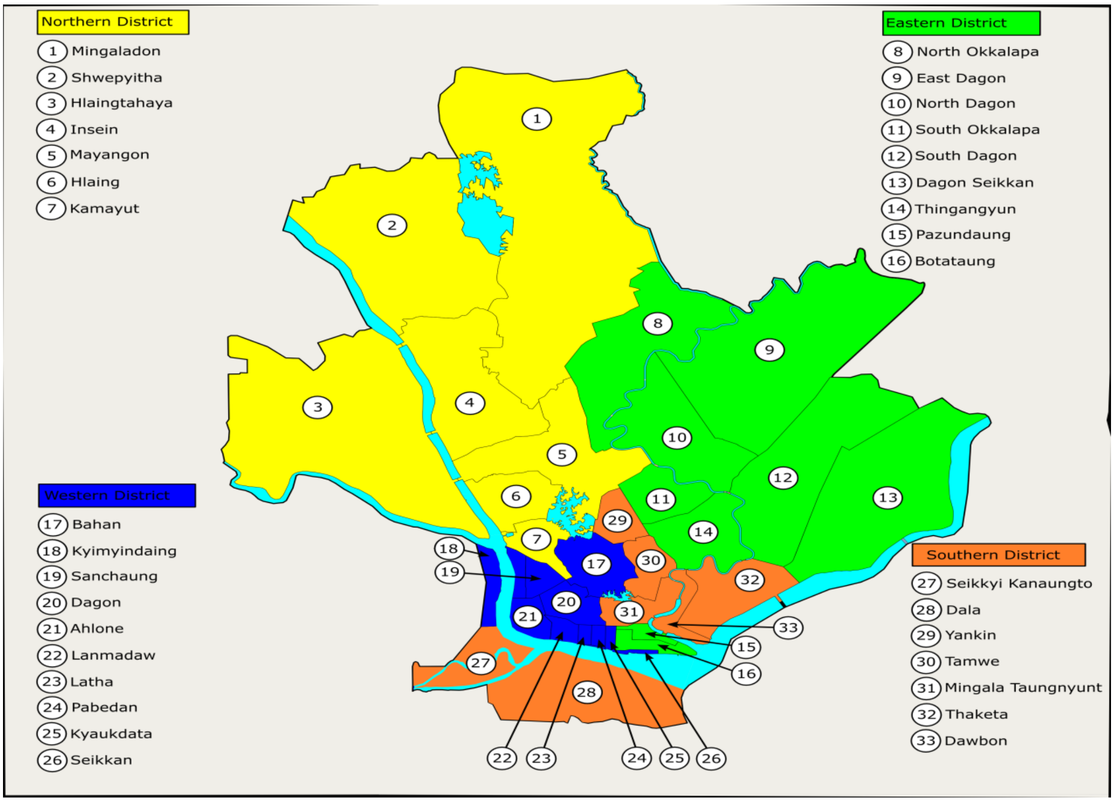


Figure 4.2 Townships in Yangon Region

4.1.2 Yangon Bus System

Public transportation network of Yangon is creating the network element structure by recording the topological relations among of the bus stops and routes. Really, it does not exist in practice. Yangon Bus System is composed of 100 bus lines and bus numbers are designated as different colors. For some bus stops, although the bus number is same, bus number's color is different depend on the bus's route. In the previous system, many types of buses such as Special Bus, Big Bus, Dyna, Mini Bus, SRT and Air-Conditioned Bus are used. For this case, most of the buses are small.

Nowadays, government removed uncomfortable small buses for the public bus system. Regular buses are replaced by air-conditioned buses to give the comfortable service to passengers.

The Yangon Bus Service, also called YBS, is the main local transportation for Yangon Region. It runs from early morning 5:00 a.m. till 12:00 p.m. The cost is very cheap about 200 or 300 Kyats. There are six bus colors for different regions of Yangon, Myanmar as described in Figure 4.3. Among of the100 bus lines, the proposed system implements for 84 bus lines. The routes that run in some bus line are described in Table 4.2.

Figure 4. 3 Different Colors in Bus Numbers

Table 4.2 Bus Routes

Bus Line	Routes
Bus No (1) In front of Hle Gu Market	Tar Sone -->Da Yae Pho -->Pauk Gone -->Ye Mon, COT-->Za Yet Circle-->In Tine -->Ta Yoke Gone -->Tha Net Pin-->Thingangyun-->Ywar Thar Gyi -->Taung Myaut Intersections-->San Pya Market -->Zawana-->Thumingalar-->Yadanar Road -->Myin Tha-->San Pya Market.
Bus No (2) Yu Za Na (Khine Shwe War Road)	Pyi Htaung Su Road-->Hlawgar Road-->Maung Ma Kan Road-->No. 2 Road-->Taung Myaut Intersection -->Pyi Htaung Su Road--> General Ba Htoo Road North Okkalapa Circle-->Khay Mar Thee Road -->University of Medicine (2)-->North Okkalapa Hospital-->Thu Dhamma Road-->Chaut Kwae--> Aung Mingalar Highway Bus Station

4.1.3 Network Connectivity

When the transportation system is implemented, the study on behaviors of roads network is important infrastructure in most countries. The behaviors of this system: how to connect the roads geometry, roads' directions, Bus lines that run on the road, bus-stops and bus routes. There are two different types of networks (1) dynamic network and (2) static network. Table 4.3 describes the characteristics of dynamic network and static network. Bus stops are used as the vertices and edge is the connection between two vertices. The length of each edge is the distance between two vertices. Each vertex has a value of latitude and longitude and each edge's length is represented in kilometer. Yangon bus network considers direction about the network connectivity due to the most of the bus lines runs in two ways for going and coming. Yangon bus network is also assigned as the static network because it does not need to change the topology of bus network's connection. However, it is the complex network because different bus lines pass in the same route.

Table 4.3 Dynamic and Static Network

Dynamic Road Network Characteristics	Static Road Network Characteristics
1. The network topology can change over times 2. Nodes and edges can increase or decrease 3. Length of weight can vary over times	1. The network topology does not change depend on time but the network configuration can modify 2. The number of nodes and edges cannot change 3. Length of weight cannot vary

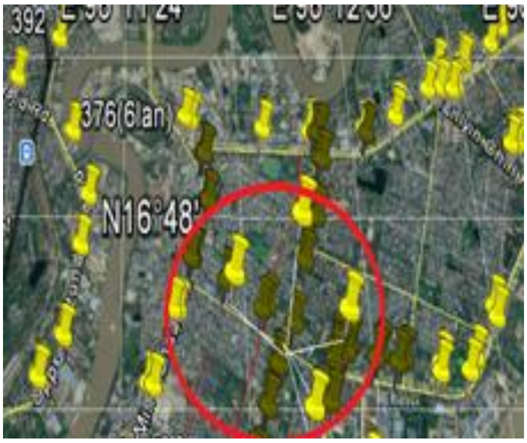
4.1.3.1 The Nature of the Bus Stop

A network is the connection together with vertices and edges according to network topology. Single bus stop is assigned as a vertex and the links between two vertices are occurred according to the routes. The routes with direction are considered when the vertices are connected for network construction. A bus stop can be passed by many bus lines due to different bus numbers run on the same road. This signature can be viewed clearly in the below Figure 4.4(a) and this phenomenon is demonstrated on

Google Earth as described in Figure 4.4(b). Figure 4.4(a) describes the bus lines passing in a road using Java Script Object Notation (JSON). For example, KanBaeLan Road is passed by bus lines “12” and “29”. After routes are fixed at the network, each vertex may be many incoming and outgoing edges as described in Figure 4.4(b).


```
{
  "name": "ကံဘဲ့လမ်း",
  "bus_ids": [
    "12",
    "29"
  ]
},
{
  "name": "ကျန်စစ်သားလမ်း",
  "bus_ids": [
    "7",
    "29",
    "30",
    "53",
    "54"
  ]
},
```


(a) Bus Lines



(b) One bus stop with multiple edges

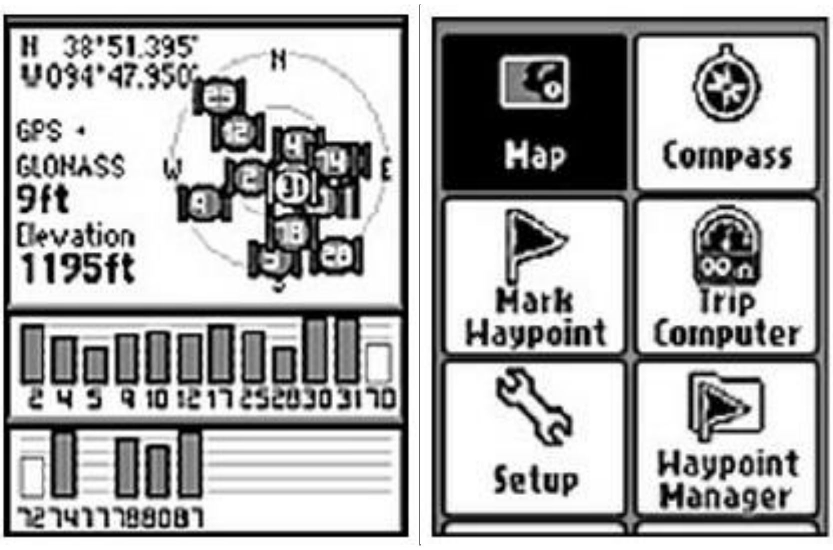
Figure 4. 4 Nature of Yangon Bus Network

4.1.4 GPS Etrex-10 Handheld Device

The GPS Garmin eTrex 10 can provide to mark the routes and input the waypoints easily. This GPS device can read the GPS Data under the dense tree or the sunlight and also display current location address accurately. The eTrex 10 can also provide Paperless Geocaching. This device can store not only location data but also terrain. Moreover, it supports for spine-mounting decorations such as Biking, Boating, Fishing, Hunting, Hiking, Motorcycles, Off roading, Overlanding and Sailing. Figure 4.5 demonstrates this GPS device.

Figure 4. 5 GPS Device

GPS Garmin eTrex 10 can produce geocaching GPX files and this can be extracted from GPS device using Computer to apply in the geographical research area. Moreover, this viewpoints and GPS data can be viewed clearly on this GPS device. GPS Garmin eTrex 10 GPS can track Glonass satellites and GPS simultaneously. When the device is on, it tries to get the satellite connection and the number of satellites that do in current location can be checked as the following Figure 4.6(a). To locate itself about GPS, this device requires at least four satellites. Figure 4.6 describes the menu of GPS Garmin eTrex 10 and to check satellite connection, the user can click the last icon of the menu.



(a) Checking the satellites (b) Menu of GPS Garmin eTrex 10

Figure 4. 6 The Functions of GPS Garmin eTrex 10

4.2 Data Structure of Public Transportation

Yangon Bus System of Public transportation contains (1) bus stops (2) bus lines and (3) routes and (4) bus links. Each bus stop contains not only latitude and longitude but also address detail such as bus stop's name, road's name and township's name. There are two types of bus lines: ring line and one-way line. Starting point and end point is the same when the bus line is the ring line structure. One-way line may vary about starting point and end point and they different uplink and downlink generally. Routes are needed to check carefully because these routes are different nature: ring line and one-way line. When the connection is incorrect about the links between bus stops if the route's nature cannot be specified properly. Firstly, this system constructs the graph database by connecting between the bus stops according to routes. To save and retrieve the required data of the public bus transportation system easily, the suitable data structure is needed to apply. There are many data structures for data storage. For example, File format, matrix construction, caching and database is the popular techniques about the storage management of data. The system uses the xml file at the client side to retrieve the required words for the keyword search. Then, these input data are passed to the server and the calculations are made at the server extracting the required data from Oracle Database. Finally, the results are passed as the Json object to the client side over the internet.

4.2.1 Data File

There are many types of storage file and these are used according to the system's requirements. Microsoft excel and Microsoft access files are used for data storage in some systems. These files own many similarities and both tools can store a lot of data, execute queries and retrieve sub-part from data and do the calculations on the data. For data integrity when the data is accessed by multiple users, Microsoft access is better than Excel but Excel is more suitable for calculation. But these files have long time when the data is passed from one place to another. JSON and XML is the light weight for data carry when these files are passed within different places. Moreover, the space is less when these files are stored in the hard disk. The proposed system contains two parts: client and server. At the client side, to retrieve the required data when the user makes key word search, XML file is used for storage the Yangon Bus Network dataset because it can store these XML files in phone's storage easily and lightly. The space of the file is less than the space of database.

4.2.1.1 Extensible Markup Language (XML)

Extensible Markup Language (XML) is used to create information and it is extensible because it can do unlimited and self-defining about the markup symbols. It is a simple document structure of the Standard Generalized Markup Language (SGML) standard. XML can be used together with Hypertext Markup Language (HTML) for Web Applications. It is also similar HTML tag but HTML cannot create own tags and it is mainly used for the visual effects such as beautiful user interfaces of the Web Pages. The web browser (Chrome, Firefox, Opera, Safari) do to interpret HTML documents to determine how to display the document. The Figure 4.7 describes XML file of proposed system. There are many records in the XML file of the Public bus transportation system. One record contains bus line number, node name, the direction of a route, bus stop name, road name and township name in English and Myanmar and spatial data (latitude and longitude) that represented as the defined tags.

```xml
<record>
  <bus_id>1</bus_id>
  <point_id>2011</point_id>
  <sequence>1</sequence>
  <busstop_nameen>Hlegu Zay</busstop_nameen>
  <busstop_namemm>လှည်းကူးဈေး</busstop_namemm>
  <road_nameen>No. 1 Main Road</road_nameen>
  <road_namemm>အမှတ်(၁)လမ်းမ</road_namemm>
  <township_en>Hlegu</township_en>
  <township_mm>လှည်းကူး</township_mm>
  <lat>17.096823</lat>
  <lon>96.219456</lon>
</record>
```

Figure 4. 7 Xml Tree Structure of Proposed System

4.2.2 JavaScript Object Notation (JSON)

JSON can be used when data is sent in client and server application. Syntax structure is same to the JavaScript objects codes because it derives from the JavaScript object notation but its format is text only. JSON contents can be read by writing any programming languages. The proposed system sends the required information with results from server to consumer using JSON object over the internet. Latitude, Longitude of all points about two shortest paths will include in the results.

JSON objects can create using multiple name and value pairs and it can be written inside curly braces. For example, let key value for latitude data be “Resultvalue” and then the symbol of full colon “:” is written. After writing full colon, the values can be written. When the client side is reached, the “Resultvalue” object can be used to retrieve spatial information of the points that include in shortest results. For example, the JSON object is represented as follows:

Resultvalue = {

"latitude":["96.223276", "96.225736 ", " 96.294871"],

"longitude":["17.098546", "17.099458 ", " 16.960345"],

};

4.2.3 Spatial Database

Spatial Database contains geo information on the earth. An Oracle database is a collection of data treated as a unit. Oracle database is already built in eclipse tool that supports for creating Standalone Java Application and Java 2 Enterprise Systems. The purpose of using database is to store and retrieve required information. It is reliable for information management about large amount of data when the same data is accessed concurrently by many users. In addition, this can provision to get high performance when the database transactions are occurred. Database server can provide efficient solutions for failure recovery. The spatial information that gets from map can be saved at MySQL database using Quantum Geographical Information System (QGIS). MySQL database is a part of Oracle Database Server. This database can hold geometry values such as points, line string and polygon. The proposed system uses the Oracle Database to save the related information after data collection. Data Collection is made in different ways: saving spatial data from Garmin Hardware Device, getting from the authority of Yangon Region Transport Authority (YRTA) and using the geospatial tools such as Google Earth, Google Map and QGIS etc. The working of the query statement of Oracle and MySQL is similar for transaction of the system. Figure 4.8 also mentions MySQL database that saved the spatial data getting from Google Map in QGIS.

