

# TRAFFIC ROUTING ALGORITHM FOR ROAD NETWORK

**Mhnd FARHAN**  
mhndfarhan@yahoo.com

UNIVERSITY OF BAGHDAD, IRAQ

## ABSTRACT

*Nobody in this world wants to be held hostage in a traffic jam for hours and hours. The need to have efficient traffic flow in our roads cannot be overemphasized. It is in this understanding that this paper delves into the study and implementation of a suitable algorithm for traffic routing. The implementation is based on graph theory, where a road network map is first converted to a graph with nodes and edges where the edges represent the roads and the nodes represent the various road intersections. The roads are given various weights depending on their distance and traffic density, and the time a motorist would take to move from one point to another is estimated. A suitable routing algorithm, which in this case is the Dijkstra's algorithm is applied on the weighted graph.*

## KEYWORDS:

*Routing algorithm, Dijkstra, road network, traffic jam, efficient traffic flow*

## 1. Introduction

The need for efficiency and reliability in life has led human beings to develop technology that will see this achieved. The transport sector has not been left behind, thus engineers continue to brainstorm and test systems that will ensure efficiency on the roads. However, this cannot be achieved without proper data collection and processing using suitable algorithms. The availability of real time data round the clock of the traffic density on the roads will greatly ease traffic congestion, increase accessibility of places and henceforth improve the economy of the nation (Mu, Yu & Liu, 2009).

Traffic congestion is a real problem facing people everyday especially those that live in major cities. Critically analyzing this problem, a lot of time and money is lost in the traffic snarl-ups, therefore greatly contributing to the stunted economic growth in the country. As a result, many people prefer to wake up way early and start to travel at the wee hours when there are less motorists on the road. However, the evening traffic still catches up with them as they go back to their places of residence after work. Most of these people get home late due to the traffic and adding to the fatigue of the people, they are quite unproductive in the evening, thus, they only get rest and prepare

for the next day's work. As a result, a reliable and efficient system is necessary to route vehicles to roads with the less dense traffic. This system must provide the precise information needed to improve the traffic situation (Nha, Djahel & Murphy, 2012).

The system to be implemented ought to be able to determine the traffic density and to determine the optimal routes from a certain origin to a given destination. A suitable routing algorithm is to be employed. The shortest path between two vertices in a road map is to be identified. Why is it necessary to develop such an algorithm? The fact is that efficiency is critical in everything that we do. If we can have a program that gives us the best route at a given time from one point to a given destination, then it would save us much time and fuel to move from one point to the other. This paper, therefore, is much relevant to the world today (Agafonov & Myasnikov, 2017).

The area under study is graph theory. This is an area of study that deals with the study of graphs. Graphs are mathematical structures that are useful in modeling pairwise relations of objects. Graphs are widely used to model several types of information, social, biological and physical processes and relations. A graph at the very least must contain one single node. A graph with a single node is called Singleton graph. In the context, a graph is made up of several nodes or vertices. These are connected together by use of edges, lines or

arcs. The graph, therefore is made up of ordered pairs of vertices and edges. The graphs could take different forms, for instance, walks, trails, cuts trees, forests and directed graphs. Basically, a road network will take the form of a graph, with the origin and the destination connected by a number of roads and paths. It is in such a graph that the algorithm to find the optimal route is applied (Agafonov & Myasnikov, 2018).

## **2. Methodology, design and simulation**

The design of our research requires the conversion of a map of a road network to a suitable graph with nodes and weighted edges. The nodes represent the road intersections, while the edges represent the roads and the weights are given depending on the length of the road and its traffic density. From the flow chart shown in Figure 1, the program starts by taking the nodes, edges and edge weights of a graph to generate the graph. Once the graph is ready, the program queries the user to enter the origin and the destination in which the shortest is to be determined. The algorithm then checks whether the origin and the destination entered are valid nodes on the graph. If they are valid, the algorithm then proceeds to determine the shortest path. If they are not valid, the algorithm loops back to allow the user to input the origin and the destination again.

