Computers and Geosciences

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

--Manuscript Draft--

Manuscript Number:	CAGEO-D-20-00048R1
Article Type:	Research Paper
Section/Category:	Algorithms
Keywords:	GIS; Road Density Analysis; Shortest Path; A* algorithm; Uniform Cost Search; Fare Estimation
Corresponding Author:	Md. Rahat- uz- Zaman, B.Sc. Khulna University of Engineering & Technology Khulna, Khulna BANGLADESH
First Author:	Md. Rahat- uz- Zaman, B.Sc.
Order of Authors:	Md. Rahat- uz- Zaman, B.Sc.
	Shadmaan Hye, B.Sc. in Computer Science and Engineering
Abstract:	History clearly shows from the former times to recent times that the development of a region is heavily contingent on the transport network. Nowadays, however, there are a number of network transport problems, including increased operational costs for vehicles, delay, road congestion, pollution, and road accidents, which are becoming the major issues in a developing country like Bangladesh. Digitized analysis and strategy formulation on constructing, maintaining and upgrading are crucial to get the greatest available solution for these problems. This research contributes towards this analysis by creation, evaluation and compilation maps from obtained information. The present study made an effort to prepare road networks and application of several Geographic Information System (GIS) techniques on the pedestrian, rickshaw, metro and public transit transports systems and receive the optimum transport solution on a variety of cost metrics like traversing distance, velocity and time. Another important discovery made in this research is the relationship between regional road density to cost and travel time. This article demonstrated that in Dhaka city area, based on efficient shortest path algorithm design and a graph generated with 49869 nodes and 53878 edges, the shortest distance between two points may not always be cost-effective and conversely.
Suggested Reviewers:	
Response to Reviewers:	Dear sir, Thank you for your helpful advice on our work. We have updated the highlights according to the requirements. Thank you again. Sincerely, Md. Rahat-uz-Zaman, Shadmaan Hye Department of Computer Science and Engineering Khulna University of Engineering & Technology

20th June, 2020

Pauline Collon, Dario Grana, Derek Karssenberg
Editor-in-Chief
Journal of Computers & Geosciences

Dear sir,

I am pleased to submit an original research article entitled "**Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques**" by **Md. Rahat-uz-Zaman** and **Shadmaan Hye** for consideration for publication in Computers & Geosciences. This research work was started in November, 2019 with a hackathon named Code Samurai. The task of the hackathon was to design a backend map direction provider of Dhaka city according to the users cost and time demands. The authors of this research have developed an algorithm with Dijkstra's shortest path and were able to get two awards in the hackathon. Later on, we further researched about the topic with the data provided in the hackathon, and developed some very useful algorithms similar to Uniform Cost Search and A* path finding algorithm.

In this manuscript, we show a local transports fare estimation algorithm and its application in Dhaka city. Furthermore, with the help of this, we show the preprocessing of street map data, road network density analysis and best path finding algorithm for time and expense.

We believe that this manuscript is appropriate for publication by Computers & Geosciences because it has the relation between Computer Science algorithms and Geosciences. The proposed methodology has algorithms, visualization and street map analysis which are strongly inside the scope of the journal. Our manuscript creates a paradigm for future studies of the evolution of street map analysis and graph algorithm design.

This manuscript has not been published and is not under consideration for publication elsewhere. We have no conflicts of interest to disclose. If you feel that the manuscript is appropriate for your journal, we will be glad to see our hard work on your journal.

Thank you for your consideration!

Sincerely,

Md. Rahat-uz-Zaman
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna-9203
Mobile: +8801521331082
Email: rahatzamancse@gmail.com

Response in plain text

Dear sir,

Thank you for your helpful advice on our work.

We have updated the highlights according to the requirements. Thank you again.

Sincerely,

Md. Rahat-uz-Zaman, Shadmaan Hye
Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Highlights

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

Md. Rahat-uz-Zaman, Shadmaan Hye

- Local transport costs in road network require algorithms to estimate accurate fare.
- Shortest path algorithms fail due to inconsistent fare of transportation rates.
- A* algorithm performs comparatively better for both cheap and shortest path analysis.
- Many transports can be merged with parameters to create finer direction support.

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

Md. Rahat-uz-Zaman[1,*], Shadmaan Hye[2]

Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Abstract

History clearly shows from the former times to recent times that the development of a region is heavily contingent on the transport network. Nowadays, however, there are a number of network transport problems, including increased operational costs for vehicles, delay, road congestion, pollution, and road accidents, which are becoming the major issues in a developing country like Bangladesh. Digitized analysis and strategy formulation on constructing, maintaining and upgrading are crucial to get the greatest available solution for these problems. This research contributes towards this analysis by creation, evaluation and compilation maps from obtained information. The present study made an effort to prepare road networks and application of several Geographic Information System (GIS) techniques on the pedestrian, rickshaw, metro and public transit transports systems and receive the opti-

*Corresponding Author
Email addresses: `rahatzamancse@gmail.com` (Md. Rahat-uz-Zaman), `praptishadmaan@gmail.com` (Shadmaan Hye)

[1]Conceptualization of this study, Writing - Original draft preparation, Methodology, Software, Algorithm Design, Implementation

[2]Data curation, Writing - Review and Revision, Computational resource collection, Implementation

mum transport solution on a variety of cost metrics like traversing distance, velocity and time. Another important discovery made in this research is the relationship between regional road density to cost and travel time. This article demonstrated that in Dhaka city area, based on efficient shortest path algorithm design and a graph generated with 49869 nodes and 53878 edges, the shortest distance between two points may not always be cost-effective and conversely.

