The International Journal of Engineering and Science (IJES)
|| Volume || 8 || Issue || 10 Series I || Pages || PP 40-47 || 2019 ||
ISSN (e): 2319 – 1813 ISSN (p): 23-19 – 1805

Graph Theory and Dijkstra's Algorithm:A solution for Mumbai's BEST buses

*Arjun Kejriwal[a], Prof. Aqsa Temrikar[b]
(a: student, Dhirubhai Ambani International School, b: teacher, Vibgyor High)
*Corresponding Author: *Arjun Kejriwal[a]*

--ABSTRACT--
The paper addresses the lack of a systematic map of the BEST bus routes in Mumbai and people's, especially newcomers' or tourists', limited knowledge of how to travel from one station to another in the shortest amount of time. By using Graph theory and Dijkstra's Algorithm (shortest-path algorithm), the paper not only provides a map for the BEST bus routes in Mumbai's Western suburbs to start with but also provides pseudocode that finds the shortest path between two areas (nodes).

Date of Submission: 06-10-2019 Date of acceptance: 23-10-2019

I. Introduction

Everyday, approximately 2.9 million people travel in Mumbai's BEST buses, out of which 40% are either unfamiliar with the route of the bus or are unaware of how to travel to a particular destination in the least amount of time [1]. From a recently conducted survey and published statistics, it was concluded that a person wastes 15 minutes, on average, due to a lack of knowledge of how to get to a destination in the fastest possible way. So if about 1.16 million commuters each waste an average of 15 minutes, Mumbai's economy is losing upon about two hundren thousand hours of productive work (assuming each person's productivity to be two-thirds on average). This huge loss, coupled with the fact that some commuters may also end up spending more money by taking a longer route, if improved, can give a major boost to the Indian economy.

Fig1. Mumbai BEST buses

The solution can be found in Graph Theory. Dijkstra's algorithm (also called the shortest-path algorithm), a major part of Graph Thoery, can be used to find the shortest path between two areas/stations of Mumbai based on actual routes of the BEST buses. This report not only discusses the full application of this algorithm in this problem but also provides the specialised pseudocode of this algorithm, which can act as a starting point for the formation of several applications or websites pertaining to this dire issue. It can also provide the groundwork and mechanism for the upcoming 1000 air-conditioned BEST buses.

II. DESCRIPTION

2.1 Overview of Graph Theory:

Graph theory is a fundamental part of Discrete Mathematics that deals with graphs, which are data structures defined by nodes (vertices) and edges (lines joining thre nodes) [2]. Graphs are usually denoted by $G(V, E)$, where V is the number of nodes (vertices) and E is the number of edges. Graphs mainly show the interconnectedness of several nodes, which can represent anything ranging from cities to devices in a LAN. Graph theory has extensive applications in social networks, LAN and WAN systems, chemical compound structures and transportation routes. However, its most common and useful application is in scenarios where one uses algorithmsto identify a problem's solution that minimises a certain parameter, such as the length of a path, the time taken to travel the path, connecedness of the nodes, etc. Thus, Dijkstra's algorithm is the most

appropriate and effective tool for minimising the distance covered by bus commuters to get from one station to another.
Graph theory is one of the most widely used mathematical topics across the globe given its extensive applications. However, it is also a very sohpisticated, nuanced and complex topic. Therefore, there are certain terms that areexclusive to Graph theory. They are defined below, for clarification purposes and the removal of ambiguity.

Table1. Graph Theory termologies

Term	**Definition**
Node (Vertex)	A circular point on a graph that represents an object or a location
Edge	A line connecting two nodes, thus showing the relation between them
Degree of a Node	The number of vertices a node is connected to/ The number of edges incident on a node
Path	A journey along the graph from one node to another without any repetition of nodes and edges.
Undirected Graph	A graph where all edges are bidirectional and a path can be constructed either way

2.2 Local Survey:

In order to understand the primary problems faced by BEST bus commuters, a Google Forms Survey was created and sent to around 50 employees in the company 'Seals Retail World Private Limited'.Figure 2 below shows the results of one of the most important questions on the form.