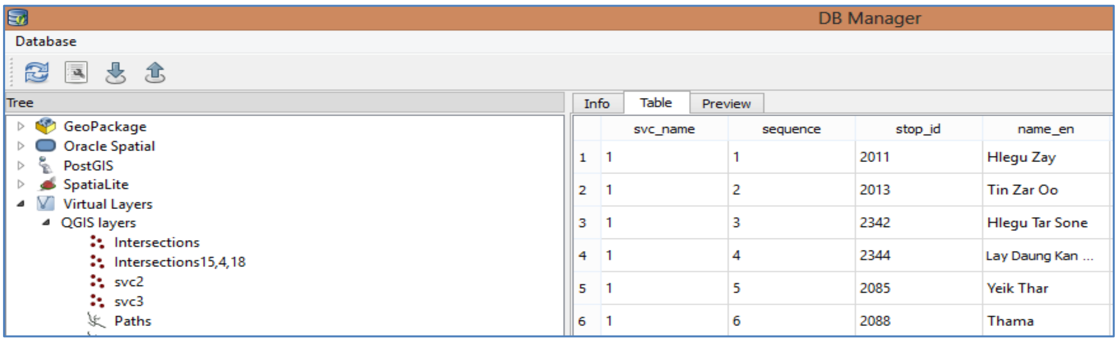


Figure 4. 8 Sql Database in QGIS

4.3 Combined Forward and Backward Heuristic Search (CFBHS) Algorithm

The proposed system applies the forward and backward search for finding the shortest paths. The system uses two threshold values, the first one is the calculation of threshold value called "th" using proposed degree-based threshold calculation technique. It is used for separating the large tree into small trees. When the network is huge and dense, the threshold value may be large. The second one is defining the number of shortest results $|rs|$ that will produce from each subtree. Figure 4.9 describes the step by step works of the proposed method using system flow.

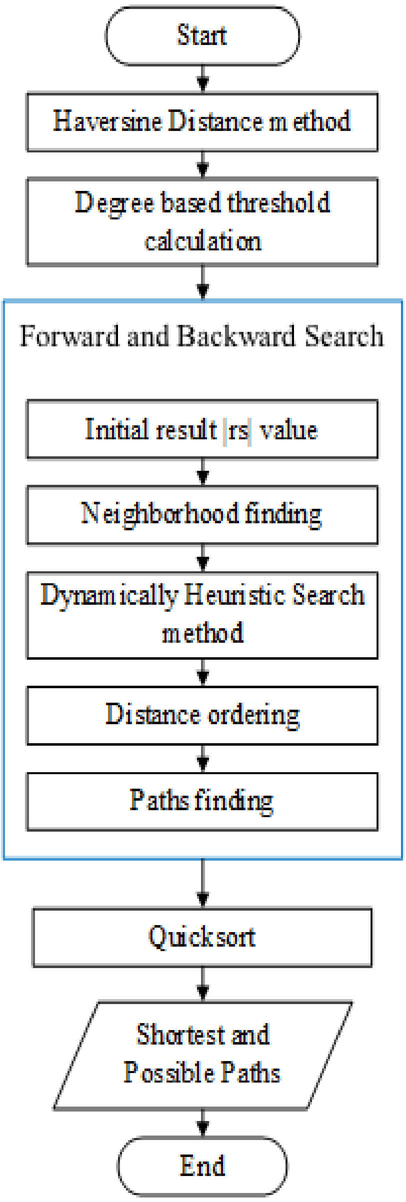


Figure 4. 9 System Flow Diagram of Public Bus System

The customer firstly inputs source and destination to the system and then the system checks the number of nodes after expanding the current node. When these numbers are greater than or equal to the defined threshold value, tree is divided into subtrees. Each small tree will calculate the shortest results until the number of results reach to the predefined value $|rs|$. While the result is calculating from each subtree, the heuristic searching is run. CFBHS algorithm will search the shortest results using forward (source to destination) and backward (destination to source) search separately. The proposed algorithm does not traverse all nodes of the network and it especially considers only on the adjacent nodes using heuristic search at finding the shortest and possible routes. Finally, the proposed system will determine which one is the shortest according to the shortest results from source to destination and from destination to source.

4.3.1 Proposed Method

The proposed method of the system is described in Figure 4.9. During the pre-processing phase, the graph database is constructed and spatial information is saved in it and the distance between nodes is calculated using Haversine Distance according to Equation (4.2), this is especially considered on the sphere for latitude and longitude data.

$$haversin\left(\frac{d}{r}\right) = haversin(\emptyset_2 - \emptyset_1) + cos(\emptyset_1)\,cos(\emptyset_2)haversin(\lambda_2 - \lambda_1) \ldots\ldots \quad (4.2)$$

where d represents the distance between two points with latitude $\emptyset_1$, $\emptyset_2$ and longitude λ_1, λ_2 , r assigns the value as the radius of the Earth. Threshold value th is needed to split the large tree into small subtrees. Therefore, after pre-processing, a degree-based threshold calculation method is applied to find it. If the network is same, it does not need to consider for calculation again.

In Figure 4.9, the processes within the blue box are executed to compute the shortest and possible routes from each subtree and finally the results are saved in the database. After getting the results from both directions, global optimum results are computed using quicksort algorithm. It is used to compare the local optimum results and sort them and the time complexity of this algorithm costs O(NlogN) in average case. It is better than popular sorting algorithms such as bubble sort, selection sort and insertion sort.

The proposed method contains three main parts: (1) finding threshold value “th” and splitting the big tree into small subtrees (2) calculating the heuristic values for deciding the expand node for next level and (3) computing and collecting the local optimum results and finally finding the global optimum results.

4.3.2 Subdivision Scheme

Complex bus network can be divided into segments to be assigned the jobs on different pieces described in Figure 4.10. Two threshold values th and rs are used. To separate the tree, threshold value th is computed according to degree-based threshold calculation approach. This proposed method contains four phases: (1) checking the degree of vertices (2) division different subtrees (3) finding the shortest paths and (4) considering the accuracy and performance of the shortest paths.

According to the behavior of the graph and degree of vertices, threshold value may differ. Different degree values of the graph are applied as the testing threshold values. The tree division occurs if the current level owns the number of edges that equal or greater than predefined threshold. Each segment calculates the paths using dynamically heuristic approach with selected start and goal nodes. After calculating the routes, it decides the threshold value according to two factors: the performance and the accuracy of the shortest routes. Tree is divided into segments when the level 3 is reached in Figure 4.10.

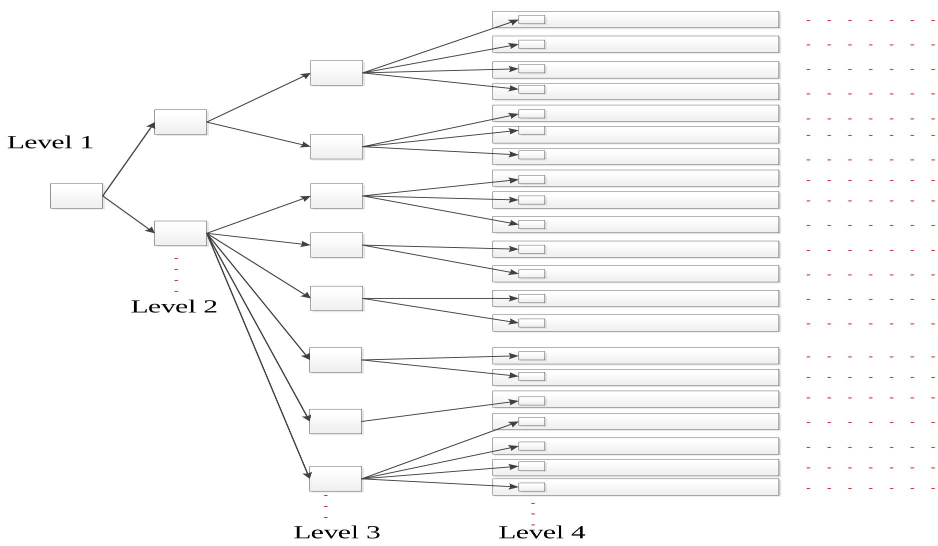


Figure 4. 10 Demonstration of Subdivision

Firstly, the system traverses the path starting from level 1 to next level or step. When the next step is reached, two parts are worked that are tracking the path and the degree of vertices of the current level. When a path among of the paths touch to target, this route is placed into the database server. After this situation, this route will not include in tree currently. If the predefined threshold value is smaller than or equal to the edges' condition of current level, tree will be spitted. Shortest results will be searched according to the predefined value rs. In Figure 4.10, each segment $\{st_i \mid 1 \leq i \leq 8\}$ will calculate three local optima for their results.

4.3.3 Choosing the Next Expanded Node

The proposed system uses heuristic values while finding the results in route analysis phenomenon and it calculates these values dynamically while execution is made. The system does not need to consider the visited nodes again and other subtrees while selecting the next expanded node from current subtree. Each subtree works separately to find the shortest results. The distance is measured between current neighbor nodes and target node using Euclidean Distance. After computing the distances between the neighbors and target, orderly linked list is used to store the nodes according to descending order of these heuristic values and the minimum heuristic value is chosen for the next level. Visited linked list is used to store the result nodes while going each level to get the shortest path.

4.4 Finding Shortest and Possible Paths Using CFBHS

This section describes the pseudo code to calculate the shortest and possible results. CFBHS algorithm finds the shortest and possible paths between source and target point according to the forward, backward and heuristic searching methods. The system will start from the MainFunction (Algorithm 1). It creates two objects of CFBHS java class to do forward and backward search and it will pass the parameters (start, end, flag) parameters together with objects. The functions of the CFBHS java class will work via from created objects from MainFunction. The flag variable is used to test whether both forward and backward direction job is completed. At first, calculate function of CFBHS java class will be run. After that, it calls another function named divideLargeTree (Algorithm 3) to produce the small sub trees. This algorithm checks the number of edges of current level by comparing the predefined threshold value. The number of edges of current level is same to the number of adjacent vertices of all nodes of that level. The level that exits in root of a tree is called level 1 and this level is checked at Algorithm 3 and other levels are detected according to Algorithm 3 and 4. Section (4.5) describes the detailed work for subdivision. After dividing the large tree, Algorithm 3 calls the CFBHS function of Algorithm 5. Algorithm 5 calculates the local optimum results from each subtree based on heuristic search and it finds the distance between current neighbor and target nodes using Euclidean Distance and it does not consider the unnecessary nodes and edges while trying to reach the goal. The number of results from each subtree is searched consistently with predefined threshold value “rs”. If the “rs” value is “m”, m local optimum results will be produced from each segment. After producing one local optimum result, this will be saved in database and the number of local optimum result will be increment by calling the printPath (Algorithm 6) function from Algorithm 5. Finally, after working in both directions for finding the local optimum results, the calculate function (Algorithm 2) will demand printGPath (Algorithm 7) for finding the optimal shortest results using quicksort algorithm with parameterized local optimum results.

Algorithm 1 MainFunction_

1: Create the two objects to call CFBHS class

2: firstobj(startnode,targetnode,flag) //forwardsearch

3: secondobj(targetnode,startnode,flag)//backwardsearch

Algorithm 2 calculate

Input: start node, target node, flag

1: call divideLargeTree(startnode, threshold) function

2: **if** (flag == 2) then

3:    call the printGPath function

4: **end if**

Algorithm 3 divideLargeTree

Input: startnode, threshold, Graph

Output: subtrees

1: j ← 0

2: adjacentLinkedlist ← adjacentNodes(startnode) // level1

3: i ←size(adjacentLinkedlist)

4: k ←i

5: **while** (i>0) do

6: check the neighbor nodes

7: if the current neighbor node is equal to the target node, store this traversed path into the local optimum database

8: remove this traversed path from tree

9: i←i-1

10: **end while**

11: **if** (k >= threshold) then

12: separate the tree into subtree

13: call the CFBHS(traversedpath) function to find the local optimum results from each subtree

14: **else**

15: work step 16 to check the number of adjacent nodes of all nodes that exist at current level // level 2 and other levels

16: **while** (k>0) do

17: j← j+ CheckNumofAdjNodes (adjacentLinkedlist(k))

18: do from step 6 to 8

19: k ←k -1

20: **end while**

21: **end if**

22: **if** (j>= threshold) then

23: do step 12 and 13

24: **else**

25: k←j

26: j ←0

27: do step 15

28: **end if**

Algorithm 4 CheckNumofAdjNodes

Input: A node that wants to find adjacent nodes

Output: number of adjacent nodes of current input node

1: adjacentLinkedList ← adjacentNodes(node)

2: i ← size(adjacentLinkedlist)

3: return i

Algorithm 5 CFBHS (Combined Forward and Backward Heuristic Search)

Input: Graph, start node, target node, traversed path (visited linked list), flag, rs

Output: local optimum paths

1: get the last node from the traversed path in visited linked list of current subtree

2: find the adjacent nodes of the current node

3: insert the adjacent nodes of the current node to the Linkedlist named as adjacentnodes

4: check each node in adjacentnodes

5: **if** a node in adjacentnodes contain in visited linked list

then

6: remove this node from adjacentnodes

7: **else**

8: compute the distance between the nodes in adjacentnodes linked list and target node by using Euclidean distance

9: **end if**

10: store the nodes in the orderly linked list according to descending order using heuristic distance values

11: choose minimum node from orderly linked list

12: add this choice node into the visited linked list

13: **if** current choice node is equal to target node then

14: call the printPath function

15: **else**

16: call the CFBHS function recursively

17: **end if**

18: **if** (current result number > number of predefined result named as rs)) then

19: terminate the job of current subtree

20: **else**

21: remove the last two nodes (target node and a node before target) from the visited linked list

22: choose the second minimum node from orderly linked list

23: add this node at the end of the visited linked list

24: call the CFBHS function recursively

25: **end if**

Algorithm 6 printPath

Input: Graph, visited Linked list

1: Calculate the total distance of a route that gives from the visited linked list

2: Store the local optimum result into the database

3: Increment the current number of results

Algorithm 7 printGPath

Input: local optimum paths

Output: shortest and possible paths

1: Use the Quick-Sort function to get the shortest and possible paths

---

4.5 Working Example

The working of the algorithm for finding the shortest path between source node “s” and target node “o” is demonstrated in Figure 4.11. Threshold value th and rs are assumed as 12 and 2. Firstly, the algorithm checks the number of edges of current level to divide the tree. The total number of edges of node s are 3 at level 1 and it is smaller than threshold value 12. Therefore, next level is checked. At level 2, the number of edges of three nodes d, c and b exist 3, 3, 4 and it is 10 totally at that level. Then, next checking will come for level 3. Dividing the tree will occur at level 3 because it owns the edges that greater than predefined threshold value. After that, the system will get six subtrees such as (s,d,h), (s,d,g), (s,c,g), (s,b,c), (s,b,f) and (s,b,e) that already traversed the graph from level 1 to level 3.

Each subtree computes the local shortest results using heuristic approach until the second threshold value rs is equal. For example, the neighbors of h are d, g, k, l, m for the subtree (s, d, h) and d is already visited. Therefore, the distance values between other remaining nodes g, k, l, m and target node o are calculated. k is the nearest distance value according to heuristic method, spatial latitude and longitude data and it is selected for going to next level. This subtree (s, d, h) produces the shortest path s, d, h, k, o. In this way, each subtree will find two shortest results. Finally, the system will find the global shortest and possible results using the local optimum results from each subtree and quick sort algorithm. The shortest results can be searched in the short time because of dividing the large tree into the small subtrees, using heuristic based technique and searching nodes for going next level depend on adjacent nodes of current node.