Keywords: GIS, Road Density Analysis, Shortest Path, A* algorithm, Uniform Cost Search, Fare Estimation,

1. Introduction

Geographic Information System (GIS) technology is one of the fastest-growing road network technologies and the most widely utilized method for science in the world. GIS can capture all the necessary data, store it, manipulate, analyze, present and quickly visualize for the required effective solution of the user.

In a densely populated city center such as Dhaka (DHK) Bangladesh, several traffic queues alongside high transportation costs are traffic congestion, limited utility of broad roadways, a complex travel network, a comparatively higher number of road accidents Hoque [1], rapid urbanization triggering unevenly wide roads [2, 3], traffic concerned problem [4] and the problem of highway logistics [5]. For all emergency transport, it is proved that the main criteria is its readiness to reach any destination as fast as possible or at least within a fixed period [6]. The key aim of this groundwork is to provide enhanced network analysis and algorithms focused on certain cost parameters such as time and money.

In the following sections, firstly related works done by other researchers are addresses in 2. Section 3 specifies initial assumptions for the method. Then the methodology itself is discussed from section 4 to 6. Then it is extended to some real collected data, which is covered by subsection 7.1. Finally, the results of the implementation of this research on the datasets are defined in subsection 7.3.

2. Literature Review

A well amount of work on the transportation system is done recently in different countries in the world. The works include shortest path analysis, cost analysis, algorithm development and suggestions of increasing efficiency of traffic systems.

W.G. Rees [7] has used Dijkstra's algorithm to find the least-cost path in mountainous terrain of Wales. They have built the cost function from the topography alone and only considered footpaths. They were successful to build a foot-path map of Wales and determined the divergence of their path with the true straight line. In this paper, Uniform Cost Search algorithm, which is very similar to Dijkstra has been used to solve very similar objective. But a greater number of means of transportation and cost is considered.

S. Ahmed et al. [8] has worked on the Greater Cairo Region (GCR) to find the correct destination during an emergency. He has also proposed a method based on GIS to detect the shortest path from an emergency situation to the required destination. He also has applied closest facilities analysis on the region and achieved impressive results. The lack of various transportation methods such as bus, metro was a major limitation for their research.

L. Yang and D. Qian [9] presented a method vulnerability evaluation which is helpful to examine network vulnerability through final travel time cost and the structure of the network. They have conducted a simple case study on urban traffic and found a method to recover the traffic damaged due to roadblocks. But their most effort was only to detect the vulnerability and find alternate solutions if a road is blocked for any reason.

D. Das, A. Ojha, H. Kramsapi et al. [10] has prepared road transportation network for Guwahati city and did a similar work like S. Ahmed et al. With proper shortest path analysis and digitization, ArcGIS software [11] is used to determine the shortest path from a preselected source to required destination. Their result shows an impressive collection of data and visualization like road network map, ward map, junction and edges map, density map of both major and minor roads, lastly lane density. The analysis was very specific to Guwahati city and some of those cannot be applied to other regions like Bangladesh.

A. Quium and S. Hoque [12] has analyzed the vulnerability of road network in Bangladesh with the help of some indexing such as alpha (α) and gamma (γ). They have also measured the density of national and regional road networks and found that Dhaka city has more than average density of road in Bangladesh compared to other districts. But most of their works are based on inter-district national roads instead of regional roads inside a city. They have neither done any network analysis nor shortest path analysis which is done in this paper.

Qingquan Li [13] has proposed a technique to represent a road network on tiny mobile devices. The interesting part of the adaptive multi-scaled road network representation is that, the method can automatically set a feasible scale to geographic scope and other parameters. This can be used as the representation of this proposals application for different parameter e.g. travel by vehicle type.

In all the above-mentioned works, an application-focused proposal was not present or only representation of network is proposed in some works. In this paper, the authors try to use their proposal on applications so that the users may wish to analyze and detect the cheapest or fastest route in Dhaka city. Also unlike others, this methodology can be used for more and different vehicles which made the research work novel.

3. Transportation System of Bangladesh

This paper has proposed a suitable route analysis from a source location to a destination location based on the transportation system of Dhaka city. Because there are a lot of deficiencies in the existing transportations system in Bangladesh [14], usual road transport analysis developed for other countries will not work well in this country. In this country, there are 2 unique vehicles which are rickshaw and auto-rickshaw (CNG). As they are similar in size, shape and availability [15], in this research, they are considered of the same class vehicle. Though the fair of these two classes are very different, only cost of rickshaw is surveyed and taken into account for the research. Further mention of rickshaw in this paper will indicate both rickshaw and CNG. The common means of transportation are discussed in subsection 3.1.

3.1. Common Means of Transportation

There are various transportation systems in Bangladesh. Among those transportation systems, the most popular means of transportation in Dhaka city [16, 17] are discussed below.

1. **Metro route** [18, 19]: Highly cost and time-efficient. But only one route available in the whole city. Passengers need to wait at the station for the arrival time of the metro.
2. **Bus route** [20]: Cost-efficient and fast transportation. Passengers need to wait less time than the metro for the arrival of the bus at the bus stoppage.
3. **Rickshaw** [21, 22]: Costly and takes time to travel far distance. But always available on almost all roads except VIP roads. Most people choose rickshaw for short-distance transportation [23].
4. **Walking** [24]: Preferred for feasibly short distance. It takes the highest time but expense-free and can walk almost everywhere.