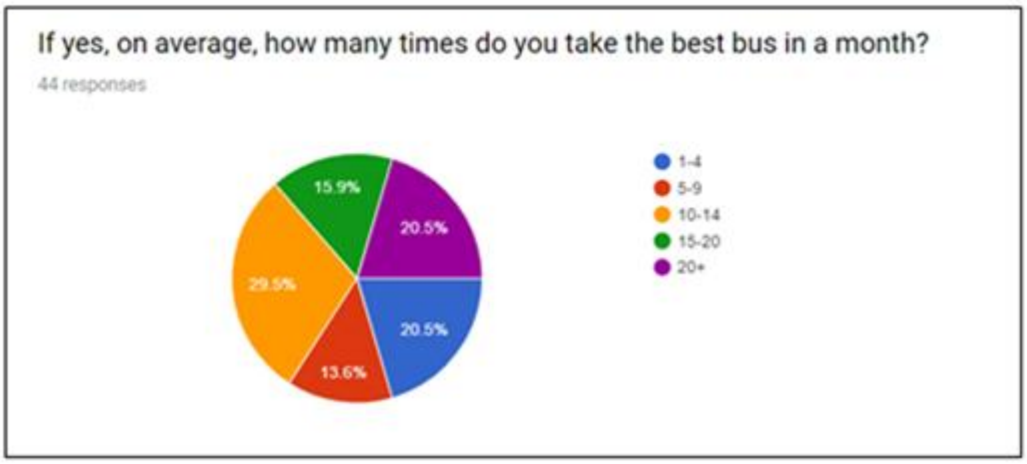
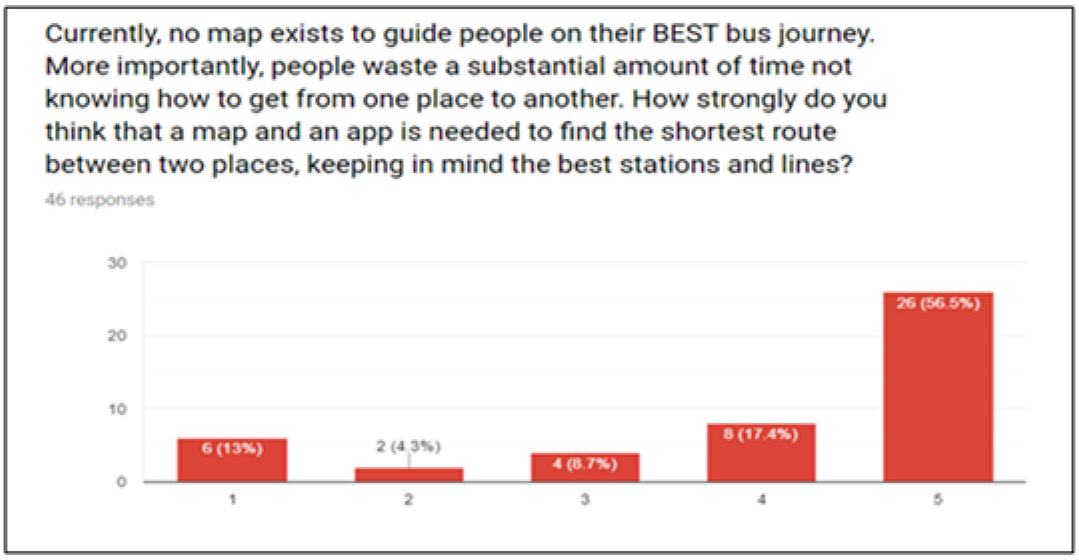


Fig2. Number of times the surveyors took the bus in a month

The maximum number of people answered that they take the BEST buses 10-14 times in a month, while around 40% mentioned that they take the bus more than 15 times in a month. Having been accustomed to travelling often in the BEST buses, the employees were then asked which of the problems they face while travelling in BEST buses is the most challenging. At least 70% of the surveyers mentioned that the lack of a systematic map and the unawareness of how to go from one station to another in the shortest amount of time were their primary concerns while travelling in BEST buses. Thus, this report addresses this very problem. Moreover, the surveyers were asked whether the solution being proposed in this report would benefit them. As shown by the results in Figure 3 on the next page, the majority felt that the solution will benefit them tremendously (5 being strongly agree and 1 being strongly disagree).