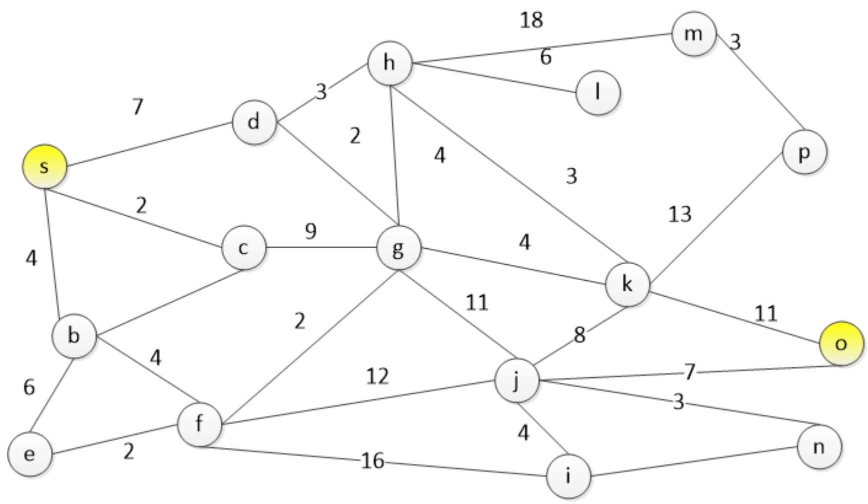


Figure 4. 11 Undirected Graph

4.6 Transit Nodes Information

The working system of public transportation is a little different if it is compared with the taxi system because the network structure between them is not the same. The road network of the taxi system can find the shortest route on any road and street that exits on the map. However, the public bus system can use only the specified ways to run public buses even if the other roads exits. Some ways have the circle structure, that is, the forward and backward direction is the same but some are different. Moreover, the passengers need to know the bus numbers that they will take for the shortest route between starting bus stop and target bus stop. Therefore, the system defines the formulae for finding the direct and transit bus numbers after finding the shortest routes. It is easier and more efficient to get bus transit information by using the bus data after finding the shortest route because it needs to consider on the shortest information instead of using the whole database. It especially defines direct route, one transit and multiple transit formulae. YRTA managed a completed and comfortable bus travelling architecture in 2017, aiming to be easy for passengers while travelling by bus.

4.6.1 Direct Route

Firstly, the system searches the shortest routes between the start and end point. And then, the bus information such as bus number, point number and other required data about the shortest routes are extracted from Yangon Bus Database. This bus information is saved in temporal table of database and it is used for finding the transit and direct information related with shortest route. Direct route can be searched using

$$Busnames(A) \cap Busnames(B) \quad \ldots\ldots\ldots\ldots\ldots\ldots. \quad (4.3)$$

where let A is the end node and B is the start node. If the bus names of two nodes are same, the passengers can take using direct bus to reach the target location. The user inputs start bus stop and end bus stop to the system and the system finds the related points of two bus stops. In Figure 4.12, let assume point number "1001" is for start bus stop and "1007" is for end bus stop. The bus names pass for point number "1001" are "B1" and "B2". The bus names for point "1007" are also "B1, B2". By intersection these bus names of two points, the system will get the result as "B1, B2" for direct route related with current searched shortest route.

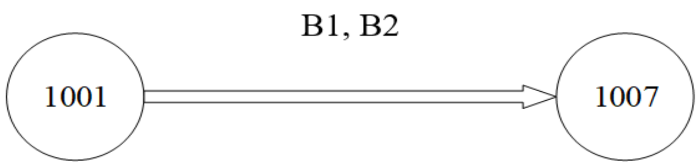


Figure 4. 12 Direct Route between Two Bus stops

4.6.2 Transit Route

If the route does not have the direct bus name, the system needs to produce transit bus stop and the next required bus name for the user. The system finds the bus names of end point, then retrieve the related points with bus names from database. After that, it searches the bus names for each point. If these bus names of a point among the points are same about the bus names for the start point, this point is the transit point. To find the first transit route, the system can use the following Equation (4.4).

$$Busnames(B) \cap [Busnames(\forall point(Busnames(A))] \quad \ldots\ldots\ldots\ldots \quad (4.4)$$

where A is the end point and B is the start point that user desires. In Figure 4.13, point “P1” can be passed by bus “B1, B2, B3, B5” and “P1” is one of points from bus stops related with end point. Start point can be passed from bus name “B5” and end point can be passed from “B1, B2, B3”. Therefore, according to this phenomenon, the user can take bus name “B5” to reach from start point to transit point number “P1” and bus names B1 or B2 or B3 can be used to reach from point “P1” to the destination. According to this approach, the system finds the transit bus numbers after finding the routes using proposed heuristic based algorithm. Figure 4.14 describes three transactions about detailed public bus information after getting the shortest result.

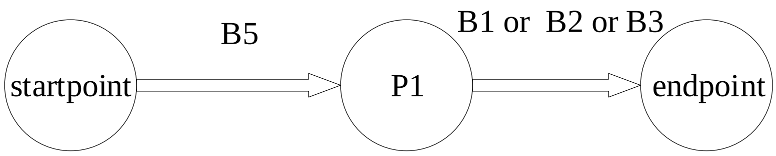


Figure 4. 13 Two Transactions between Two Bus Stops

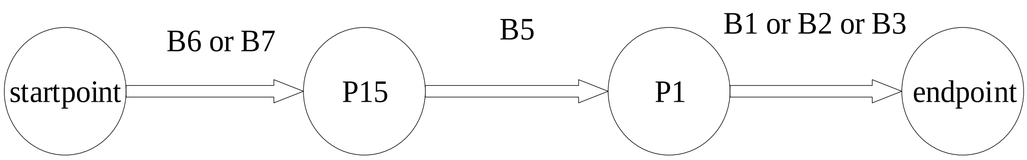


Figure 4.14 Three Transactions between Two Bus Stops

4.7 Summary

This chapter describes the architecture of the proposed system. The proposed system uses the client server architecture. People can use the proposed system on the android mobile phone. Android Programming Language is used at the client site and Java 2 Enterprise Edition(J2EE) Programming Language and tomcat server is used at the server site. Localhost and port number 8089 is used to run the server. Different Data Storage Techniques are described in this chapter. These are Microsoft Access File, Microsoft Excel File, XML file, Json file and Oracle database. Oracle database is used to retrieve the spatial data and related information of the system and it is stored at the server site. The system stores XML file at the client site because the system intends to support auto complete keyword search technique for the input of the users. Simple API for XML Parsing (SAX) parser is used to retrieve the required information from XML file and it is faster than Document Object Model (DOM) parser because SAX parser does not parse the whole document. Json Object is used for passing the information between client and server to reduce the data passing time. Moreover, this chapter describes the townships of the Yangon Region and the nature of Yangon Bus System. And it explains about the proposed algorithm CFBHS in detail working example and step by step explanation of pseudo code. The proposed algorithm can solve the unnecessary nodes repeatedly considering problems of A* algorithm and its time complexity is faster than heuristic search A* algorithm. Chapter 5 will discuss the experimental results and evaluation after inspiring and describing the proposed system and implementation in this chapter.

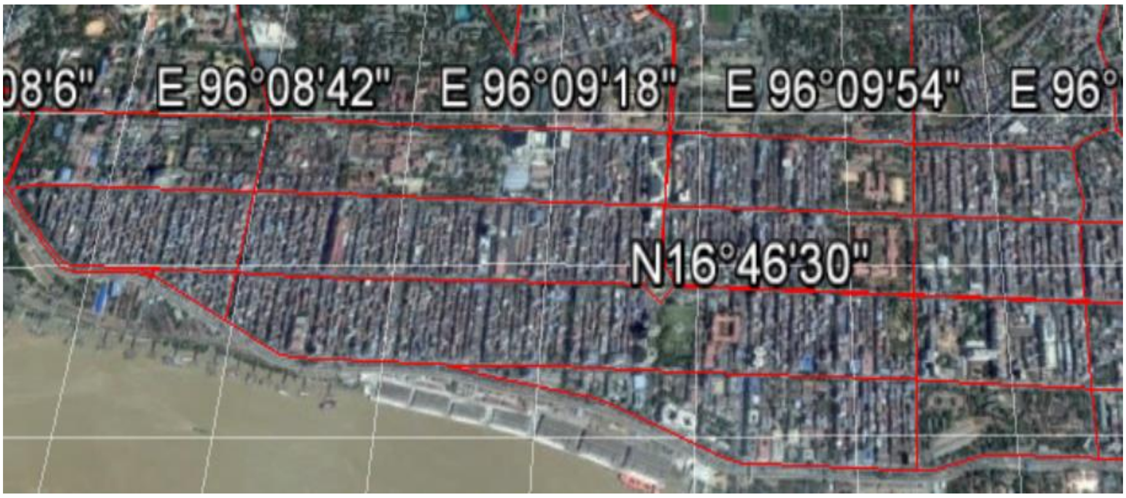
CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATION

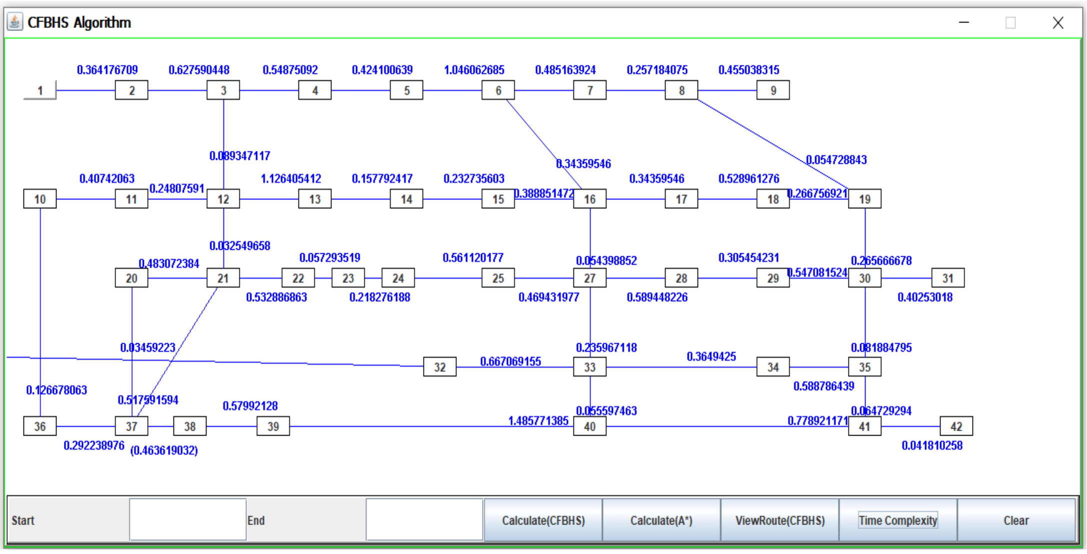
This chapter describes the evaluation and experiment results for finding the optimal and possible paths of public bus system in Yangon Region. The proposed algorithm is applied in two different networks such as Road Network and Bus Network. Although these two networks are seemed to be similar, their architecture is different. Bus network is more limited because it cannot go every road while trying to get the optimal route. It depends on the architecture of the bus lines to reach the destination point. Yangon region is organized by 33 townships and Latha, Lanmadaw, Kyauktada, Pabeldan, Botahaung and Pazundaung are downtown areas and the main transportation townships. This chapter includes five major parts: checking different thresholds for different components, the complexity assessment between different algorithms and data components; describing the statistical analysis for different graph networks such as undirected Yangon downtown road network and Hypergraph Yangon bus network; implementing the Yangon bus system on android mobile phone using the proposed algorithm and subjective evaluation for Yangon bus system.

5.1 Using Different Threshold Values for Different Components

Yangon downtown map is constructed as a simulation graph for the standalone system. By constructing the simulation graph, the optimal routes can be easily checked whether these output values are correct or not between two nodes. The simulation graph contains 41 nodes and junction points are assigned as the nodes. Yangon downtown graph is constructed for the road network and one way is enough to consider for this graph while finding the optimal route. The next graph is Yangon Bus Network and node is assigned as the bus stops and an edge is joined between two nodes according the bus lines of the bus network architecture. Two way is considered for the bus network and it is more constrained than the road network. The system uses the degree values of the nodes for the threshold testing parameters th. Figure 5.1(a) describes the Yangon downtown map on google map using google earth tool. The road lines are described using red color and this special latitude and longitude data is imported as keyhole markup language (kml) file to google earth tool. Figure 5.1(b) constructs the Yangon downtown network as the graph network and compares the performance between proposed CFBHS and A* algorithm.

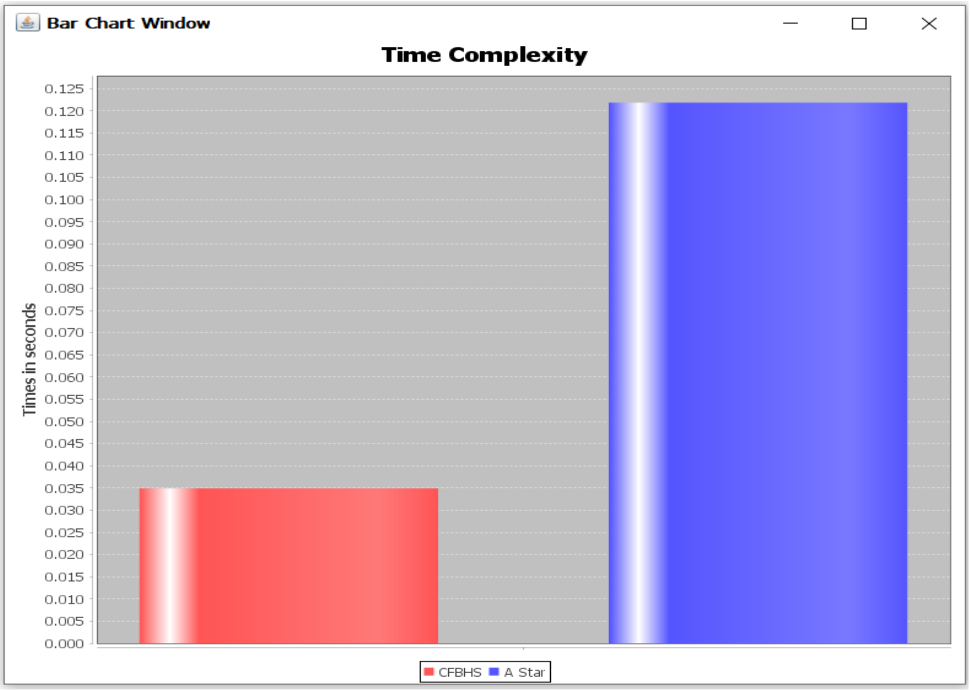


(a) Describing Yangon Downtown Area on Google Map



(b) Simulation Graph for Yangon Downtown Area

Figure 5.1(c) describes the bar chart using JFreeChart Utilities about the comparison time between Combined Forward and Backward Heuristic Search Algorithm and A* Algorithm. These two algorithms can search the shortest result during the short time. Comparison Time Unit is described in seconds as the following Figure 5.1(c).