4. Building Graph from Geo-spatial Data

For the analysis of transportation, a graph from geo-spatial data is necessary. Now to build the graph, the costs of transportations are estimated and then with the help of the costs, several graphs are created. The final graph on which the analysis is applied is found after merging all the graphs.

4.1. Costs of Transportation in Graph

The main costs that are considered in this research are time and expense. While building the graph, time taken for a path is dynamically estimated while traversing from the length of the road, the speed of the transport vehicle considered and the traffic time parameter. The expense function is dependent on the transport vehicle, source and destination location of path in the network by interpolating from a database of collected fares. While building the graph, both of these attributes discussed above are included in each edge. The attributes of each node and edge are discussed in subsection 4.2.

As metro transportation system is not complete during this research in Bangladesh, the fares of metro routes are not available. As the fare should be lesser than bus fares in Dhaka city, the authors have used a reasonable fare of BDT 5 per kilometer which satisfies the above condition and used it where needed during the research. The fare between two stations is also rounded during usage.

Another function is required for the graph building which is waiting time at a bus or metro station. Based on some research [25], it is assumed that each bus will start at a specific stoppage in the morning and will be at a certain time at each station with an interval time.

Sample of dataset collected for rickshaw, bus and metro are presented correspondingly on table 1, 2 and 3. All possible costs of bus routes can be collected with ease. But for the reason that there are numerous roads for rickshaw, it is almost impossible to collect all possible rickshaw fare for all roads. For this reason, an interpolation method is used to estimate fare of rickshaw in an unknown road which is shown in equation 1.

The working of the interpolation method is visualized in figure 1. Each edge directly connected to the edge e will contribute a weighted fare (W_i) for

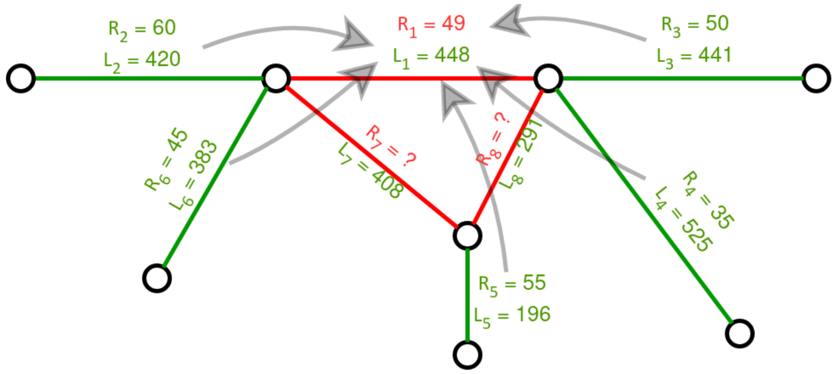

Figure 1: Demonstration of interpolation method to estimate rickshaw fare.

the estimation of fare of edge e. Here r_i is cost of rickshaw and R_i is cost of rickshaw per unit in road i with length L_i where i is a road directly connected via a node to edge e. This process is needed to apply several times as there might be some edges that are not directly connected to any edge with known fare. In these situations, the fare is estimated first for those adjacent edges of e and then are used for fare estimation of e.

$$\begin{aligned}
S_{ei} &= \{\text{set of length of edges of shortest path} \\
&\qquad\quad \text{from e to closest edge whose fare is known}\} \\
w_i &= \frac{1}{L_i + \sum_{l \in S_{ei}} l} \\
W_i &= \frac{w_i}{\sum_j w_j} \\
R_i &= \frac{r_i}{L_i} \\
R_e &= \sum_i W_i R_i
\end{aligned} \tag{1}$$

Table 1: Sample of dataset of rickshaw cost survey.

Source	**Destination**	**Cost (BDT)**	**Straight Line Distance (km)**	**Graph Traversal Distance (km)**	**Time of asking**
Bashundhara City	Banglamotor	50	0.50	0.71	10:00
Banglamotor	Mogbazar	45	1.59	1.88	13:30
BUET Quarter	Polashi	20	0.48	0.65	16:00
Azimpur	Nilkhet	25	0.42	0.44	14:30
Shimanto Square	Sonkor	60	1.70	1.70	21:00
Katabon	Motalab Plaza	30	0.76	1.00	20:00
...	...	...	...	...	...

Table 2: Bus fare survey for several source and destination locations.

Source	**Destination**	**Cost (BDT)**	**Straight Line Distance (km)**	**Graph Traversal Distance (km)**
Azimpur	Science Laboratory	10	0.90	0.95
Asad Gate	Kalyanpur	15	2.29	2.30
College gate	Gabtoli	20	3.11	3.40
Paltan	Shahbag	15	1.75	2.10
Banglamotor	Farmgate	10	1.22	1.30
Agargaon	Mirpur 10	30	3.48	3.49
...	...	...	...	...

Table 3: Metro fare survey for several source and destination locations.

Source	Destination	Cost (BDT)	Graph Traversal Distance (km)
Airport	Kamalapur	76	12.65km
Uttar Badda	Rampura	19	3.1km
Jhigatala	Golap Shah Mazar	26	4.4km
Gabtoli	Purba Dasherkandi	56	9.3km
Modhumoti	Mirpur 1	43	7.1km
Kochukhet	Gulshan 2	20	3.4km
...	...	...	...