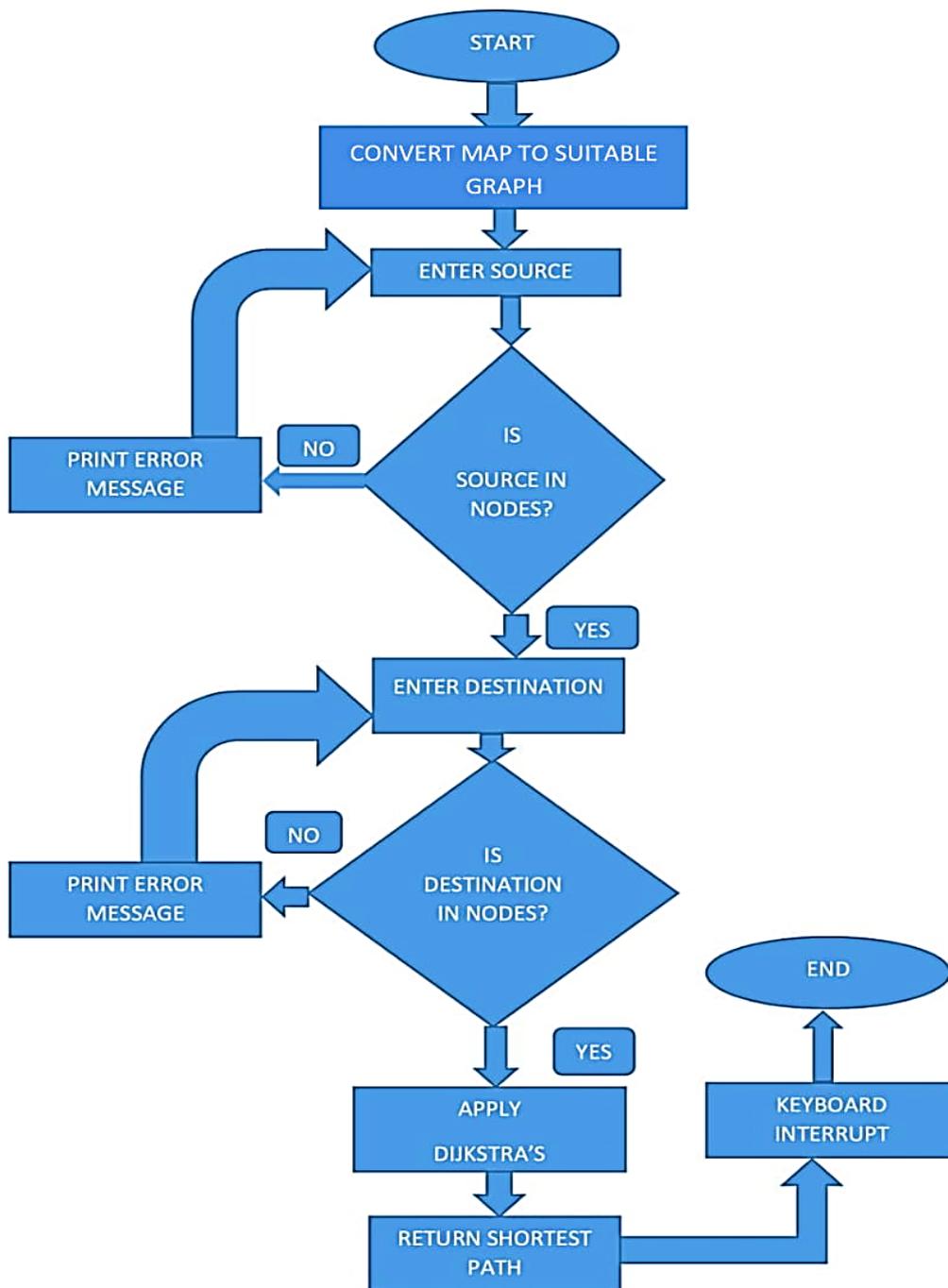


Figure no. 1: Flowchart of the traffic routing algorithm implementation

### 2.1. Design

A section of the map shown in Figure no. 2 is converted to a graph. In the graph implemented, the traffic flows in both directions. This section of roads is chosen for the purposes of demonstrating the

working of the algorithm. The traffic density on the roads is susceptible to change with peak and off peak hours. This would translate to variations in the weights assigned to the roads at different times.