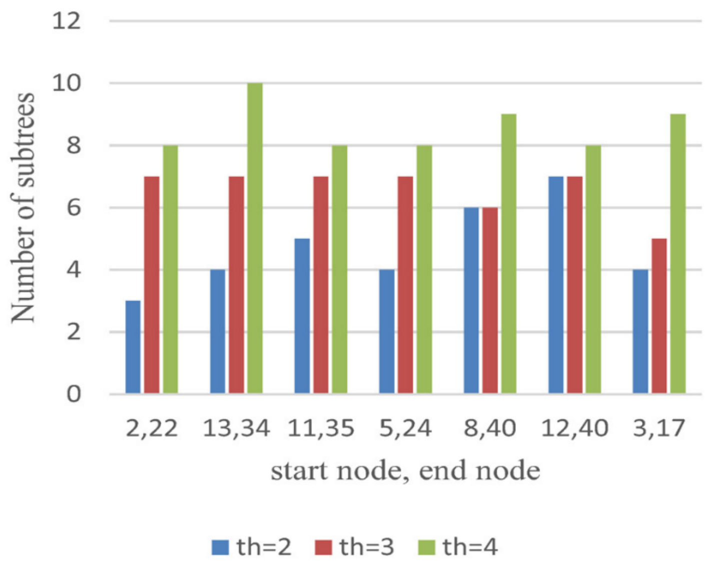


(c) Time Comparison between CFBHS and A* Algorithm

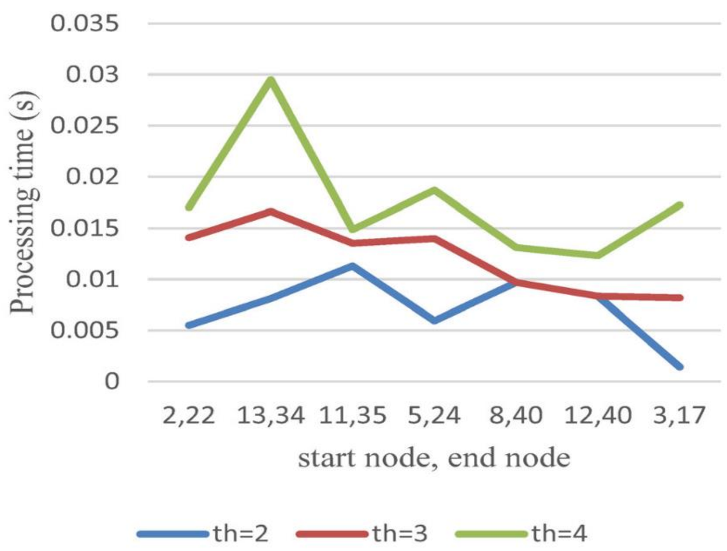
Figure 5. 1 Constructing Yangon Downtown Spatial Data Network into Graph Data Network

5.1.1 Testing Yangon Downtown Map with Different Parameterized Threshold Values

Yangon downtown area contains the vertices that own degree one, two, three and four. Degree one is not used for threshold value because this cannot divide the tree. Other different threshold values such as two, three and four are used as testing parameters for Yangon downtown area. Start and end points about spatial data of Yangon downtown area are randomly selected for comparing the difference between processing time and the number of subtrees of the system. The performance of the proposed system will be faster due to the small number of subtrees and it is directly proportional as shown in Figure 5.2(a). It shows that when the threshold value is increased, the number of subtrees is also increased. In addition, all threshold values can produce the shortest path according to the experimental results but threshold value two is the best parameter because the execution time is the fastest as described in Figure 5.2(b).



(a) Number of subtrees for different thresholds with randomly designated nodes



(b)Processing time for producing the shortest routes with different thresholds

Figure 5. 2 Using the Different Threshold Values

5.1.2 Testing Yangon Region with Different Parameterized Threshold Values

Public bus transportation network is composed of about 1894 nodes and over 7000 edges. The edges are connected according to the designated bus lines. The maximum degree of vertices of this network is degree 17. The different threshold values are tested for the Yangon Public bus network and finally the system decides the threshold value “4” to choose for dividing the tree. Only second shortest path can be often produced when degree 2 and degree 3 is used as threshold parameter as shown in Figure 5.4. According to the comparison, suitable threshold values of two different components differs depend on the nature of the graph and the vertices' degrees of them. Figure 5.3 describes different degree of vertices in Yangon Bus Network. Threshold value is suitable for choosing the degree of vertex that occur many times. In Yangon Bus Network, degree of vertices is 1,2,3,4,5,6,7,8,9,10,11,12,13 and 17. Among of these, degree 2, 3, 4, 5 and 6 occur repetitively as the degree of vertices according to Figure 5.3. Therefore, these vertices such as "2, 3, 4, 5, 6" are checked as threshold parameters because if the threshold parameter is seventeen, the large tree will use subdividing scheme only sometimes.

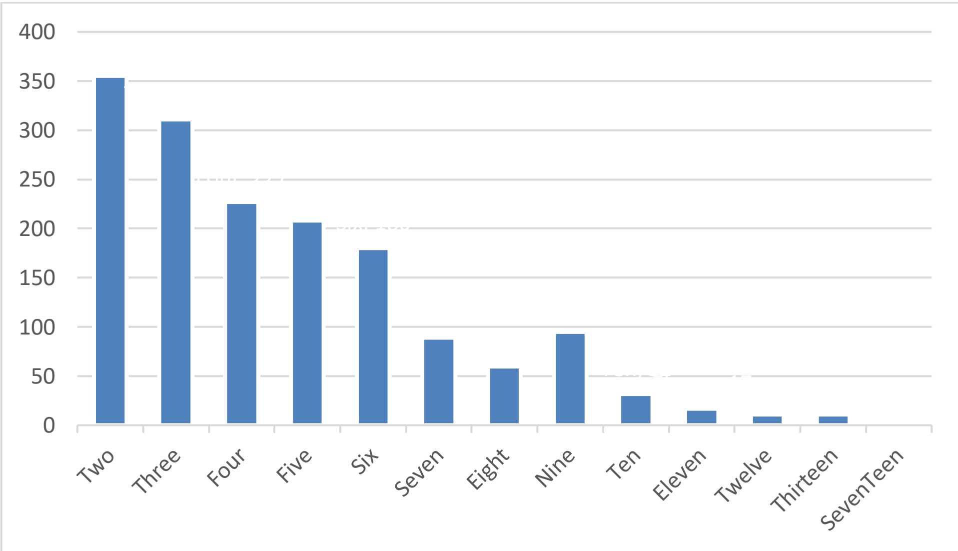


Figure 5. 3 The Number of Vertices according to Degree

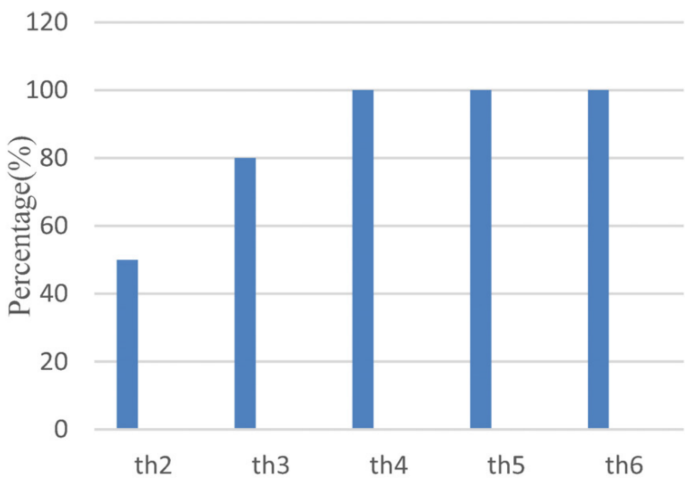
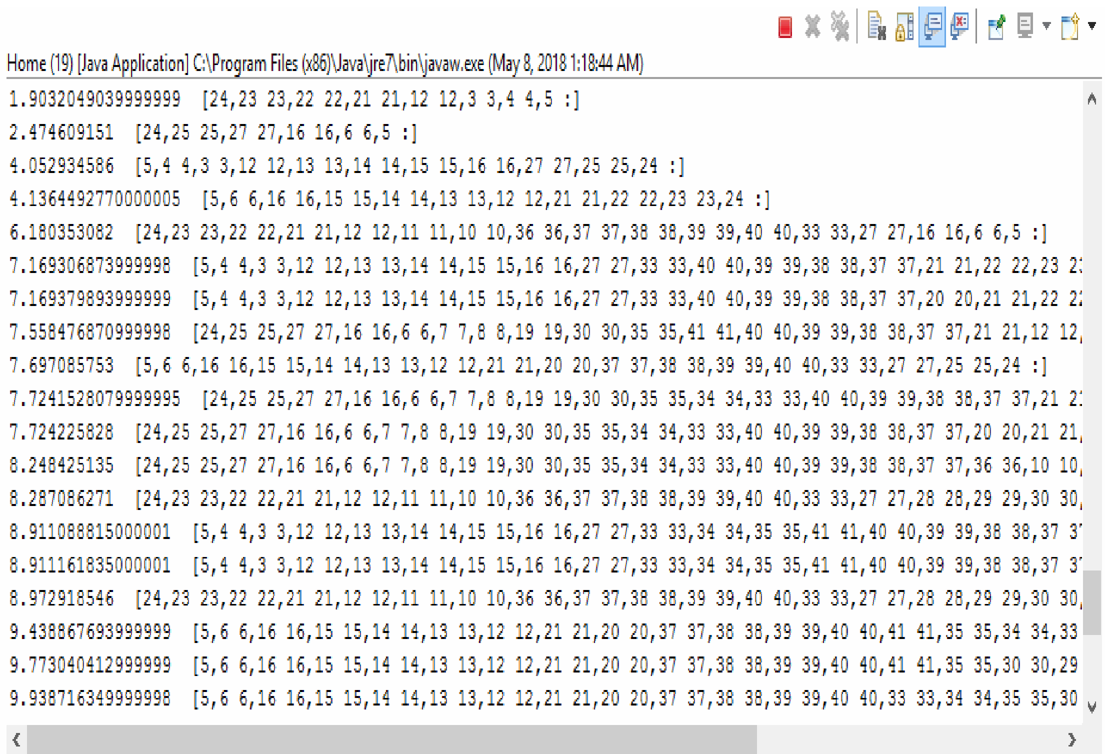


Figure 5. 4 Yangon Bus Network with Different Threshold Values

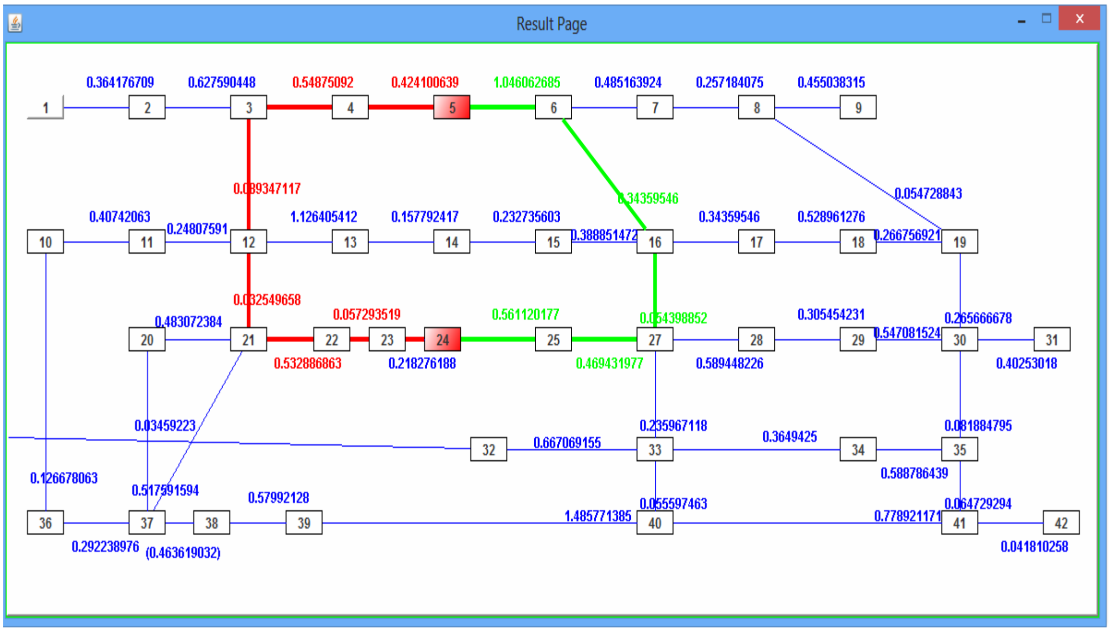
5.2 Time Complexity for Yangon Region

The system can produce not only the shortest route but also the possible routes at the same time. Threshold value two and rs value five are used for finding these shortest results. In Figure 5.5(a), these all results are displayed in text mode of eclipse java integrated development environment (IDE). Two shortest results of Yangon downtown area are displayed on the simulated graph as described in Figure 5.5(b). The first shortest result and the second shortest result is described in red color and green color on the graph. Figure 5.6 describes the comparison about the time complexity among of the proposed CFBHS algorithm and other popular short finding algorithms such as A star and Bi-directional Dijkstra algorithms. Threshold parameter rs is used to define the number of local optimal results for the graph. In Figure 5.6, threshold parameter is defined as one and ten. When the threshold value is ten, each subtree will search ten local optimal results for the graph.



```
Home (19) [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (May 8, 2018 1:18:44 AM)
1.9032049039999999  [24,23 23,22 22,21 21,12 12,3 3,4 4,5 :]
2.474609151  [24,25 25,27 27,16 16,6 6,5 :]
4.052934586  [5,4 4,3 3,12 12,13 13,14 14,15 15,16 16,27 27,25 25,24 :]
4.1364492770000005  [5,6 6,16 16,15 15,14 14,13 13,12 12,21 21,22 22,23 23,24 :]
6.180353082  [24,23 23,22 22,21 21,12 12,11 11,10 10,36 36,37 37,38 38,39 39,40 40,33 33,27 27,16 16,6 6,5 :]
7.169306873999998  [5,4 4,3 3,12 12,13 13,14 14,15 15,16 16,27 27,33 33,40 40,39 39,38 38,37 37,21 21,22 22,23
7.169379893999999  [5,4 4,3 3,12 12,13 13,14 14,15 15,16 16,27 27,33 33,40 40,39 39,38 38,37 37,20 20,21 21,22
7.558476870999998  [24,25 25,27 27,16 16,6 6,7 7,8 8,19 19,30 30,35 35,41 41,40 40,39 39,38 38,37 37,21 21,12 12,
7.697085753  [5,6 6,16 16,15 15,14 14,13 13,12 12,21 21,20 20,37 37,38 38,39 39,40 40,33 33,27 27,25 25,24 :]
7.7241528079999995  [24,25 25,27 27,16 16,6 6,7 7,8 8,19 19,30 30,35 35,34 34,33 33,40 40,39 39,38 38,37 37,21
7.724225828  [24,25 25,27 27,16 16,6 6,7 7,8 8,19 19,30 30,35 35,34 34,33 33,40 40,39 39,38 38,37 37,20 20,21 21,
8.248425135  [24,25 25,27 27,16 16,6 6,7 7,8 8,19 19,30 30,35 35,34 34,33 33,40 40,39 39,38 38,37 37,36 36,10 10,
8.287086271  [24,23 23,22 22,21 21,12 12,11 11,10 10,36 36,37 37,38 38,39 39,40 40,33 33,27 27,28 28,29 29,30 30,
8.911088815000001  [5,4 4,3 3,12 12,13 13,14 14,15 15,16 16,27 27,33 33,34 34,35 35,41 41,40 40,39 39,38 38,37
8.911161835000001  [5,4 4,3 3,12 12,13 13,14 14,15 15,16 16,27 27,33 33,34 34,35 35,41 41,40 40,39 39,38 38,37
8.972918546  [24,23 23,22 22,21 21,12 12,11 11,10 10,36 36,37 37,38 38,39 39,40 40,33 33,27 27,28 28,29 29,30 30,
9.438867693999999  [5,6 6,16 16,15 15,14 14,13 13,12 12,21 21,20 20,37 37,38 38,39 39,40 40,41 41,35 35,34 34,33
9.773040412999999  [5,6 6,16 16,15 15,14 14,13 13,12 12,21 21,20 20,37 37,38 38,39 39,40 40,41 41,35 35,30 30,29
9.938716349999998  [5,6 6,16 16,15 15,14 14,13 13,12 12,21 21,20 20,37 37,38 38,39 39,40 40,33 33,34 34,35 35,30
```

(a) Different Possible Routes



(b)Simulated Graph

Figure 5. 5 Finding the Shortest and Possible Routes for Yangon Downtown

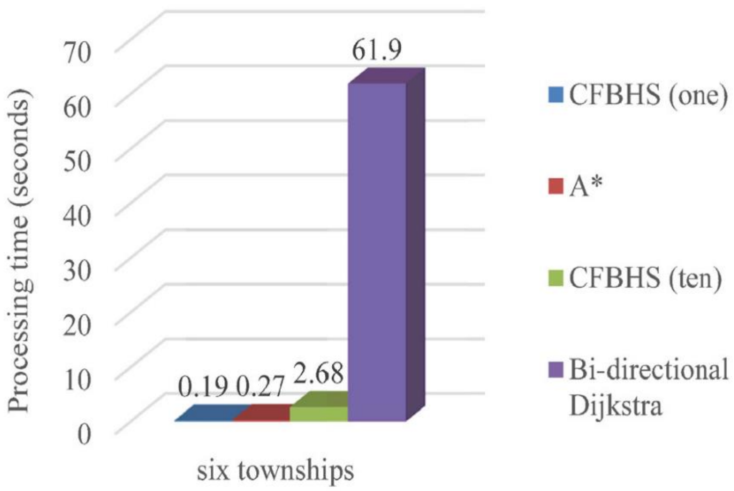


Figure 5. 6 Time Complexity of Three Algorithms for Yangon Downtown

Different value is used as the testing data for the threshold value. When the threshold value is huge, the number of results that produce from the graph is also large as described in Figure 5.7.