4.2. Construction of Graph

The graph [26] for the transportation system in Bangladesh will be a directed graph with dynamic self-loops and parallel edges. A small preprocessed graph is displayed on figure 2. The graph must be directed because there are one-way roads in Dhaka city. Parallel edges are needed because two close locations may have more than one road. The self-loop edges will be used as waiting time in a station of bus or metro and will be dynamic depending on the requirement of waiting at the spot while traversing.

Primarily, an empty multi-edged directed graph is created. Then for all possible combinations of two paths in the input roads data, the intersection points are taken and added as nodes to the graph. A small sample of nodes data is shown on table 4 without the latitudes and longitudes. Each node must have the following attributes.

1. **Location name**: Location name in short form.
2. **Type**: Defined if the location is a bus stop, metro station or a junction of multiple roads.

3. **First arrival time**: This should be null if the type of the node is not a bus stand or a metro station. Else the first time when the bus or metro station will be in this attribute.
4. **Latitude and Longitude**: These are kept to calculate distance between two nodes.

While adding nodes to the graph, each node pair will also be joined together with an edge. A small sample of processed edges are represented in table 5. Each edge will have the following attributes.

1. **Type of route**: Defines if the route is available for rickshaw, bus or metro.
2. **Length of road in kilometers**: Length of the road calculated from the dataset.
3. **Cost of Rickshaw**: Usual cost of rickshaw to go from one node to another using this edge. The costs are calculated using interpolations from a dataset of a survey whose sample is in table 1. The interpolation is discussed in subsection 4.1.
4. **Traffic Time Parameter**: A parameter that will define how much traffic jams in the road represented by this edge. This is an important parameter for Dhaka city because almost all the roads in this city will have traffic jams [27]. This gives a value between 0 and 1 where 0 corresponds to very high traffic jam and 1 corresponds to no traffic jam. This can be fetched at times from online APIs like google maps.

After adding all the intersections of roads as nodes and the road itself as the edge linking between two intersections, the graph will be completely

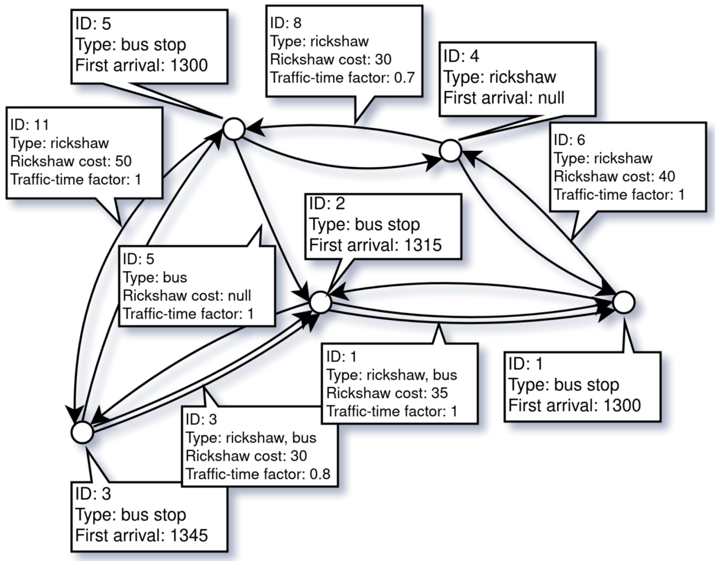

Figure 2: A sample small graph with the attributes of nodes and edges built from sample table 4 and 5.

Table 4: Sample list of nodes with their attributes.

Node ID	Location Name	Type	First Arrival Time
1	Banglamotor	bus stop	1300
2	Bashundhara City	bus stop	1315
3	Science Lab	bus stop	1345
4	Shat Moshjid	rickshaw	null
5	Dhaka University	bus stop	1300