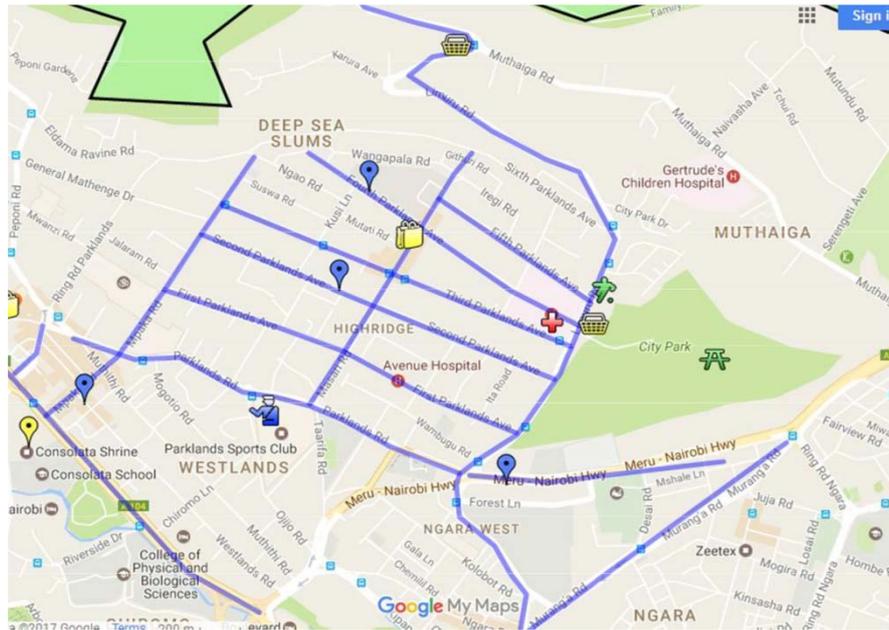


Figure no. 2: A map section of the road network in Parklands  
(Source: Google maps)

Table no. 1 below shows the nodes used on the graph to denote the various road junctions.

**Table no. 1**  
*Road junctions represented by nodes*

<b>NODE</b>	<b>JUNCTION</b>
A	Parklands Road/Limuru Road Junction
B	Parklands Road/Masari Road Junction
C	Parklands Road/Mpaka Road Junction
D	First Parklands Avenue/Limuru Road Junction
E	First Parklands Avenue/Masari Road Junction
F	First Parklands Avenue/Mpaka Road
G	Second Parklands Avenue/Limuru Road Junction
H	Second Parklands Avenue/Masari Road Junction
I	Second Parklands Avenue/Mpaka Road Junction
J	Third Parklands Avenue/Limuru Road Junction
K	Third Parklands Avenue/Masari Road Junction
L	Third Parklands Avenue/Mpaka Road Junction
M	Fifth Parklands Avenue/Limuru Road Junction
N	Fifth Parklands Avenue/Masari Road Junction
O	Mpaka Road/Ngao Road Junction
P	Sixth Parklands Avenue/Limuru Road Junction
Q	Sixth Parklands Avenue/Ngao Road Junction

(Source: Google maps)

## 2.2. Tools

This paper primarily uses the Dijkstra's algorithm to determine the shortest path. The algorithm is implemented using python. Therefore, all the program writing is done using python. Using the knowledge of graph theory, the road network considered in the study is converted to a graph with V represents number of vertices and E represents number of edges.

## 2.3. Methodology

The data used is a section of the Google map shown in Figure 2 above. Since it is not possible to indicate the nodes on the map, Table 1 above summarizes the nodes and the junctions they denote. These can be traced back to the map at the various road intersections. The network of the major roads is converted to a graph where the various roads sections are represented by the edges while the various road intersections / junctions are represented by the vertices. The weights on the edges of the graph, i.e., the road sections, are obtained from the real time values of the approximate time taken to drive from one road junction to another. This is done by the

program querying the values of the weights from Google maps using a public Application Programming Interface (API). The data is returned as a Json object which is unpacked to obtain the value of time that a motorist would take, hence giving the weight of the roads. The weights of the roads are updated every time the program runs.