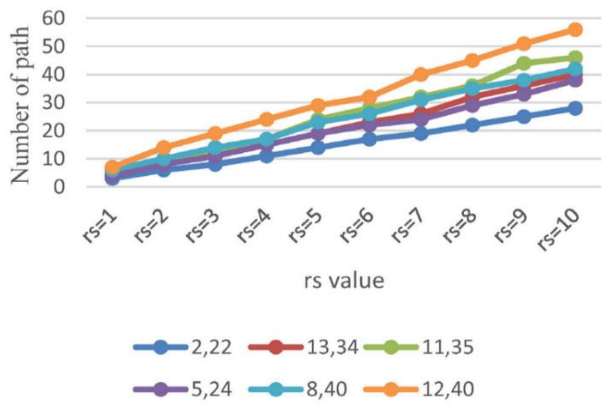


Figure 5. 7 The Number of Paths according to rs value for Yangon Downtown

The time complexity of the proposed method is demonstrated depending on different amount of spatial data. Yangon bus network is constituted 84 bus lines and it contains many nodes and edges. Bus stops are designated as the nodes of the graph. There are 39 townships in Yangon, Myanmar. Among of them, 33 townships are applied as the research area. It contains 1894 nodes and there are many connections between them. The processing time of the proposed algorithm are compared on two different data sets: Yangon downtown road network and Yangon bus network. Although the amount of data set is significantly increased in Yangon bus network, the time complexity is a little different after comparison is made as shown in Figure 5.8.

Figure 5. 8 Comparison of Time complexity of CFBHS for Six Townships and 33 Townships in Yangon Region

The proposed short finding algorithm can search for both shortest and possible routes simultaneously of a network. To get one optimal shortest route, finding one local optimal result is enough for each subtree. However, to get more perfect possible routes, it needs to calculate more than one local optimal path. For example, if the threshold parameter value rs is equal to m, each subtree will calculate a local optimal path that varies the previous searched local optimal routes until the m time is reached. The local optimal results will be searched from each subtree in sequentially. Therefore, the time complexity will increase a little, when the number of subtree or local optimal results that defined in each subtree are increased. The processing time depends on local optimal results of each subtree in 10 times run can be seen in Figure 5.9. The proposed algorithm can perform to chance the two purposes with a little time complexity.

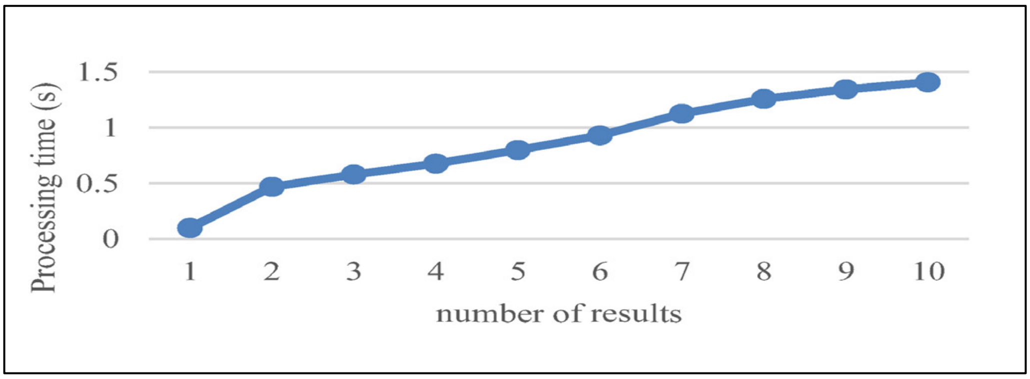


Figure 5. 9 The Variance of Time Complexity depends on the Number of Results of Each Subtree

5.2.1 Statistical Analysis of Undirected Road Network and Directed Hypergraph Network

The network architecture of the Yangon bus network is a complex hypergraph network and it is constructed with 84 bus lines and it contain huge number of nodes and edges. In addition, it needs to consider in direction for network. Each node owns many inwards and outwards edges. The proposed CFBHS algorithm, A* and Bi-directional Dijkstra algorithms are compared about the time complexity for Yangon downtown area. Figure 5.10 describes the average time complexity of proposed CFBHS algorithm is better than the popular A* although the proposed algorithm uses different threshold value. Moreover, Figure 5.11 depicts the total time complexity of different algorithms depends on different execution times. In Figure 5.12, the processing time is compared on different data size using proposed algorithm and the time complexity of proposed algorithm changes a little although the data size becomes large.

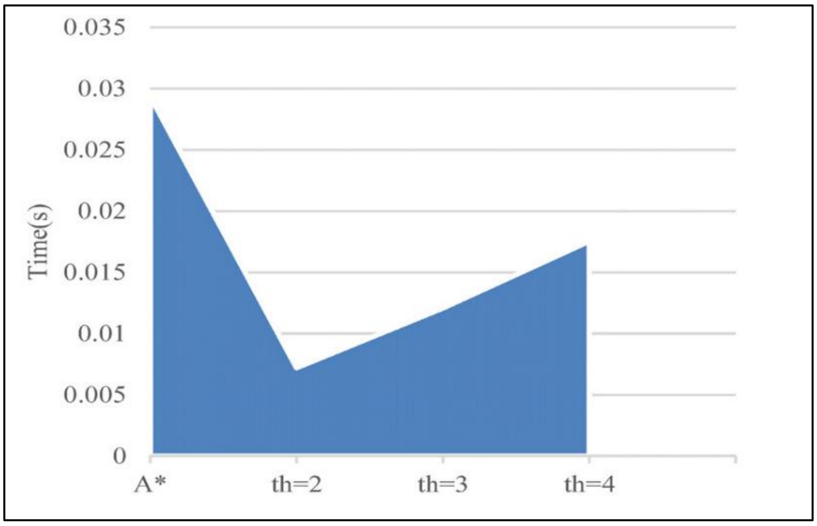


Figure 5. 10 Average Time Complexity for A* and Proposed Algorithm with Different Threshold Values

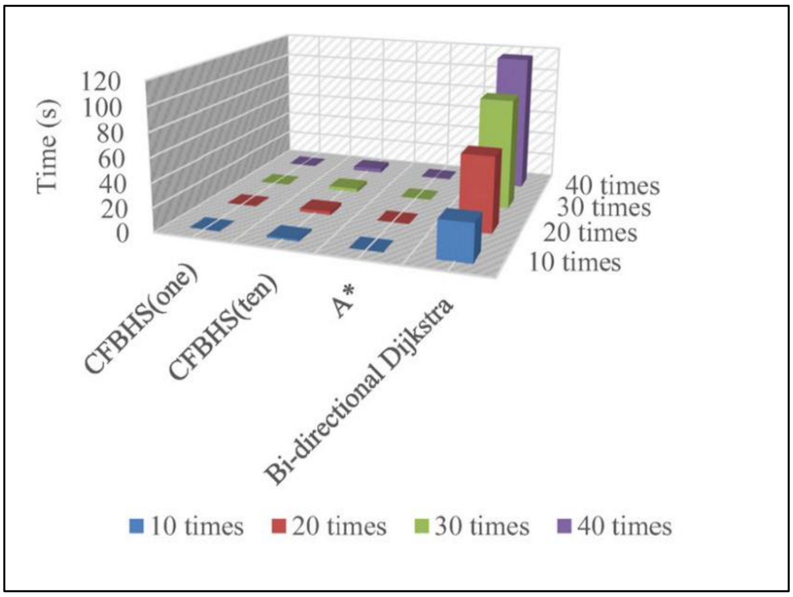


Figure 5. 11 Time Complexity among Different Algorithms depends on Execution Time

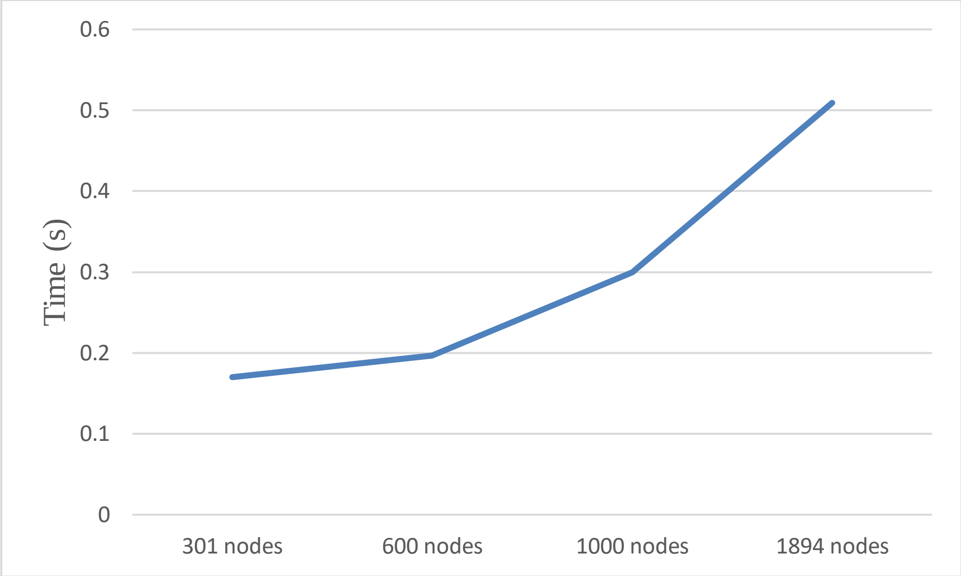
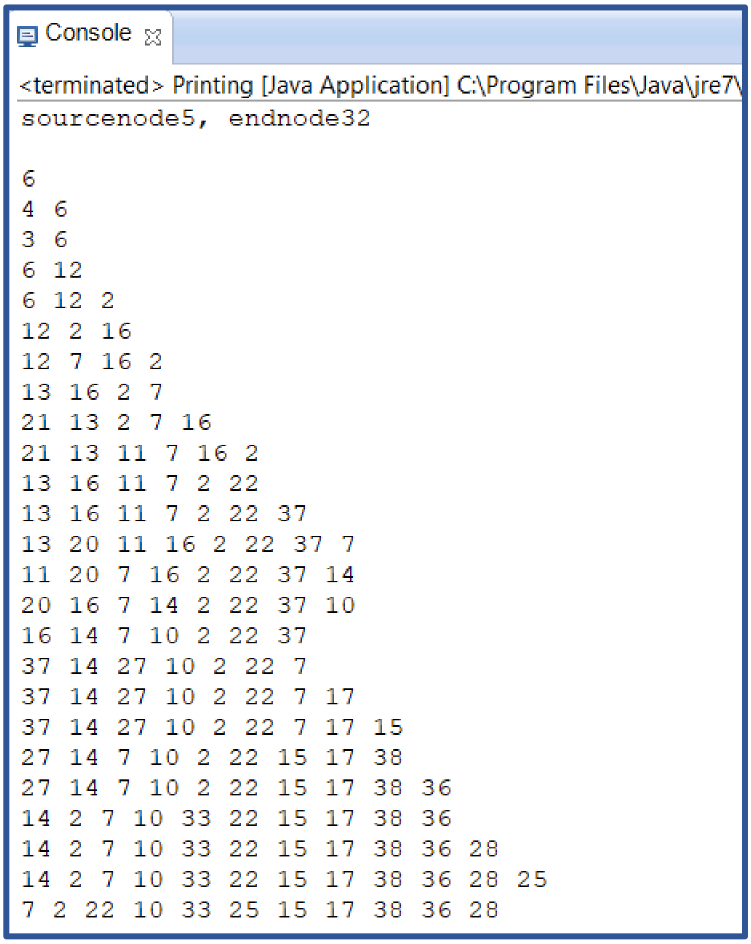


Figure 5. 12 Time Complexity of the Proposed Algorithm depends on Different Data Sizes

5.2.2 Average Number of Nodes that Need to Compare for Going the Next Level

Figure 5.13 describes the number of compared nodes while finding the short path between two nodes, node 5 and node 32 for the simulation graph that already depicts in section 5.1. The system counted the number of times that compared among of the nodes for going the next level to reach the destination. The number of times is 423 for start node 5 and target node 32. A* considers repeatedly the nodes in the open list. For example, A* algorithm checks the neighbor nodes of the current node 5. The neighbors of node 5 are node 4 and node 6. A* algorithm choses the node 4 and the remaining node 6 is saved in the open list.

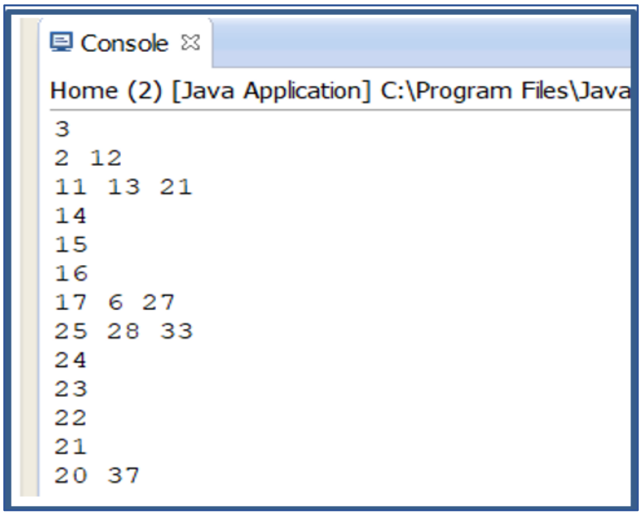
At the next line in Figure 5.13, it considers node 6 again. When the tree becomes deep, the number of nodes in the open list will be much. Then, the comparison times will be last. Figure 5.14 describes the situation of the nodes that need to compare using CFBHS. In this condition, it does not need to consider the node comparison phenomenon if the number of neighbors about current node exits only one. The number of compared nodes is 75 for forward direction and 51 is backward direction and the total compared nodes is 126 when the proposed CFBHS is used to search the shortest route. Figure 5.15 also describes the statistical analysis of compared nodes between CFBHS and A* algorithm. After analyzing the number of compared nodes, A* algorithm needs to compare more times than proposed CFBHS algorithm. The tree of compared nodes in A* algorithm will be more expanded when the data set is bigger.



```
Console
<terminated> Printing [Java Application] C:\Program Files\Java\jre7\
sourcenode5, endnode32

6
4 6
3 6
6 12
6 12 2
12 2 16
12 7 16 2
13 16 2 7
21 13 2 7 16
21 13 11 7 16 2
13 16 11 7 2 22
13 16 11 7 2 22 37
13 20 11 16 2 22 37 7
11 20 7 16 2 22 37 14
20 16 7 14 2 22 37 10
16 14 7 10 2 22 37
37 14 27 10 2 22 7
37 14 27 10 2 22 7 17
37 14 27 10 2 22 7 17 15
27 14 7 10 2 22 15 17 38
27 14 7 10 2 22 15 17 38 36
14 2 7 10 33 22 15 17 38 36
14 2 7 10 33 22 15 17 38 36 28
14 2 7 10 33 22 15 17 38 36 28 25
7 2 22 10 33 25 15 17 38 36 28
```

Figure 5. 13 Compared Nodes Using A* Algorithm



```
Console
Home (2) [Java Application] C:\Program Files\Java
3
2 12
11 13 21
14
15
16
17 6 27
25 28 33
24
23
22
21
20 37
```

Figure 5. 14 Compared Nodes Using CFBHS Algorithm

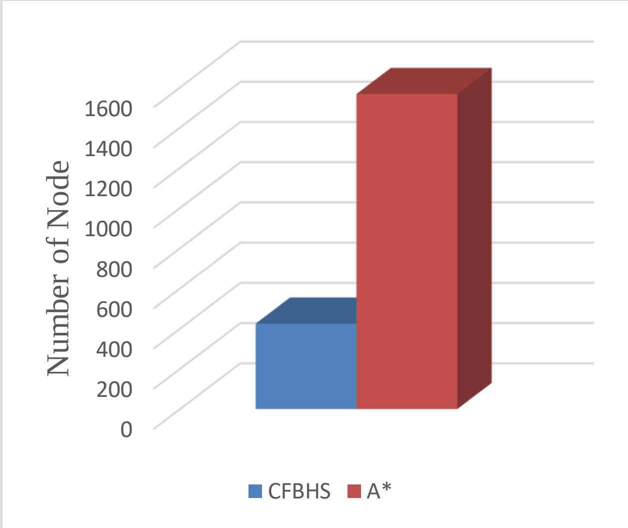


Figure 5. 15 Compared Nodes between CFBHS and A* during Five Times

5.3 Performance Analysis

The performance of proposed algorithm is measured using asymptotic analysis. It has computed the effectiveness of algorithm that measures the amount of taking time of a function after the input is given. It does not require to calculate depend on machine specific features such as processors, operating system and hardware. This section describes in two parts: detailed time complexity of proposed algorithm and A* and the next one is the comparison of performance of two algorithms for Yangon downtown area.