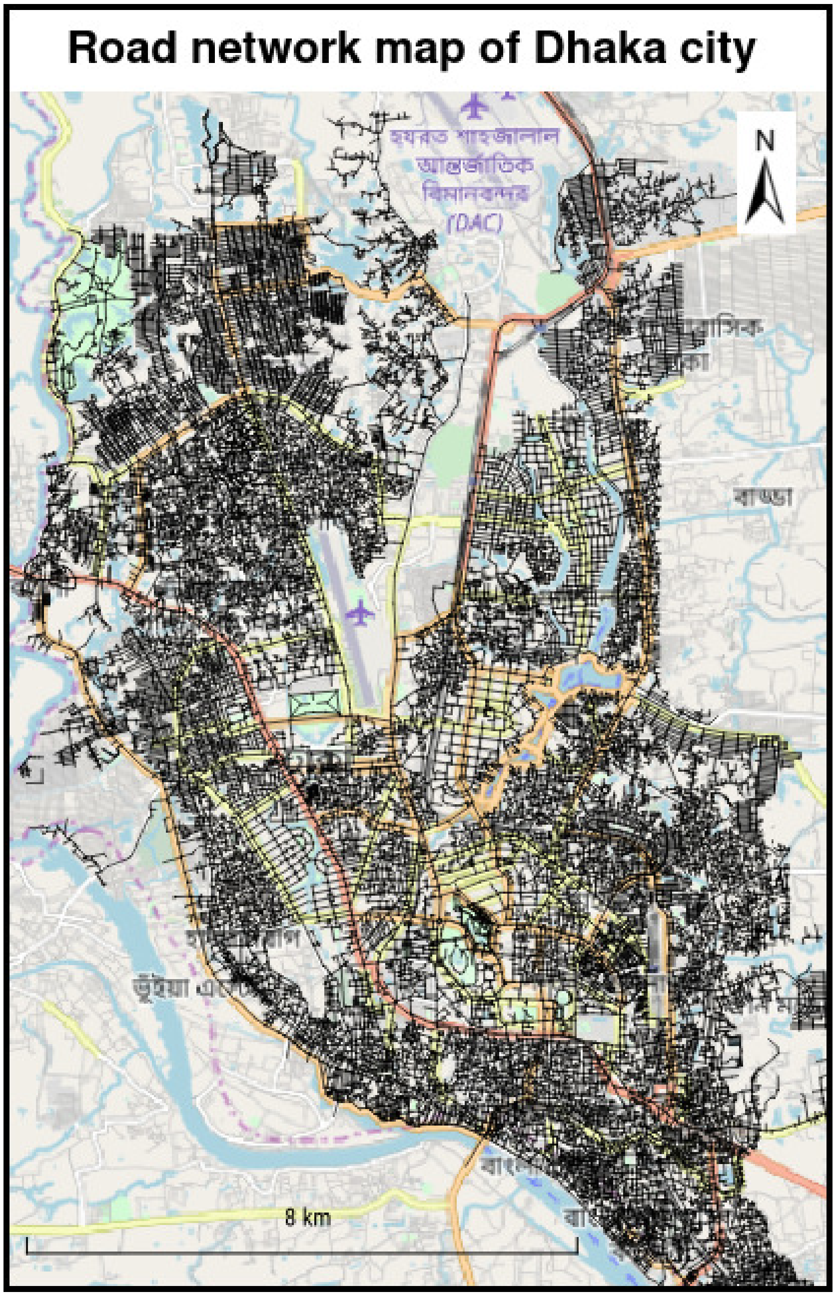

Figure 3: Complete transportation system of Dhaka city.

Table 5: Sample list of edges with their attributes.

Edge ID	**Type of route**	**Length (km)**	**Rickshaw cost (BDT)**	**Traffic-Time Parameter**
1	rickshaw	0.713	33.35	0.83
3	rickshaw	2.19	90.15	0.85
5	bus	2.52	96.27	0.91
6	rickshaw	3.11	124.03	0.76
8	rickshaw	3.70	129.32	0.81
11	rickshaw	1.45	61.17	0.89
...	...	...	...	...

usable for shortest path analysis for walking and rickshaw. The bus routes and metro routes come in a separate database which will be joined in the existing graph created for walking and rickshaw.

The bus and metro routes are similarly added to the graph to each other. The routes are firstly converted into nodes and edges just like the conversion of walking and rickshaw transportation graph. The nodes of the bus transportation graph must be a subset of the nodes of walking and rickshaw transportation graph because the roads where bus rides take place are also available for walking in Bangladesh. But this is not true for edges of the graphs. Because there might be a junction of roads which will turn into a node in walking transportation graph, but that node may not be a stoppage for a bus. The same reason also applies to metro transportation graph.

4.3. Merging of Graphs

The three newly created graphs are needed to be merged. The merging of the nodes of the transportation graphs will be edge-based. For each edge

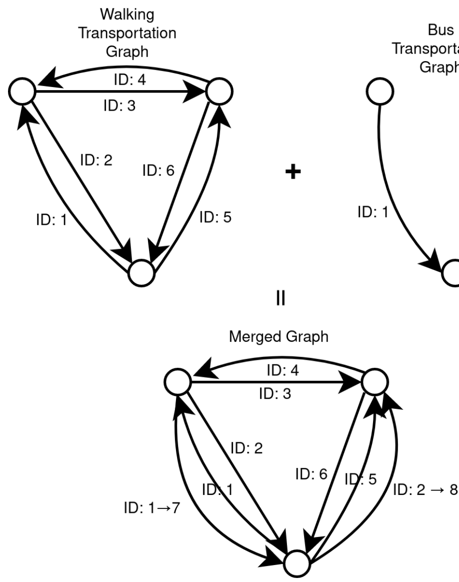

Figure 4: Graph construction through merging of two graphs.

of the bus transportation graph, a new edge will be added to the walking transportation graph. The existing edges of the walking transportation graph will not be used by modification as this is a multi-edged graph. The metro transportation graph will also be merged exactly in this manner. Figure 4 demonstrates the process of merging two seperate graphs.

5. Road Density Analysis

Road density analysis conveys important statistical and metric data at macro level road network generalization [28, 29]. There are numerous techniques to analyze the density of roads in a group of polygon areas. Firstly, the roads are cut into intersections with polygon's edges. Then in each polygon, the sum of lengths of major roads is calculated. In this paper, density analysis provides a very important relationship between cost, especially for rickshaw, and time in each administrative region of Dhaka city. In figure 5, the result of road density analysis is visualized.

The result shows that the southern regions have more roads than the eastern regions. Also, the northern regions are bigger than the western regions. Because of the small sizes of wards in the west of Dhaka city, the road density per ward is low.

Rickshaw's expense shows a negative relationship with road density. The wards with more road density tend to have a lower cost. During data collection, the authors have encountered that the bigger regions with less road density have less number of rickshaws and more costly. These insights can be used during interpolation of cost of rickshaw instead of just using simple interpolation to get estimated rickshaw cost of an unknown road that is left for future work.

6. Shortest Path Analysis

Before applying the path-finding algorithm, the graph is preprocessed in some phases. As the graph contains a very high number of nodes and edges, the preprocessing is required to lessen the pathfinding complexity.