## 2.4. Simulation

To simulate the working of the algorithm, a road network map of major roads is converted into a graph. The time taken to move from one location to another is obtained from Google maps by use of a public API. Dijkstra's algorithm is applied on the graph where one inputs the origin and the destination and the algorithm returns the fastest route and the shortest time a car is likely to take. Several tests are conducted by running the program at different times.

## 3. Results and discussion

### 3.1. Results

Table no. 2 shows the roads and their weights at different time for three tests, test 1, tests 2, and tests 3.

**Table no. 2**  
*Roads and their weights at different times*

Edge	Roads represented by edge	Weight Test 1 Carried at 1:06 PM on 23/5/2017	Weight Test 2 Carried at 11:06 AM on 30/5/2017	Weight Test 3 Carried at 5:45 PM on 2/6/2017
A-B	Parklands Road/Limuru Road Junction-Parklands Road/Masari Road Junction	87	222	222
B-C	Parklands Road/Masari Road Junction-Parklands Road/Mpaka Road Junction	181	181	184
D-E	Parklands Road/Mpaka Road Junction-First Parklands Avenue/Masari Road Junction	196	196	201
E-F	First Parklands Avenue/Masari Road Junction- First Parklands Avenue/Mpaka Road	141	141	143

<b>Edge</b>	<b>Roads represented by edge</b>	<b>Weight Test 1 Carried at 1:06 PM on 23/5/2017</b>	<b>Weight Test 2 Carried at 11:06 AM on 30/5/2017</b>	<b>Weight Test 3 Carried at 5:45 PM on 2/6/2017</b>
G-H	Second Parklands Avenue/Limuru Road Junction - Second Parklands Avenue/Masari Road Junction	146	146	152
H-I	Second Parklands Avenue/Masari Road Junction - Second Parklands Avenue/Mpaka Road Junction	166	166	170
J-K	Third Parklands Avenue/Limuru Road Junction - Third Parklands Avenue/Masari Road Junction	184	184	184
K-L	Third Parklands Avenue/Masari Road Junction - Third Parklands Avenue/Mpaka Road Junction	143	143	142
M-N	Fifth Parklands Avenue/Limuru Road Junction - Fifth Parklands Avenue/Masari Road Junction	128	128	127
P-Q	Sixth Parklands Avenue/Limuru Road Junction-Sixth Parklands Avenue/Masari Road Junction	410	410	400
A-D	Parklands Road/Limuru Road Junction-First Parklands Avenue/Limuru Road Junction	86	86	90
D-G	First Parklands Avenue/Limuru Road Junction-Second Parklands Avenue/Limuru Road Junction	59	59	66
G-J	Second Parklands Avenue/Limuru Road Junction-Third Parklands Avenue/Limuru Road Junction	36	36	39
J-M	Third Parklands Avenue/Limuru Road Junction-Fifth Parklands Avenue/Limuru Road Junction	45	45	45
M-P	Fifth Parklands Avenue/Limuru Road Junction-Sixth Parklands Avenue/Limuru Road Junction	55	65	66
B-E	Parklands Road/Masari Road Junction-First Parklands Avenue/Masari Road Junction	46	46	47
E-H	First Parklands Avenue/Masari Road Junction-Second Parklands Avenue/Masari Road Junction	50	50	52
H-K	Second Parklands Avenue/Masari Road Junction-Third Parklands Avenue/Masari Road Junction	49	49	53
K-N	Third Parklands Avenue/Masari Road Junction-Fifth Parklands Avenue/Masari Road Junction	85	85	85

Edge	Roads represented by edge	Weight Test 1 Carried at 1:06 PM on 23/5/2017	Weight Test 2 Carried at 11:06 AM on 30/5/2017	Weight Test 3 Carried at 5:45 PM on 2/6/2017
Q-O	Sixth Parklands Avenue/Masari Road Junction-Fifth Parklands Avenue/Mpaka Road Junction	15	15	15
C-F	Parklands Road/Mpaka Road Junction-First Parklands Avenue/Mpaka Road	102	102	102
F-I	First Parklands Avenue/Mpaka Road-Second Parklands Avenue/Mpaka Road Junction	70	70	70
I-L	Second Parklands Avenue/Mpaka Road Junction-Third Parklands Avenue/Mpaka Road Junction	28	28	28
L-O	Third Parklands Avenue/Mpaka Road Junction-Fifth Parklands Avenue/Mpaka Road Junction	80	80	80

(Source: Google maps)

Table no. 3 shows fastest routes between various points using test 2 weights.