5.3.1 Performance Analysis of CFBHS Algorithm

The proposed algorithm is evaluated according to four main factors: forward and backward search, the assessment time for defining a selected node to go next level and the total number of subtrees. Let n be the total number of vertices of the network graph, k is the number of subtrees and m is the value that subtract one from maximum degree value. If the shortest route length is n, the comparison time while the next level is traversed along this route is mn. After that, the total number of compared nodes for one subtree are mnk. It does not need to consider for both level 1 and the other level that reach to target. Then, the current time complexity is $mk(n-k)$. For forward and backward direction, $2mk(n-k)$ will cost. Finally, quicksort algorithm is used to choose the best one from local optimal paths from each subtree. The time complexity of quicksort algorithm is $O(n)^2$, where n is the total number of paths that need to sort. After that, the time complexity of proposed algorithm is $O(mnk)$ as the worst case.

5.3.2 Performance Analysis of A* Algorithm

A* is the popular short finding algorithm and the time complexity is $O(b^d)$ and b is the branching factor and d is the depth. The total time complexity of A* is described as $\sum_{i=1}^{d-1} T_i$, where T is the time complexity for each step and d is the depth of path. In Figure 5.16, the total time complexity is 15 after the algorithm runs three steps. At step 1, the algorithm compares three nodes, then T (1) =3 and step 2 and step 3 is T (2) =5 and T (3) =7.

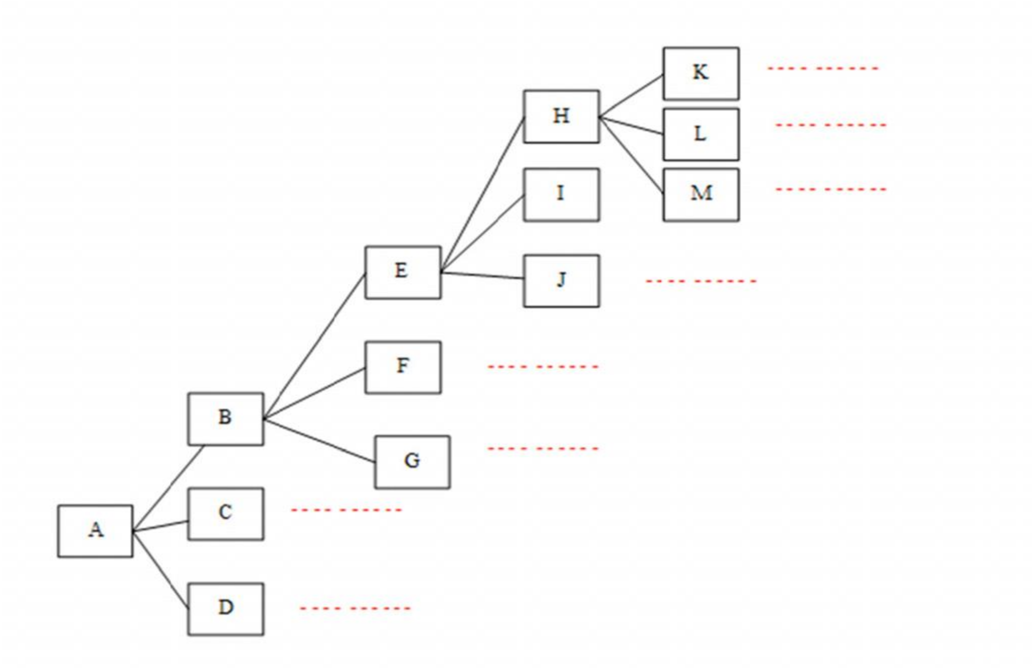


Figure 5. 16 Branching the Tree for Finding the Shortest Path between Node A and Node L

The time complexity of A* algorithm differs depends on the computed routes. However, when the route is long, this formula is not easy to calculate time complexity. Then, $O(b^d)$ is applied and then the time complexity is 81 as the worst case because b is 3 and the value of d is 4.

5.3.3 Performance Comparison between CFBHS and A * Algorithm

The performance of CFBHS and A* is compared using Yangon downtown data described in section 5.2. This graph contains 41 vertices and maximum degree is three and then m value is three. Subtree value k is calculated using degree-based threshold calculation method. The execution times of both algorithms are computed using asymptotic analysis. The time complexity of proposed CFBHS algorithm is depicted in Table 5.1.

Table 5.1 Detail Performance Analysis of CFBHS Algorithm

k	m	n	Checked nodes $(n-k)$	Execution time
2	3	41	39	476
3	3	41	38	702
4	3	41	37	920

The total time complexity of A* algorithm is 1680 for Yangon downtown data and the complexity is depicted as$\sum_{i=1}^{40} T_i$ .Therefore, the performance of proposed CFBHS algorithm is better than heuristic search A* according to asymptotic analysis. These two algorithms are comparing according to the worst case. Actually, these two algorithms are faster the other classical approaches according to the experiment results in real time.

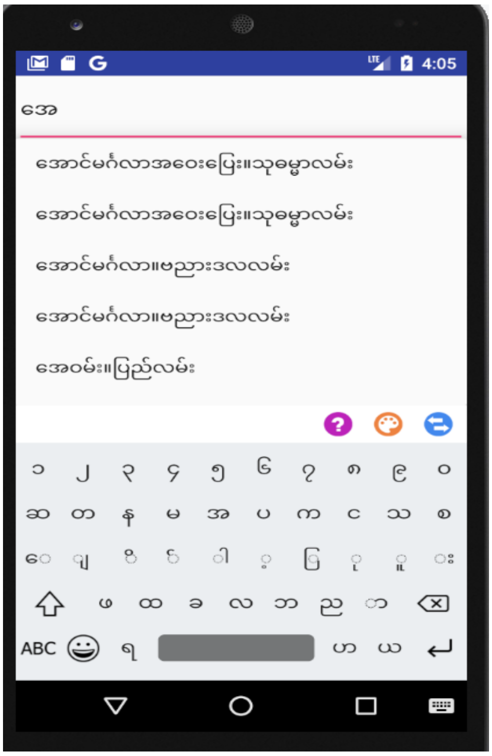
5.4 Implementation of Yangon Bus System using Proposed Algorithm

This system uses public bus data network of Yangon Bus System. The system applies client and server architecture. Localhost and port number "8092" is used to start the Tomcat server. Oracle database is used at the server side and bus stop names, latitude and longitude, road names, city name, bus line id, bus line number, point_id in English and Myanmar language are stored in this database. Spatial data such as latitude and longitude among of the fields are used and retrieved from database for constructing the network graph. Two tables are created, and the first one is storing bus information that includes both spatial data and non-spatial data and second table is storing the distance data among of the edges by calculating spatial latitude and longitude data using Haversine Distance Method. This method has already been described in Section 4.3.2.

5.4.1 User Input and Bus Stop Information

User can travel within 33 townships of Yangon using public bus without paying many costs. Over 100 bus lines are composed for travelling within the Yangon city. This system especially focuses on seven townships in Yangon and it contains 84 bus lines. These bus lines are constructed about 2000 bus stops and it includes many connections between the edges according to bus lines architecture. User can input bus stop names applying Myanmar or English Language as described in Figure 5.17(a). User's input bus stop names relate with latitude and longitude spatial data. The identified number of the bus stop names are defined by bus_id in database. The duplicated names of the bus stop are separated according to road name or township name. Auto Keyword Searching Technique is used in the phenomenon that accepts input data in textboxes as described in Figure 5.17(b). This supports the passengers to view easily the related bus stop names by typing only the initial word.

(a) Searching the Routes



(b)Keyword Search

Figure 5. 17 Data Input using Myanmar or English Language

The users can also search and view the spatial information of bus stops via on a map and this can be viewed in Figure 5.18.

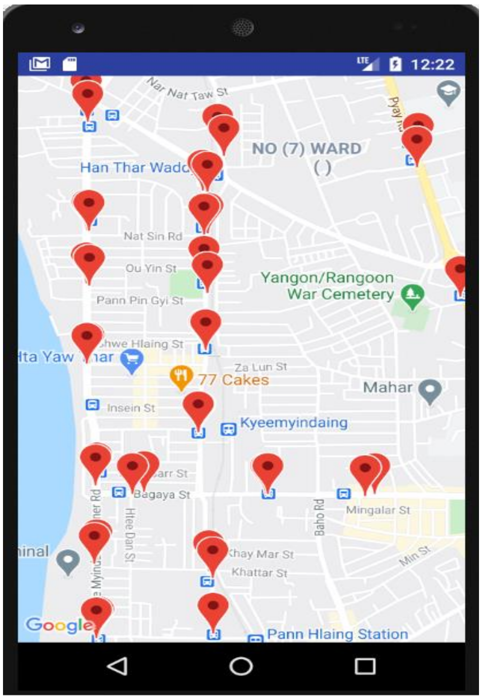
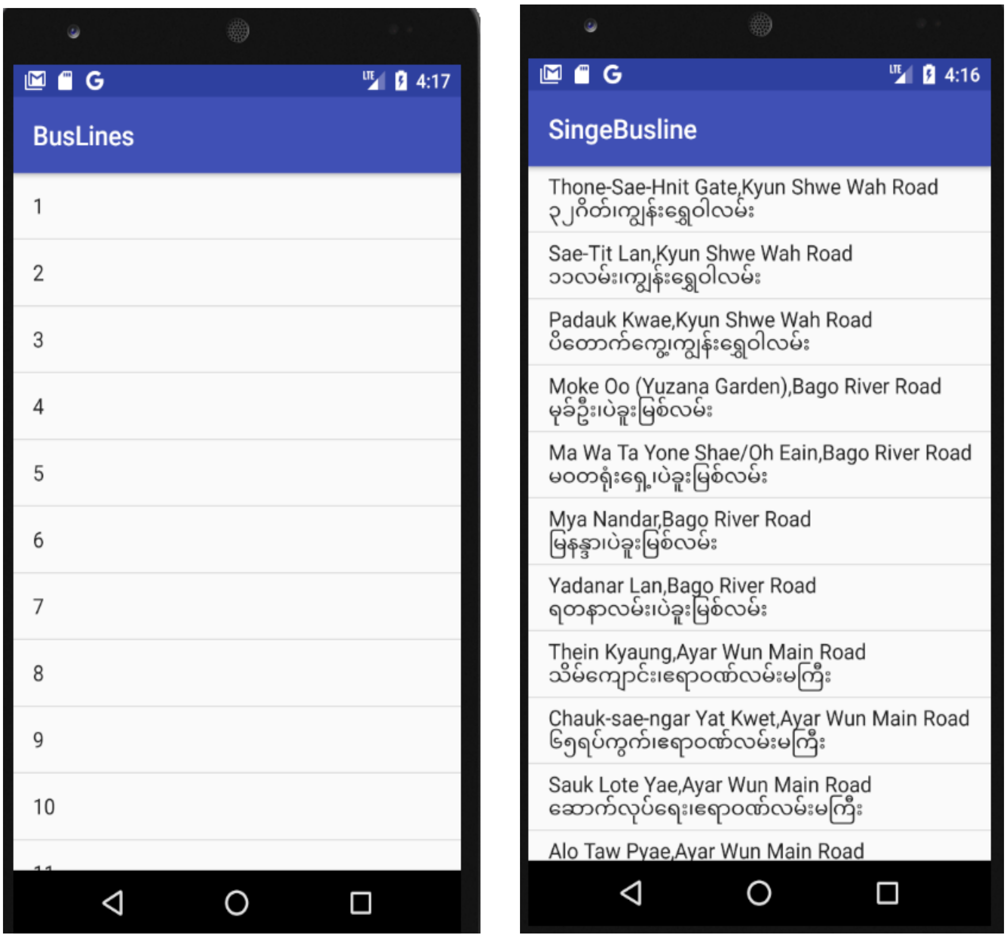


Figure 5. 18 Bus Stops in Yangon Bus System

In addition, the users can see on the map about the nearest bus stop from current location due to the Google map service is applied in this system. Moreover, the system also provides the detailed bus stop information of each bus line as the following Figure 5.19. Data is collected from the Google Earth and some are composed of using popular spatial data collecting device named Garmin Device. YRTA also support required bus line information of buses passing within Yangon Region. Yangon Public bus network contains over 1894 bus stops and more than one bus line is passing on each. Let assume one bus stop as a node and then this node owns many edges due to multiple bus lines are passing.