Firstly, the graph is preprocessed irrespective of the source and destination location. This preprocessing contains the arrival time of bus or metro at each node starting from a given time. This depends on the start time and the first time the bus services start. After this preprocessing, each node will have a set of time, when the bus and metro arrive if it is a bus or metro station. Another preprocessing is the query of traffic time parameter from an API.

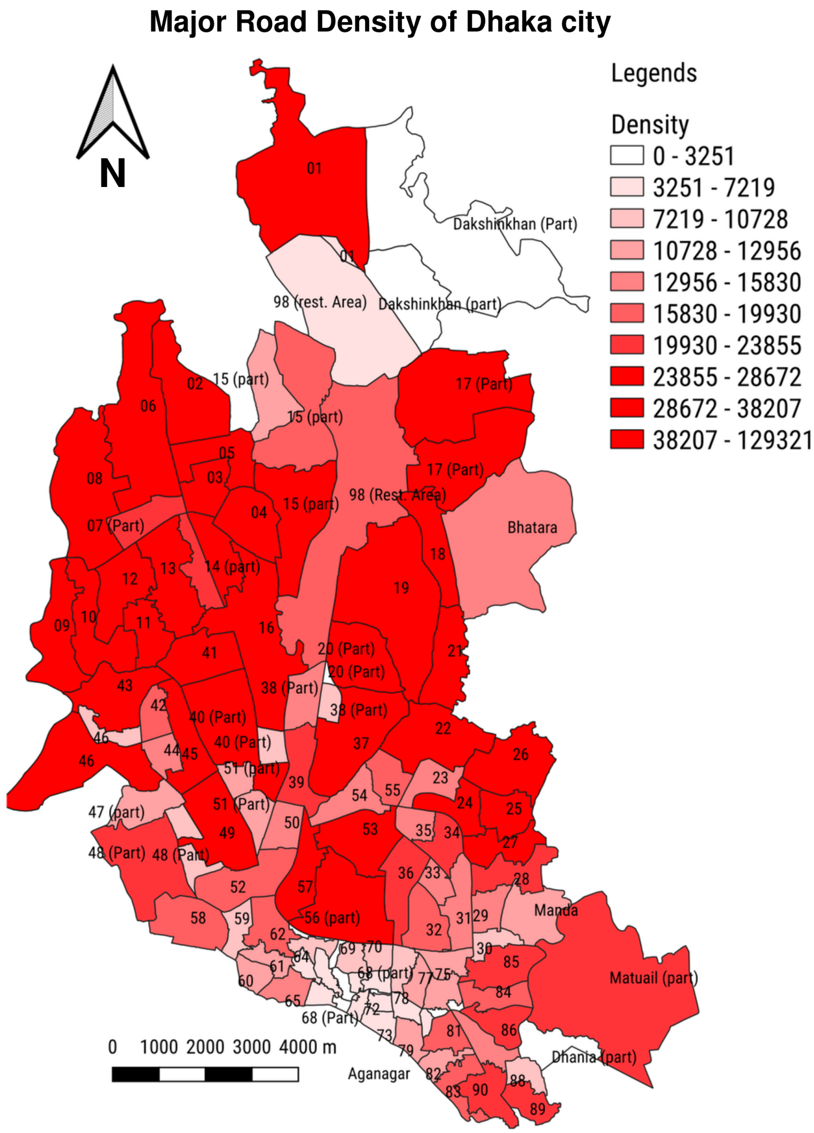

Figure 5: Road density on each ward regions represented by different shades of red color.

Secondly, the graph is augmented from the information on the source and destination location. Depending on the application of the algorithm in software, the source and destination might not be an already existing node in the graph. For example, the source or destination might be inside a house which is not a road, thus not a node or not on top of an edge in the graph. Even if the source or destination position is on an edge, it might not be on a node, instead, it is in the middle of an edge. For this reason, the source location and the destination location are converted to nodes with location values (latitude, longitude), and added to the graph. The closest edge to the source or destination point is determined. Then a point closest to the source or destination point on that edge is detected. This closest point will be considered as a node and added to the graph. This node and the source or destination node will be connected with a new edge. Finally, the closest node on the closest edge to the source or destination will be connected to the two nodes of the edge. All the parameters of this edge will be calculated from the existing data. A demonstration of this process is shown on figure 6.

Finally, real-time changing data is considered in the network. The only real-time changing data is time in the network. Time may vary depending on the traffic of the path and the arrival rate of the transports. The passenger may need to wait on a single place or node for a bus or metro, rather than moving by walking or with rickshaw for a cheaper cost. The graph can further be optimized or added more parameter to getting more effective and realistic space-time modeling of traffic flow and arrival time of bus or metro in each node with help of Yiannis Kamarianakis and Poulicos Prastacos [30]. Algorithm 1 is used to get the first arrival time of transport after a certain

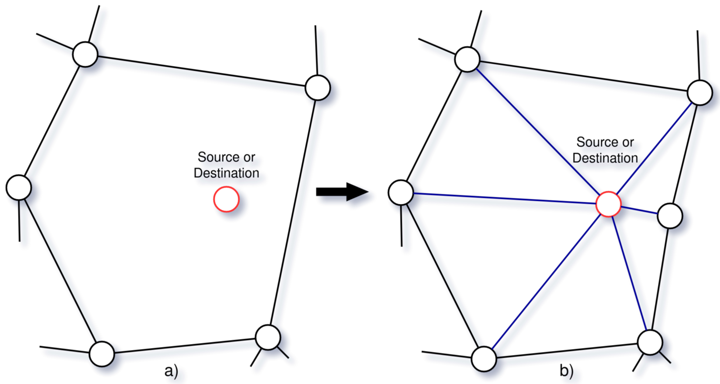

Figure 6: New node creation and graph augmentation if the source or destination location is not a node in the graph.