**Table no. 3**  
*Fastest routes between various points using test 2 weights*

ORIGIN	DESTINATION	FASTEST ROUTE	APPROXIMATE TIME (Using Test 2 Weights)
A	G	A-D-G	2 min
C	J	C-B-E-H-G-J	7 min
F	P	F-E-G-H-J-M-P	6 min
O	A	O-L-K-H-G-D-A	9 min
M	C	M-J-G-H-E-B-C	8 min
D	Q	D-G-H-K-L-O-Q	8 min
E	P	E-H-G-J-M-P	4 min
B	N	B-E-H-K-N	3 min
G	I	G-H-I	5 min
O	E	O-L-I-F-E	5 min
J	F	J-G-H-E-F	6 min
L	D	L-K-H-G-D	6 min
Q	D	Q-O-L-K-H-G-D	8 min
N	G	N-M-J-G	3 min
K	C	K-H-E-B-C	5 min
E	Q	E-F-I-L-O-Q	5 min
A	H	A-D-G-H	4 min
J	F	J-G-H-E-F	6 min
A	L	A-D-G-H-K-L	8 min
B	P	B-E-H-G-J-M-P	5 min

(Source: Google maps)

### 3.2. Discussion

The system develops a section of road network. The user inputs the origin and the destination and the system applies Dijkstra's algorithm and outputs the fastest route and the approximate time a motorist would take to move between the points. Each road is represented by a particular edge which is assigned a weight that corresponds to the approximate time the motorist would take to move between the two points, depending on the traffic condition. This data is obtained from a public Google maps API. Each time the program runs, a query is sent to Google through the public API. Google sends back a JSON object containing the distance between the two points as well as the time a motorist would take to move between the two points. This JSON object is decoded and the value of the time taken in seconds is used to give the weight of the roads. When all the roads are weighted, the system proceeds to apply Dijkstra's algorithm on the weighted graph of roads. Therefore, the algorithm considers the route with the minimum time and returns it together with the total approximate time in minutes.

As seen in Table 2 above, the three sets of weights clearly show that the traffic is time dependent, since the traffic density will depend on the time of the day the query is made. Peak and off peak hours determine the weight placed on each road for instance, it is clear that for test 3, some roads have more weights compared to test 1. Applied on a wide scale network of roads, the system can be helpful to route motorists to the fastest routes. This would help alleviate the problem of traffic snarl-ups in the major cities.

### 4. Conclusions

In conclusion, it is evident that in the current world, efficiency and reliability are critical when developing systems. There is a high need to save time especially on the roads and also obtain information on the traffic situation that can be relied upon. The system developed shows that the fastest route from one point to another can be determined with a suitable algorithm. This meets the main objective of the paper. Additionally, reliable data was obtained from Google maps which are used to give weights to the various routes.

### REFERENCES

- Agafonov, A., & Myasnikov, V. (2017). Efficiency comparison of the routing algorithms used in centralized traffic management systems. *Procedia Engineering, Vol. 201*, 265-270.
- Agafonov, A., & Myasnikov, V. (2018). Vehicle routing algorithms based on a route reservation approach. *Journal of Physics: Conference Series, Vol. 1069*, 1-9.
- Mu, H., Yu, J., & Liu, L. (2009). Shortest path algorithm for road network with traffic restriction. *IEEE 2nd International Conference on Power Electronics and Intelligent Transportation System*, China.
- Nha, V., Djahel, S., & Murphy, J. (2012). A comparative study of vehicles' routing algorithms for route planning in smart cities. *IEEE First International Workshop on Vehicular Traffic Management for Smart Cities*, Ireland.