Currently, no map exists to guide people on their BEST bus journey. More importantly, people waste a substantial amount of time not knowing how to get from one place to another. How strongly do you think that a map and an app is needed to find the shortest route between two places, keeping in mind the best stations and lines?

46 responses

Fig3. Mumbai BEST buses

2.3 Regions covered and Bus Route details:

For the purpose of simplicity, the report only considers the Western part of Mumbai, which contains the highest number of BEST buses and will contain a large amount of air-conditioned buses too. Specifically, the report focuses on the 14 most popular and busy regions in Western Mumbai, each of which have about three stations within them. Again, for simplicity, the report combines the stations into one major station for each region. This will help in finding the connectedness (the existence of an edge between two nodes) between different regions across Western Mumbai, thus allowing commuters to minimise travel time while travelling long-distance.
The different regions are each asigned one node. They are as follows [3]:

Table2. Nodes representing popular regions in Mumbai Western

Node	Area with BEST station	Node	Area with BEST station
JO	Jogeshwari	SW	Santacruz West
LO	Lokhandwala	KW	Khar West
VE	Versova	PH	Pali Hill
AM	Amboli	BW	Bandra West
AE	Andheri East	TLE	Taj Lands End
JU	Juhu	KH	Kherwadi
VP	Vile Parle	BKC	BandraKurla Complex

Whether a node is connected to other nodes depends on the BEST routes considered. For simplicty, the report focuses on 13 different and popular BEST routes in Mumbai Western.

Table3.The BEST routes

ROUTE NO.	ITENERY
Route 317	Bandra Kurla Complex ➔ Kherwadi
Route 215	Bandra Kurla Complex ➔ Taj Lands End
Route 211	Kherwadi ➔ Pali Hill
Route 187	Kherwadi ➔ Bandra Kurla Complex
Route 220	Bandra West ➔ Khar West
Route 630	Santacruz West ➔ Khar West
Route 231	Santacruz West ➔ Juhu
Route 627	Andheri East ➔ Juhu
Route 254	Jogeshwari ➔ Andheri East
Route 257	Amboli ➔ Juhu
Route 268	Lokhandwala ➔ Andheri East
Route 222	Versova ➔ Bandra West
Route 221	Versova ➔ Andheri East

The above routes combined will determine if one node is connected to another node (for example, if Andheri East is connected to Bandra West). With the nodes and edges established, the graph of Mumbai Western's BEST bus system can be made.

III. METHODOLOGY

3.1 Creation of Graph:

Using the location nodes and the BEST routesmentioned above, a graph matrix can be formed showing how each node is connected to other nodes.

Table4. Graph matrix showing node connectedness

Location (Node)	**JO**	**LO**	**VE**	**AM**	**AE**	**JU**	**VP**	**SW**	**KW**	**PH**	**BW**	**TLE**	**KH**	**BKC**
JO	--	0	0	1	1	0	0	0	0	0	0	0	0	0
LO	0	--	0	1	0	0	0	0	0	0	0	0	0	0
VE	0	0	--	1	0	1	1	0	0	0	1	0	0	0
AM	1	1	1	--	0	0	1	1	0	0	0	0	0	0
AE	1	0	0	0	--	1	0	0	0	0	1	0	0	0
JU	0	0	1	0	1	--	0	1	0	0	0	0	0	0
VP	0	0	1	1	0	0	--	1	0	0	0	0	0	0
SW	0	0	0	1	0	1	1	--	1	0	0	0	0	0
KW	0	0	0	0	0	0	0	1	--	1	1	0	0	0
PH	0	0	0	0	0	0	0	0	1	--	1	0	0	0
BW	0	0	1	0	1	0	0	0	1	1	--	1	1	0
TLE	0	0	0	0	0	0	0	0	0	0	1	--	1	0
KH	0	0	0	0	0	0	0	0	0	0	1	1	--	1
BKC	0	0	0	0	0	0	0	0	0	0	0	0	1	--