time at a specific place. The function assumes like Bangladesh that, buses or metro transports come from a starting node after N minutes interval, where N can be varied from 15 minutes to 45 minutes. By calculating the time taken due to traffic of the road with respective data, and speed of the transport, the function returns the next time a bus or train will arrive at a given node. The function is given on Algorithm 1. The function also considers a situation where one is at a starting point of a station, but the waiting time of the bus or metro transport is high. But a bus starting from another stoppage will come at the current node earlier. In that case, the earliest bus occurring time in the node is returned from the function.

Algorithm 1: Calculate next arrival time to a given node.

Data: Graph, Starting nodes of bus or metro, interval of bus leaving source node

```
Function nextarrival(node, current_time, busormetro):
    time[node] ← 0
    add node to priority queue q with highest priority value
    if node is starting node for busormetro then
        waiting_time ← calculate from current_time + t and interval
        t ← waiting_time
        add node to priority queue q with t as priority value
    end
    while q not empty do
        (p, t) ← pop most prior node from q and its priority
        if p is starting node of busormetro then
            return current_time + t
        end
        foreach incident edge e on p if type(e) = busormetro do
            n ← source node of edge e
            time[n] ← traffic time of e + t
            add n to q with t as priority value
            if n is starting node for busormetro then
                waiting_time ← calculate from (current_time + t and
                 interval)
                // Adding same node with waiting
                t ← t + waiting_time
                add n to q with t as priority value
            end
        end
    end
    return busormetro does not come at node
end
```

Two shortest path generating algorithms are used for the analysis. The first one is Uniform Cost Search (UCS) algorithm [31]. The second one is a modified version of the heuristic A* algorithm [32]. In the following subsections 6.1 and 6.2, the application of these two algorithms is explained.

6.1. Uniform Cost Search

Dijkstra algorithm works on a multi-edged directed graph. On this application, uniform cost searching, Dijkstra's modified algorithm has been applied. This similar algorithm was used to determine the shortest paths in Europe and the USA and it took less than one millisecond.

The algorithm used is presented on 2. The presented algorithm only applies to the cheapest pathfinding. From line 1 to 18, the initialization part is done. If there is no limit to walking distance in the traveling, the algorithm will always return cost 0 for the whole path and conclude to walk the whole path. That is why a walking distance limit is used. For each road, it is checked if the person has walked more than the limit on line 32. The algorithm will, by its nature, always check for the metro transport first, as it has the lowest cost and will be on top of the priority queue most of the time.

Algorithm 2: Cheapest Path Uniform Cost Search

```
Data:
  Graph: Merged graph nodes and edges with all attributes
  rate_b: Cost of bus per unit length
  rate_m: Cost of metro per unit length
  speed_w: Speed of walking
  speed_r: Speed of rickshaw
  speed_b: Speed of bus
  speed_m: Speed of metro
Operators:
  A ∼i B: Length of ith edge from node A to B
Functions:
  nextarrival(start, current[p], busormetro): Algorithm 1
  rate_r(p, q, i): Rickshaw cost of ith edge from node p to q
Input:
  start_place
  max_w
  current_time
Result: Cheapest path, Time at each place, Total Cost

foreach node n ← Graph do
    time[n][n] ← 0
    type[n][n] ← wait
    arrival[n] ← ∞
    parent[n] ← null
    cost[n] ← ∞
    current[n] ← ∞
    t_walked[n] ← ∞  // Total walked to reach this node
end
for each node n1, n2 of edge in Graph do
    time[n1][n2] ← ∞
    type[n1][n2] ← null
end
current[start_place] ← current_time
t_walked[start_place] ← 0
if start_place is incident node of edge type bus or metro then
    arrival[start_place] ←
      nextarrival(start_place, current[start_place], bus or metro)
end
insert start_place into the priority queue with priority 1, q
if start_place is out of graph then
    throw error
end
// Else continue until the priority queue is empty
while q not empty do
    p ← dequeue max prior element of q
    if p is goal then
        print path with transport type
        print total cost
        print time taken at each node
        exit
    end
    for each incident c of all i edges from p do
        if type(p, c) = walking and
          t_walked[p] + (p ∼i c) <
          min(max_w, t_walked[c]) and
          cost[p] < cost[c] then
            time[p][c] ← (p ∼i c) * speed_w
            type[p][c] ← walk
            current[c] ← current[p] + time[p][c]
            t_walked[c] ← t_walked[p] + (p ∼i c)
            parent[c] ← p
            cost[c] ← cost[p]
            // walk later if it becomes cheaper
            add c to q with highest priority value
        end
        if type(p, c) = rickshaw and
          cost[p] + rate_r(p, c, i) * (p ∼i c) < cost[c]
          then
            time[p][c] ← (p ∼i
              c) * speed_r * traffic_param(p, c)
            type[p][c] ← rickshaw
            current[c] ← current[p] + time[p][c]
            parent[c] ← p
            cost[c] ← cost[p] + rate_r(p, c, i) * (p ∼i c)
            add c to q with priority value cost[c]
        end
        if type(p, c) = bus or metro and
          cost[p] + rate_b or rate_b * (p ∼i c) < cost[c]
          then
            time[p][p] ←
              nextarrival(start, current[p], busormetro)
            time[p][c] ← (p ∼i
              c) * speed_b or speed_m * traffic_param
            type[p][c] ← bus or metro
            current[c] ←
              current[p] + time[p][p] + time[p][c]
            parent[c] ← p
            cost[c] ←
              cost[p] + rate_b or rate_m * (p ∼i c)
            add c to q with priority value cost[c]
        end
    end
end
```