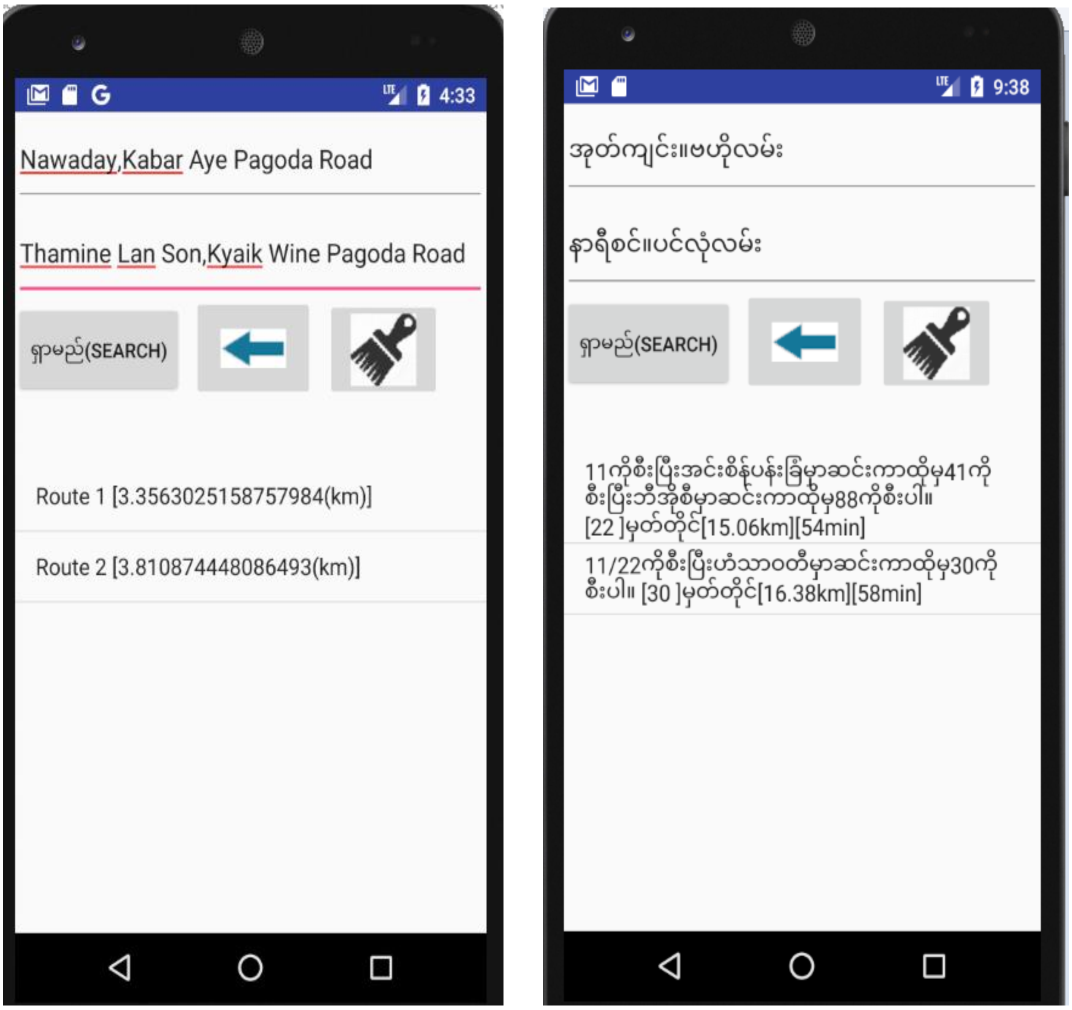


(a)All Bus Lines (b)Bus Stop Information of Single Bus Line

Figure 5. 19 Bus Lines and Bus Stop Information of Each Bus Line in Yangon Bus System

5.4.2 Finding the Routes between Start and End Bus Stops

User can search the routes by clicking the English Language Routes Menu that already described in Figure 5.17(a). Then**,** user can see two shortest routes as described in Figure 5.20(a) after start and end bus stops are inserted in the textbox. And then, each route can be clicked to view the results on the map and user can see as described in Figure 5.21. According to this way, user can also search the bus numbers and the system will produce the bus numbers that need to take as described in Figure 5.20(b). According to the YRTA's new bus line running plan, to reach the desired location, the user cannot take many bus numbers nowadays. Generally, two transit bus numbers are at most needed if the travel is long.



(a) Finding Routes **(b)**Finding Bus Numbers

Figure 5. 20 Finding Routes and Bus Numbers in Yangon Bus System

Although the users can search the routes and bus numbers using the system, some people do not know the bus stop names exactly. For this problem, they are difficult to use the public bus system. The system also supports to assign the alarm system. For the alarm sound, ring tone that included in setting of android phone system is used. Android platform provisions “Ringtone Manager” to perceive the song of ring tone for the people. Users can define destination bus stop name that want to ring alarm song and this song will be played when the assigned bus stop is reached. User can assign bus stop name by clicking “Alarm” menu in home page as described in Figure 5.22. Location based service and GPS technology is used for giving this facility for users who apply this application.

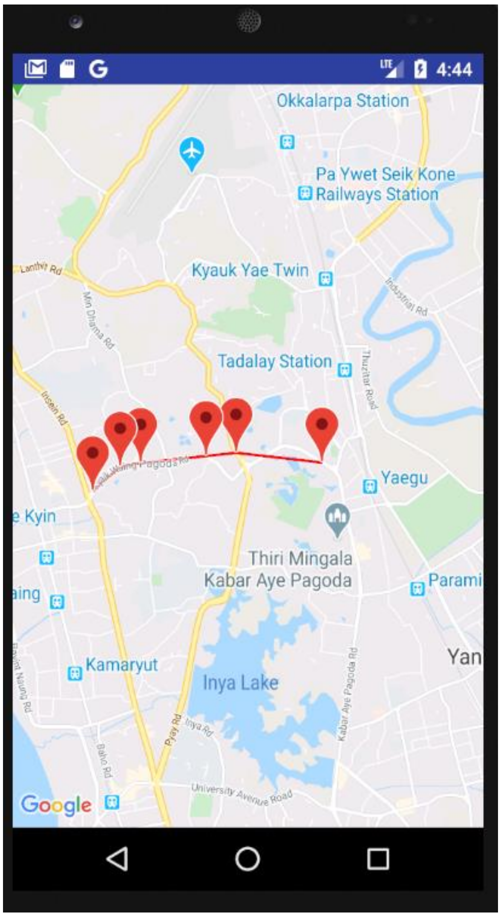


Figure 5. 21 Displaying the Route on Google Map

Figure 5. 22 Home Page of System

5.5 Subjective Evaluation for Proposed System

The proposed system measured not only objective evaluation such as asymptotic analysis and empirical analysis but also subjective evaluation. Thirty people who live in Yangon Region participated in this survey. Fifteen inputs were used for evaluation measures. Among inputs, four main groups are specified in evaluation measures. The first group is measuring the accuracy of the shortest routes for different bus lines' colors: brown, red, blue, purple, orange and green. Each district area is represented by different colors. The percentage of the accuracy is about 98 percent according to criteria such as unsatisfied, satisfied, neutral, very unsatisfied and very satisfied. The second is to know the users' desire about measurement units for searching the shortest routes. The system gives the information about the shortest route depending on distance. The users were asked whether they liked to decide the shortest route depending on distance or on the number of bus stops. Ninety percent of the users like to decide the shortest route depending on the distance. The third one is considering the processing time about searching the shortest routes and bus numbers for those routes.

The last one is considering the voice quality about the alarm system. All users like including the alarm service but 20 percent of the users answered voice is soft. Figure 5.23 describes the condition about satisfaction of the users for the accuracy of each bus line' color. Therefore, the correctness for searching the shortest route is satisfied by 98 percent of the users.

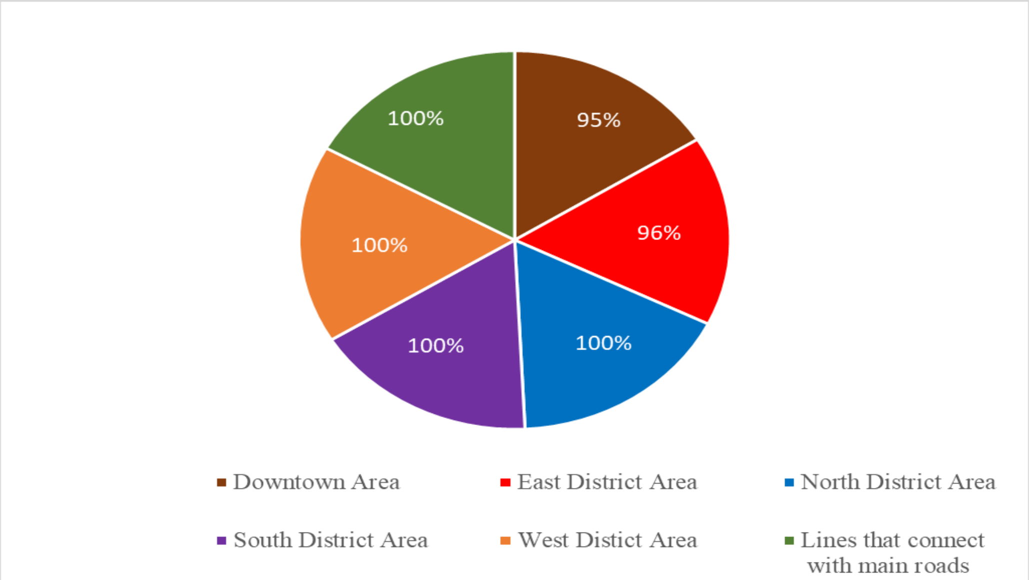


Figure 5.23 Users' Satisfaction in Different Bus Lines' Color

Figure 5.24 also describes the condition about satisfaction of the users for the processing time of searching the shortest route and bus number. When 10 users searched the shortest route, 9 users responded that they could find it in the short time and 7 out of 10 participants said that they could find the bus numbers for these routes in a short time. Therefore, in the proposed system, the findings of the subjective evaluation matched with those of the objective evaluation because the shortest routes can be searched in the short processing time.

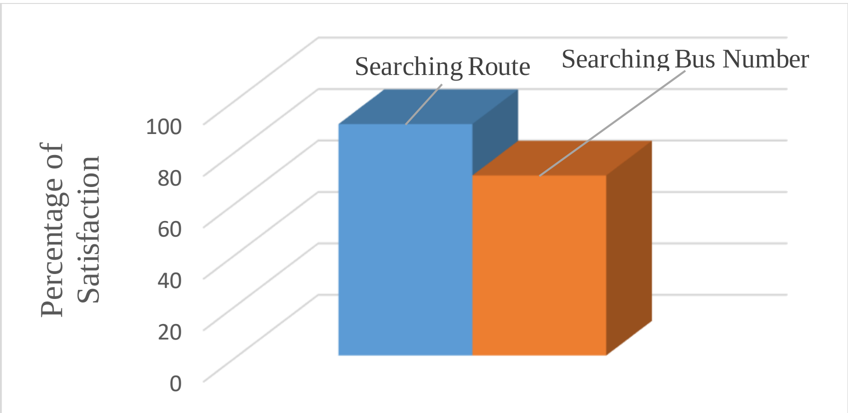


Figure 5.24 Users' Satisfaction about the Processing Time

5.6 Summary

This chapter focuses on the experimental results of the proposed CFBHS algorithm using two different networks: road network and bus network. Different Evaluation Metrics are used for the proposed algorithm. These are statistical analysis, empirical analysis and asymptotic analysis. Firstly, simulation graph for road network is constructed using J2SE technology to be easy for checking the accuracy of the proposed algorithm. Junction points are used as the nodes of the graph for road network. Two shortest results can be viewed on the simulation graph. Road network does not consider direction and it is used as the undirected graph. Cache technology is used to maintain and get the previous information between subtrees and the queries of the application's users. It considers only on the neighbor nodes of the current node for choosing the node that will go to next level to reach the target node. The proposed algorithm applies the concept of bi-directional search, heuristic search, divide and conquer technique and sorting algorithms. According to the experimental results and evaluations, the proposed algorithm is faster than the A* and Bi-directional Dijkstra. The system also compares the execution time depend on different data sizes using the data of public bus network. Although the data size is large, time complexity increases a little according to experimental results. Therefore, this algorithm is used to implement the Proposed Yangon Bus System. Yangon Bus Network is directed graph and it is complex because it is the structure of Hypergraph Network. After finding the two shortest result for Yangon Bus Network depend on user query, the system also calculates the transit bus numbers for these results using the Bus Number Calculation Equations in the previous chapter. Moreover, subjective evaluation is made for the proposed system. According to the asymptotic analysis, statistical analysis, empirical analysis and subjective evaluation, the system can search the shortest route in a short time. Then, chapter 6 will conclude the thesis and future research works.

CHAPTER 6

CONCLUSION AND FURTHER EXTENSION

The system has emphasized on finding shortest and possible paths simultaneously. Two networks are used for testing the proposed technique. The proposed heuristic based algorithm called Combined Forward and Backward Heuristic Search (CFBHS) is applied for Yangon Downtown Network and Yangon Bus Network. Popular Heuristic Search Algorithm A*, Bi-directional Dijkstra algorithm based on mature classical short finding algorithm and proposed CFBHS algorithm are compared according to statistical, empirical results and asymptotic analysis. According to the experimental results, the proposed technique produces the results in a short time and it is faster than other popular short finding algorithms. While A* and Bi-directional Dijkstra Algorithm can produce the single result, the proposed algorithm can get not only the shortest but also possible paths concurrently.

According to the network nature, two features such as static and dynamic about the network graph can be defined. Although the static network can prepare the required information about the geospatial distance values between two nodes before the processing is made, dynamic network needs to calculate the distance values dynamically during the processing time because the distance values can change according to network nature. The proposed algorithm finds the heuristic values dynamically while the shortest route is searching without pre-storing the estimated distance values. It does not require a long pre-processing time for getting the heuristic information. Moreover, it does not need to consider all nodes in the open list and it can reduce duplicated node considering problems while choosing a node of the next level to reach the goal. It considers the neighbor nodes of the current node while it traversing the graph.

The system develops two networks: the first one is Yangon Downtown Road Network and the second is Yangon Bus Network. The junctions of the road are considered as nodes and the simulated graph is developed using Java Standard Edition (J2SE) programming language. The shortest and possible paths can be easily traced whether the output results are accurate or not on the simulated graph. Yangon bus network dataset is used for finding the two shortest result and this is developed using client and server architecture. After finding the shortest results, the system produces the bus numbers for each route using transit algorithm. Yangon bus network is directed graph and it needs to consider two ways for the road. Bus stops are assigned as the nodes and these bus stops are passed by different bus lines. Each shortest route can be viewed on google map of mobile phone using Google API technology.

The system deliberated many related works such as Partitioning Techniques, Shortest Path Algorithms and Time Comparison among shortest path algorithms. There are many shortest path algorithms. These are Genetic Algorithm, Classical Mature Shortest Path Algorithms, Heuristic Search Algorithms, Forward Changing and Backward Changing Strategies, Tree or Graph Traversing Algorithms such Breadth First search, Depth First search, Best-First Search, Uniform Cost Search and Hill Climbing Search. Google Play Services, Google Play API, Location Based Services, Google Earth and Open Street Map are used in this system for collecting geospatial data and to show the spatial data on google map.

This system also described the detailed works of the shortest path algorithms: Dijkstra Algorithm, Bi-directional Dijkstra Algorithm and A* Heuristic Search Algorithm. There are many distance formulae such as Manhattan Distance, Diagonal Distance and Euclidean Distance in the heuristic search. Among of these distance formulae, the system used Euclidean Distance formula because it is the shortest time for calculating the distance between two nodes. In addition, the expertise according to the reviews and the works of these algorithms are discussed in this system. It also discusses the spatial data of Yangon Bus Network, Data Storage and detailed work of the proposed CFBHS algorithm together with working example and pseudo code.

The experimental results and evaluations according to statistical analysis, empirical analysis and asymptotic analysis are designated in this system. In addition, the people can use the proposed system on android mobile phone. The subjective evaluation was made for the proposed system. The system produced the two shortest routes together with bus numbers. These results can be viewed on the google map and users can use two languages: Myanmar and English. Results are described using three units: kilometers, minutes and number of bus stops. Keyword search technique for both two languages: Myanmar and English can be applied in this system. SAX parser was used instead of DOM parser in the proposed system because SAX parser used event-based API. This parser cannot parse the whole document to retrieve the bus information from XML file. The system gave the alert message when the destination bus stop is reached. Location based services, Google Play services and GPS technology is used for giving the alarm message. Users can use the alarm service as offline service but they need to turn on GPS on their android mobile phone. So that, the system can catch up-to-date spatial point such as latitude and longitude when the mobile phone's user is moved.

6.1 Limitations

There are some limitations in the implementation of the system. 84 bus lines are applied for Yangon Bus Network. Local Java 2 Enterprise Edition (J2EE) tomcat server is used. The weight of the links is considered as distance in static. Some bus stop names are stored by pronunciation (Myanmar language) in Myanmar and it can be a little difficult for foreigners to understand. Bus network architecture is changed in 2017 and it is not fixed certainly in current. It can also occur bus line and travel cost changes. To get the updated information about bus lines and bus management is needed because YRTA can change bus architecture in present situation.

6.2 Further Extension

The system proposed a new short path finding algorithm. It uses the divide and conquer approach. The Large tree is divided into small subtrees. The local optimal research has been searched from each subtree and finally global result is searched using quick sort algorithm. Each subtree works separately at finding the results and each run the processes one after another. The proposed algorithm can modify in future using parallel processing technique to get better performance than efficiency in current. Moreover, cloud server can be used instead of using the local server. Public bus transportation system developed on mobile phone, especially focuses to find two shortest routes and to give bus information about direct and transit ways for each route. As a future job, other factor can be considered while giving the bus information. These are the waiting time at bus stop, the time to walk to reach the bus stop and the arriving and departure time of the bus.