The low connectedness depicted in the matrix above may be because of two reasons:
a) it is challenging to design BEST routes such that all nodes are connected to several other nodes
b) the report has considered low connectedness for simplicity. Using the above graph matrix, the graph incorporating the location nodes and the connectedness in Mumbai Western can be constructed.

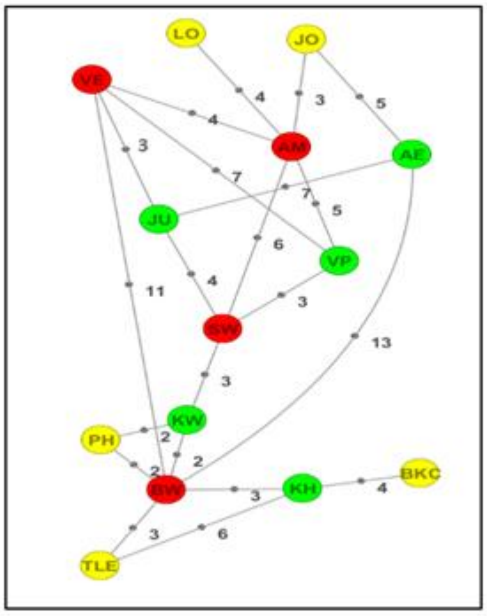


Fig5. Digital graph depicting location node connectedness through BEST buses in Mumbai Western

Graph colouring has been used in the graph above, with yellow denoting less famous areas, red denoting highly popular areas, and green denoting types of areas somewhere in between the two extremes.

Dijkstra's algorithm, as will be discussed below, needs an undirected graph with weighted edges. That is, the edges of the graph should be bidirectional and the edges should have weights. In this case, distance between nodes acts as the best weight for the edges, as it is mostly distance that determines how much time it takes for a bus to travel from one node to another. However, several assumptions are being made here. Firstly, the report is assuming that the bus will travel at the same speed while travelling along different edges. Secondly, it is assuming that the level of obstacles in each edge journey is the same. Thirdy, it is also assuming that the bus follows a straight or smooth path with no turns, where, in reality, this might not be the case. Nevertheless, for simplicitly, the report will assume that distance will be the main determinant for travel time, which it is in many cases.

Table5. Edge distances as edge weights

Edge	**Distance in km (Weight)**	**Edge**	**Distance in km (Weight)**	**Edge**	**Distance in km(Weight)**
KW-BW	1.8≈2	**BW-KH**	2.5 ≈3	**AM-VP**	4.6 ≈5
PH-BW	1.6 ≈2	**VE-JU**	3.1≈ 3	**AM-SW**	6.2 ≈6
KW-PH	2	**LO-AM**	3.8 ≈4	**KH-TLE**	6.1 ≈6
JO-AM	2.5 ≈3	**VE-AM**	3.8 ≈4	**VE-VP**	7.1 ≈7
VP-SW	2.6 ≈3	**JU-SW**	3.7 ≈4	**JU-AE**	7.2 ≈7
SW-KW	3.0 ≈3	**KH-BKC**	4.1 ≈4	**VE-BW**	10.6 ≈11
BW-TLE	2.6 ≈3	**JO-AE**	5.3 ≈5	**AE-BW**	12.7 ≈13

The above values for distance (edge weights) can effectively be used in Dijkstra's shortest-path algorithm as will be seen in the next section.

3.2 Dijkstra's Algorithm:

Dijkstra's algorithm, also called Dijkstra's Shortest Path First algorithm, is an algorithm part of Graph theory that finds the shortest path between any two nodes in a graph. This algorithm was proposed by computer scientist Edsger W. Dijkstra in 1956 and it was published in a report by him three years later. This algorithm has extensive uses in fields like computer science. However, this report focuses on the algorithm's applications in bus systems.

In the specific case of the BEST buses, if a hypothetical commuter whishes to travel from one node to another, he/she will want to do so in the least possible time. This can be made possible using Dijkstra's algorithm. But before an example is considered, the report will first briefly describe the general workings of the algorithm. The steps of the algorithm are as follows [4](current distance is defined as the distance between a node and the starting node along a defined path):

1) Pick a starting and ending node. Mark the starting node with a current distance of 0; all the other nodes should be marked with an infinity symbol as their distances from the starting node is currently unknown.
2) Identify all the nodes that are connected to the current node. Mark the non-visitied node with the smallest current distance as the new current node.
3) Identify all the neighbouring nodes to this current node. For each neighbour, add the current distance of the current node with the weight of the edge connecting the current node with the neighbour. For whichever neghbouring nodes this sum is smaller than the current distance of that neighbouring node, set this sum as the new current distance of the neighbouring node.
4) Mark the current node (not the neighbours) as visited
5) If there are still non-visited nodes left as neighbours to the current node, repeat steps 2-5.
6) Once the ending node has been reached, the current distance of the ending node will be the shortest path between the starting and ending nodes.

The paper will now apply this algorithm to a specific journey from a starting node to an ending node with a large distance in between them on the graph.

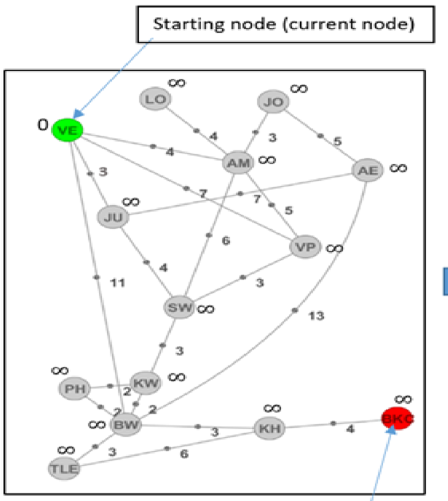


Fig6. *Stage 1 (starting green node and ending red node are defined)*

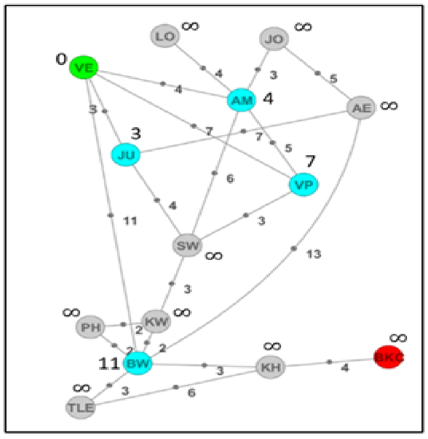


Fig7. *Stage 2 (VE is the current node, and other blue nodes show possible paths)*

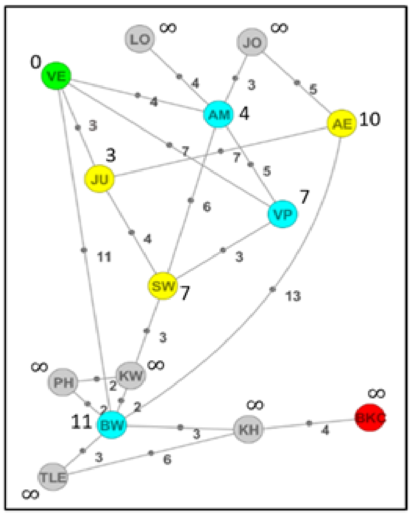


Fig8. *Stage 3 (JU is the current node, and other yellow nodes show possible paths)*