LIST OF ACRONYMS

A*	A Star Algorithm
ADV	Android Virtual Device
API	Application Programming Interface
ArcGIS	Arc Geographical Information System
ASP	Active Server Pages
CFBHS	Combined Forward and Backward Heuristic Search Algorithm
DSTAP	Decomposition Approach for Traffic Assignment Problem
ECOP	Earth-Centered Orbital Planes
ESRI	Environmental Systems Research Institute
GCS	Geographic Coordinate System
GIS	Geographical Information System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
iOS	iPhone Operating System
J2EE	Java 2 Enterprise Edition
J2SE	Java 2 Standard Edition
JSON	JavaScript Object Notation
KD	K Dimensional
KML	Keyhole Markup Language
LAMP	Linux Operating System, Apache Web Server, My Sequel database and Web programming language
NAD	North American Datum

OR	Operations Research
OSM	Open Street Map
PCS	Project Coordinate System
QGIS	Quantum Geographical Information System
RF	Radio Frequency
SDK	Software Development Toolkit
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
XML	Extensible Markup Language
YRTA	Yangon Region Transport Authority

AUTHOR'S PUBLICATIONS

[P1] Wai Mar Hlaing and Myint Myint Sein, University of Computer Studies, Yangon Myanmar, "Search Space Reduction using K-means Clustering and Adjacency matrices for GIS Usage Information Retrieval**",** The Seventh International Conference on Science and Engineering (**ICSE 2016**), Yangon Technological University, Yangon, Myanmar, December 9-10, 2016.

[P2] Wai Mar Hlaing and Myint Myint Sein, University of Computer Studies, Yangon Myanmar, "Removing Unnecessary Map-based Data for Search Space Reduction", 15th International Conference on Computer Applications (**ICCA 2017**), pp. 266-271, Yangon, Myanmar, February 16-17, 2017.

[P3] Wai Mar Hlaing and Myint Myint Sein, University of Computer Studies, Yangon Myanmar, "K-means Nearest Point Search Algorithm and Heuristic Search for Transportation", 7th Global Conference on Consumer Electronics (**GCCE 2018**), pp. 779-780, Japan, October 9-12, 2018, ISBN: 978-1-5386-6309-7, ISSN:2378-8143, DOI: 10.1109/GCCE.2018.8574798, #**IEEE**, SCI Conference.

[P4] Wai Mar Hlaing[1, 2], Shi-Jian Liu[2] and [2, 3]Jeng-Shyang Pan, [1]University of Computer Studies, Yangon Myanmar, [2]Fujian University of Technology, China, [3]Shandong University of Science and Technology, China, "A Novel Solution for Simultaneously Finding the Shortest and Possible Paths in Complex Networks", Journal of Internet Technology (**JIT**), Taiwan. Vol. 20, No. 6, pp. 1693-1707, November 2019, ISSN: 1607-9264, DOI: 10.3966/160792642019102006002, **(Scimago index – Q3)**

[P5] Wai Mar Hlaing and Myint Myint Sein, University of Computer Studies, Yangon Myanmar, "Neighbor Search with Hash Map Indexing Technique for Complex Networks", The 12th International Conference on Future Computer and Communication (**ICFCC 2020**), Proceedings of the 10th International Workshop on Computer Science and Engineering (**WCSE 2020**) , Science and Engineering Institute, USA, co-organized by Yangon (Rangoon), Myanmar (Burma), pp. 11-15,February 26- 28, 2020, ISBN: 978-981-14-4787-7, **(Ei & Scopus)**

BIBLIOGRAPHY

[1] S. Ahmed, R. F. Ibrahim, H. A. Hefny, "GIS-Based Network Analysis for the Roads Networks of the Greater Cairo Area", CEUR Workshop Proceedings, July 2017.

[2] R. K. Ahuja, K. Mehlhorn, J. Orlin and R. E. Tarjan, "Faster Algorithms for the Shortest Path Problem", Journal of the ACM, vol. 37, no. 2, pp. 213-223, April 1990.

[3] A.G. Akpan, "GIS – Based Participatory Crime Prevention: Awareness of Community on Crime Mapping and Hot spots", Journal of Scientific and Engineering Research, pp. 233-238, September 2018.

[4] A. Al-Ajlan, "The Comparison between Forward and Backward Chaining", International Journal of Machine Learning and Computing, vol. 5, no. 2, pp. 106–113, 2015.

[5] M. Arseni, A. Rosu, C. Bocăneală, D.E. Constantin and L. P. Georgescu "Flood hazard monitoring using GIS and remote sensing observations", Carpathian Journal of Earth and Environmental Sciences, 2017, pp. 329–334.

[6] J. L. Bander, C. C. White, "A Heuristic Search Algorithm for Path Determination with Learning", IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 28, no. 1, pp. 131-134, January 1998.

[7] H. Bast and I. Weber, "Partitioning Graphs to Speed Up Dijkstra's Algorithm", Experimental and Efficient Algorithms, 4th International Workshop, pp. 189-202, July 2014.

[8] S. Behzadi, A. A. Alesheikh and E. Poorazizi, "Developing a genetic algorithm to solve shortest path problem on a raster data model". Journal of Applied Sciences, vol. 8, no. 18, pp. 3289–3293, 2008.

[9] A. Bhandari, A. Panwar, P. Saklani, "Creation of a Web Based GIS Application for Spatial Wasteland Information System for Uttarakhand State Using Effective Open Source Software", vol. 7, no. 6, pp. 2405–2408, 2016.

[10] V. Bhanumurthy, V. M. Bothale, B. Kumar, N. Urkude and R. Shukla, Route "Analysis for Decesion Support System in Emergency Management through GIS Technologies", International Journal of Advanced Engineering and Global Technology, vol. 3, no. 2, pp. 345-350, February 2015.

[11] M. A. Brovelli, M. Minghini, M. E. Molinari and G. Zamboni, "Positional accuracy assessment of the openstreetmap buildings layer through automatic homologous pairs detection", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives), pp. 615–620, July 2016.

[12] I. Chabini, S. Lan, "Adaptations of the A* Algorithm for the Computation of Fastest Paths in Deterministic Discrete-Time Dynamic Networks", IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, pp. 60-74, March 2002.

[13] B. V. Cherkassky, A. V. Goldberg and C. Silverstein, "Buckets, Heaps, Lists, and Monotone Priority Queues", SIAM Journal on Computing, vol. 28, no. 4, pp. 1326-1346, January 1999.

[14] A. Dabhade, K. V. Kale, Y. Gedam, "Network Analysis for Finding Shortest Path in Hospital Information System", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 7, pp. 618–623, 2015.

[15] N. Danilina, I. Teplova, "Traffic flow organization in urban transport transit hubs", IOP Conference Series: Materials Science and Engineering, vol. 365, no. 2, 2018.

[16] E. Dere, A. Durdu, "Usage of the A * Algorithm to Find the Shortest Path in Transportation Systems", June 2018.

[17] A. Deshpande, "Providing emergency services using location based tracking on mobile devices", International Journal of Science, Engineering and Technology Research, Vol. 3, Issue 3, pp. 523-529, March 2014.

[18] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, vol. 1, no. 1, pp. 269-271, June 1959.

[19] M. Dramski, “Bi-directional Search in Route Planning in Navigation”, Scientific Journals of the Maritime University of Szczecin, vol. 39, no. 111, pp. 57-62, April 2014.

[20] F. Feng, “Learning on Partial-Order Hypergraphs”, The 2018 Web Conference, pp. 23–27, April 2018.

[21] A. Garcia, O. Arbelaitz, M. T. Linaza, P. Vansteenwegen and W. Souffriau, “Personalized tourist route generation”, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 486–497, 2010.

[22] A. Goyal, P. Mogha, R. Luthra, N. Sangwan, “Path Finding: A* or Dijkstra’s?” International Journal in IT and Engineering, vol. 2, no. 1, January 2014.

[23] V. Guihaire, J. K. Hao, “Transit network design and scheduling”, Transportation Research Part A: Policy and Practice, vol. 42, no. 10, pp. 1251–1273, 2008.

[24] C. Guo, Y. Zhen, G. Jia, X. Zhou, “Multilayer Satellite Network Routing based on Hypergraph Theory”, Cyberspace Technology (CCT 2013), vol. 7, pp. 5–8, 2013.

[25] S. Guze, “Graph Theory Approach to Transportation Systems Design and Optimization”, The International Journal on Marine Navigation and Safety of Sea Transportation, vol. 8, no. 4, pp. 571-578, 2015.

[26] Y. Huang, Q. Yi, M. Shi, “An Improved Dijkstra Shortest Path Algorithm”, 2nd International Conference on Computer Science and Electronics Engineering, pp. 226-229, Paris, France, 2013.

[27] T.Hultman, A. Boudjadar, M. Asplund, “Connectivity-optimal shortest paths using crowdsourced data”, International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 1–6, 2016.

[28] P. Jaramillo-Alvarez, C. González-Calderon and G. González-Calderon, “Route Optimization of Urban Public Transportation”, vol. 80, no.180, pp. 41–49, 2013.

[29] A. Jeyaraj, “Intelligent Transport Route Planning Using Genetic Algorithms in Path Computation Algorithms”, May 2017.

[30] J. Jokar Arsanjani, A. Zipf, P. Mooney, M. Helbich, “An Introduction to OpenStreetMap in Geographic Information Science: Experiences, Research, and Applications”, Lecture Notes in Geoinformation and Cartography, pp. 1-15, 2015.

[31] R.J. Jyothi, V.R. Prasad and N.A. N Anuradha, “Automatic Accident Detection and Ambulance Rescue With Intelligent Traffic Light System”, International Journal of Scientific Research, vol.3, no. 7, pp. 177–179, 2014.

[32] C. Kinyua and C. K. Gitonga, “Prims Algorithm and its Application in the Design of University LAN Networks”, International Journal of Advance Research, April 2016.

[33] A. Krause, “Bounded Treewidth Graphs-A Survey”, pp. 1-9, 2003.

[34] U. Lauther and S. Ag, “An Extremely Fast, Exact Algorithm for Finding Shortest Paths in Static Networks with Geographical Background”, pp. 219-230

[35] E. Lepuschitz, “Geographic Information Systems in Mountain Risk and Disaster Management”, Applied Geography, vol. 63, pp. 212-219, September 2015.

[36] E. Lepuschitz, “Geographic Information Systems in Mountain Risk and Disaster Management”, Applied Geography, vol. 63, pp. 212-219, September, 2015.

[37] X. Liu, D. Gong, “A Comparative Study of A-Star Algorithms for Search and Rescue in Perfect Maze”, In: Proc. International Conference on Electric Information and Control Engineering (ICEICE), pp. 24-27, Wuhan, China, 2011.

[38] M. S. Minal Mahure, “Place Reminder-An Android APP”, International Journal of Computer Science and Mobile Computing, vol. 4, no .1, pp. 192-200, 2015.

[39] R. H. Möhring, H. Schilling, B. Schütz, D. Wagner, T. Willhalm, “Partitioning Graphs to Speedup Dijkstra’s Algorithm”, in: 4th International Workshop of Efficient and Experimental and Efficient Algorithms (WEA), pp. 1-12, Santorini Island, Greece, 2005.

[40] P. Mubi, A. Planning, R. Polytechnic and A. F. State, “Spatial Analysis and Mapping of Crime in Mubi Metropolis Using Remote Sensing and GIS”, 2018, pp. 1-12.

[41] R. S. Parmar, B. Trivedi, “Vectored Approach to Create Highly Optimized Network for Road Traffic”, International Journal of Traffic and Transportation Engineering, vol. 5, no. 1, pp. 1-9, May 2016.

[42] M. Petrelli, “A transit network design model for urban areas”, WIT Transactions on the Built Environment, pp. 163–172, 2004.

[43] I. N. Piarsa, E. S. Hadi, N. Kadek and A. Wirdiani, “Rural Road Mapping Geographic Information System Using Mobile Android”, vol.12, no. 3, pp. 95–100, 2015.

[44] A. Pinandito, A. P. Kharisma and R. S. Perdana, “Framework design for map-based navigation in google android platform”, Journal of Telecommunication, Electronic and Computer Engineering, pp. 35–40, 2018.

[45] M. A. Qureshi, F. B. Hassan, S. Safdar, R. Akbar, “A O(|E|) Time Shortest Path Algorithm for Non-negative Weighted Undirected Graphs”, International Journal of Computer Science and Information Security, vol. 6, no. 1, pp. 40-46, October 2009.

[46] Ramadiani, D. Bukhori, Azainil and N. Dengen, “Floyd-warshall algorithm to determine the shortest path based on android”, Institute of Physics (IOP) Conference Series: Earth and Environmental Science, vol. 144, no. 1, 2018.

[47] S. Sadhu, S. K. Singh and S. Kanga, “Optimum path from tourist places to railway station and hospitals & fire services using GIS-based network analysis”, International Journal of Technology Research and Management, vol. 4, no. 8, pp. 1–5, 2017.

[48] J. Samual, "Implementation of GPS Based Object Location and Route Tracking on Android Device", International Journal of Information System and Engineering, vol. 3, no. 2, pp. 61-72, 2015.

[49] S. Sarkar, "Ambulance Assistance for Emergency Services Using GPS Navigation", International Research Journal of Engineering and Technology, vol. 3, no. 9, 1328–1331, 2016.

[50] S. K. Sharma and B. L. Pal, "Shortest Path Searching for Road Network using A * Algorithm," Int. J. Comput. Sci. Mob. Comput., vol. 4, no. 7, pp. 513–522, 2015.

[51] M. Singhal and A. Shukla, "Implementation of Location based Services in Android using GPS and Web Services", vol. 9, no. 1, pp. 237-242, January, 2012.

[52] B. Siregar, E. B. Nababan, J. A. Rumahorbo, U.Andayani, and F. Fahmi, "Nearby Search Indekos Based Android Using A Star (A∗) Algorithm," J. Phys. Conf. Ser., vol. 978, no. 1, 2018.

[53] B. V. Sowmya Shree, L. S. Krishna and Y. Rangaswamy, "Smart Detecting Accident by IoT and Smart Rescue System", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 7, Issue 4, pp. 288–295, April 2018.

[54] M. P. Uddin, M. Z. Islam, M. Nadim and M. I. Afjal, "GPS-based Location Tracking System via Android Device", International Journal of Research in Computer Engineering & Electronics, vol. 5, pp. 1–7, November 2013.

[55] R. van Bevern, I. Kanj, C. Komusiewicz, R. Niedermeier and M. Sorge, "Twins in subdivision drawings of hypergraphs", Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 67–80, 2016.

[56] R. van Nes, R. Hamerslag and B. H. Immers, "Design of public transport networks", Transportation Research Record", pp. 74-83, 1988.

[57] S. Vanjire, U. Kanchan, G. Shitole and P. Patil, "Location Based Services on Smart Phone through the Android Application", International Journal of

Advanced Research in Computer and Communication, vol. 3, no. 1, pp. 4982–4987, 2014.

[58] I. Witten, E. Frank, M. Hall, M. A. Hall, “Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)”, Morgan Kaufmann, 2011.

[59] Q. Wu and J. Hartley, “Accommodating user preferences in the optimization of public transport travel”, International Journal of Simulation: Systems, Science and Technology, vol. 5, no. 3, pp. 12–25, 2004.

[60] C. N. Yahia, V. Pandey and S. D. Boyles, “Network Partitioning Algorithms for Solving the Traffic Assignment Problem using a Decomposition Approach”, Transportation Research Record, vol. 2672, no. 48, pp. 116–126.

[61] R. Zhou, “Combining Breadth-First and Depth-First Strategies in Searching for Treewidth”, 2007.