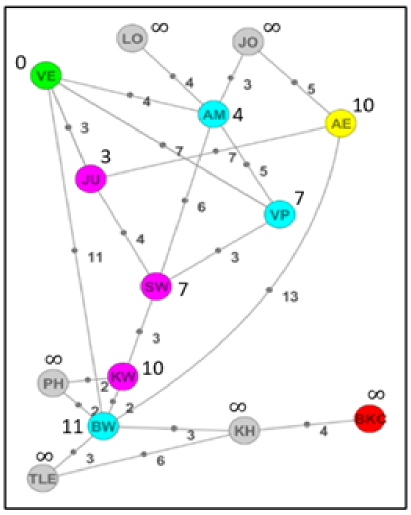


Fig9. *Stage 4 (SW is the current node, and node KW is the only next option available)*

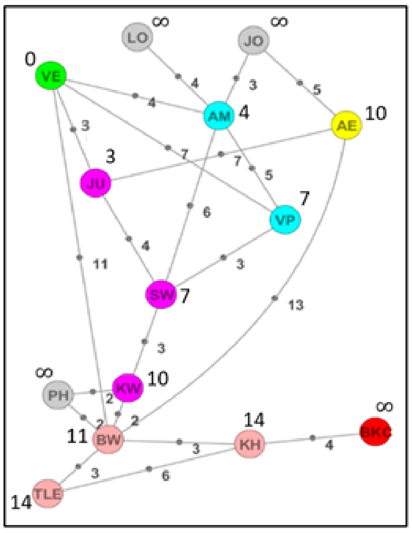


Fig10. *Stage 5 (BW is the current node, and other peach nodes show possible paths)*

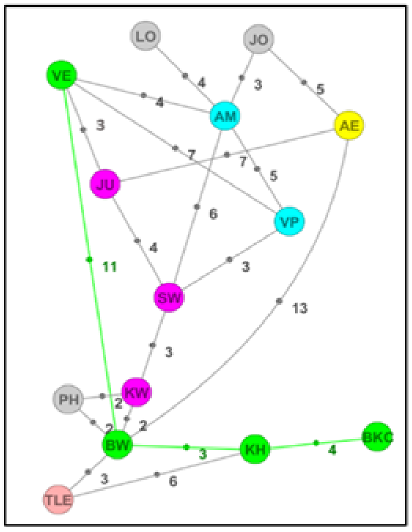


Fig11. *Stage 6 (The green coloured path with the green nodes is the shortest path from VE to BKC)*

As seen from the series of six graphs above, Dijkstra's algorithm determined that the fastest way of travelling from Versova (VE) to Bandra Kurla Complex (BKC) is to take the following route: (VE➔BW➔KH➔BKC). This shortest path is shown in green in Stage 6 above and covers about 18 kilometres. This really shows the usefullness of Dijkstra's algorithm, since people usually travel to Juhu (JU) and not Bandra West (BW) first from Versove (VE), wasting a considerable amount of time as a result of travelling 1 to 2 kilometres more.

According to the local survey conducted, Versova was rated the second most popular area for BEST buses. Since Versova has several BEST buses starting their trips from there, the paper will now look at the shortest BEST routes from the Versova (VE) node to all the other nodes.

Table6. Shortest distance from Versova (VE) node to all the other 13 nodes as determined by Dijkstra's algorithm

Location (Node)	**Approximate shortest distance from Versova (VE) node (in km)**
JO	7
LO	8
AM	4
AE	10
JU	3
VP	7
SW	7
KW	10
PH	12
BW	11
TLE	14
KH	14
BKC	18

3.3 Algorithm code [5]:

```
function dijkstra(G, C), where G denotes Graph and C denotes current distance
for each node Nin G
distance[N] ← infinite
      previous[N] ← NULL
 distance[C] ←0
If N != C, add N to PriorityQueue Q

   while Q IS NOT EMPTY
      U ←Extract MIN from Q
      for each unvisited neighbour N of U
         currDistance ← distance[N] + edge_weight(U, N)
         if currDistance < distance[N]
            distance[N] ← currDistance
            previous[N] ← U
return distance[N], previous[N]
```

The above pseudocode is for the Dijkstra's algorithm and can act as a starting point for preparing source code in Python or C++ for Dijkstra's algorithm.

IV. RESULTS AND CONCLUSION

This paper started with introducing the problem faced by several commuters in Mumbai of spending exessive time commuting in BEST buses and not being well-versed with travelling efficiently across Mumbai. After giving a brief description of the problem at hand, the paper discussed the results of a local survey conducted to consider the opinions of the residents of Mumbai on this issue. The paper then went on to create a graph representing western Mumbai and used dijkstra's algorithm to address this issue.

The paper used an example of a popular BEST route starting from the Versova area. This example highlighted the usefulness of Dijkstra's algorithm as it found that the fastest route from Versova to Bandra Kurla Complex was one of the least popular ones. People are unaware of several fast routes such as these, and informing them about this is vital for reducing people's commute time. The implementation of Dijkstra's algorithm in applications, using the pseudocode provided, can be instrumental in abating this problem and making BEST bus travel in Mumbai much more efficient for passengers.

V. EVALUATION AND FUTURE SCOPE

5.1 Evaluation:

The paper provides the BEST bus system of Mumbai with a solution that can potentially make the system much more efficient, saving substantial amounts of time for commuters and providing a wholesome map for reference. However, as mentioned before, the paper is making various assumptions while providing the solution. For example, the traffic level or the smoothness of the road can be better indicators than distance of how fast a bus can move from one node to the other. However, this can vary for different roads and areas. Distance was used as the weight for unifying the solution-forming process and because it is porportional to time taken for most areas and roads in Mumbai. It must be kept in mind that this paper acts like a starting point for further research in this field and is not putting an end or giving a final conclusion to this matter.

5.2 Future scope

The paper can be extended in several ways. Firstly, Dijkstra’s algorithm can be applied to all the other 13 nodes considered above to create a wholesome guide for efficient travel in Western Mumbai. However, this paper need not be limited to Western Mumbai only. It can address the problem of inefficiency in the BEST system in all parts of Mumbai city. Moreover, the paper can also prepare actual source code for Dijkstra’s algorithm that can directly be used in the development of applications specifically for the BEST bus system in Mumbai. The paper can also use other shortest path algorithms in Graph Theory such as Bellman-Ford algorithm and Floyd-Warshall algorithm in order to address this particular problem.

More importantly, the paper can use other types of edge weights as well, such as the level of traffic flow or the smoothness of the road. By considering each of these along with distance, the paper can provide a much more relevant, accurate and wholesome solution for the BEST bus system of Mumbai.

REFERENCES

[1]. https://www.hindustantimes.com/mumbai-news/35-000-workers-of-mumbai-s-best-buses-get-their-salaries worth-rs-1-5crore-in-coins/story-aDxnCHfRwlthhnESeOIML.html
[2]. https://www.geeksforgeeks.org/mathematics-graph-theory-basics-set-1/
[3]. https://moovitapp.com/index/en/public_transit-lines-Mumbai-3732-857915
[4]. https://www.codingame.com/playgrounds/1608/shortest-paths-with-dijkstras-algorithm/dijkstras-algorithm
[5]. https://www.programiz.com/dsa/dijkstra-algorithm
[6]. Sneyers J., Schrijvers, T. and Demoen B. Dijkstra's Algorithm with Fibonacci Heaps: An Executable
[7]. Description in CHR. 20th Workshop on Logic Programming (WLP'06), Vienna, Austria, February 2006.
[8]. Ravi, N., Sireesha, V. Using Modified Dijkstra’s Algorithm for Critical Path Method in a Project Network.
[9]. International Journal of Computational and Applied Mathematics. Volume 5 number 2 pp 217-225. 2010.

Arjun Kejriwal" Graph Theory and Dijkstra’s Algorithm:A solution for Mumbai’s BEST buses"The International Journal of Engineering and Science (IJES), 8.10 (2019): 40-47