For the fastest route finding, algorithm 2 is slightly modified. Instead of pushing nodes with cost as priority on line 39, 47 and 56, the nodes will be pushed with time as priority. Then the nodes will be explored in order of how fast one can reach there. Besides some more slight modifications, the walking distance limit is not required for the fastest path searching. As walking is the slowest of all transportation, it will be explored at the last of all other edges.

6.2. A Algorithm*

A* algorithm is a heuristic search algorithm. It is one of the most popular algorithms for graph traversing and pathfinding when a heuristic can be used. For geographical pathfinding or road analysis straight line distance can be defined easily. This makes A* algorithm perfect for application in the current scenario.

Almost in all the cases, A* algorithm will be equivalent to uniform cost search algorithm. The time consumed by the A* pathfinding algorithm will always be less than uniform cost search algorithm. The reason for this is, uniform cost search will expand in all directions from the source. Even if the path leads to opposing direction from source to destination location, uniform cost search will take it into account. But in A* algorithm, the straight line distance is used as a heuristic, which helps to provide more priority to the paths that go toward to destination.

A major problem in A* algorithm is its optimality. A* algorithm might not provide the optimal path whereas uniform cost search or Dijkstra algorithm will always provide the shortest path in the graph. For maps or geographical data, best heuristics are always straight line distance, which is used in the experiments. All the results obtained during the experiments were always the shortest path in both cases of cost and time.

7. Experiments and Results

Sevaral dataset are used for experiment with the developed algorithms. On subsection 7.1, these dataset are discussed. Then the implementation details are discussed on subsection 7.2. Finally the result along with the time taken by the algorithms to achieve the results are discussed in subsection 7.3.

7.1. Dataset Preparation

The dataset used for all the experiments is taken from geodash.gov.bd [33]. On the website, there are private datasets for all the bus stands. There are also bus routes and road maps of the country, especially detailed maps of Dhaka city. The datasets can be downloaded as various file formats like shapefile [34], geojson [35], CSV, kml etc. As shapefiles are standard for geographical data storage and processing, it is selected for the experiments.

The metro routes and stoppages dataset is obtained from a hackathon named Code Samurai 2019 [36]. In the hackathon, the authority has made public a road map of Dhaka city, a list of bus stoppages and metro stations. The road map of Dhaka city is almost identical to the road map in OSM street map [37] and the maps provided in geodash. The dataset was in CSV file format. So the data files are read and converted to shapefiles to insert in the experiment pipelines.

7.2. Implementation Details

Python programming language was used to implement all the algorithms. Fiona python library was used to manage the epsg code standards [38] and all the in-between conversions. osmnx package [39] was used to collect some roads and geographic information. Geopandas [40], momepy [41] and shapely [42] was used to manipulate, query and preprocess the collected dataset. Finally, graph implementation and algorithms were applied with the popular networkx python package [43]. QGIS application [44] was used to visualize the dataset and cross-validate the results.

The dataset was collected and stored in shapefile format. Then the graph was built with the help of networkx. The same package was used to preprocess the dataset and shapely was used to calculate the straight line distance to destination in case of A* algorithm. Finally, pre and post data analysis was done with the combination of python in jupyter notebook and QGIS application. The whole process was done on a custom-built PC having Intel Core i7, 3.5 GHz, 32 GB RAM and NVIDIA GeForce GTX 1080 Ti GPU.

7.3. Result Analysis

In section 2, it is seen that GIS application in Dhaka city is very rare. The authors have collected most of the datasets of rickshaw and CNG costs by surveying. The complete routes demonstrated on figure 3 has 49869 nodes and 53878 edges. There are two datasets collected for the bus transportation system, Uttara bus service, and Bikolpo bus service. The number of nodes and edges of the Uttara bus service is 18 and 17. For the Bikolpo bus service, it is 20 and 19. Lastly, the metro rail dataset consists of 15 nodes and 14 edges. The time it took to merge the walking transportation graph and bus transportation graph was 1.83 seconds. After this, the time taken to merge it with the metro transportation graph was 1.81 seconds.

The road density analysis provides an insight that regions that have more roads tend to have lower rickshaw costs. This result will be useful to estimate the cost of a rickshaw at an unknown place as it is tedious to inquire and get all the costs of all the roads.

The new proposed algorithm for the shortest route analysis has worked well on some test source and destination places. Figure 7 shows as example of the cheapest path using algorithm 2. Figure 8 demonstrates the fastest path applied in the same scenarios where the time of arrival of buses and metro is calculated with algorithm 1 with 15 minutes interval.

8. Discussion and Conclusion

The methodology proposed in this research has tried to create a traveling solution that will better fit in Dhaka city. Despite being accurate and optimized, the proposed system has still some limitations.

The bus routine is assumed to be starting in all assigned places at the same time. This is assumed due to a lack of data from different places and can be easily altered to accept different starting times in different places.

The cost of rickshaw is assumed to be fixed once calculated for all places. But it can be dependent on the time of year. This is left for the future work of this research.

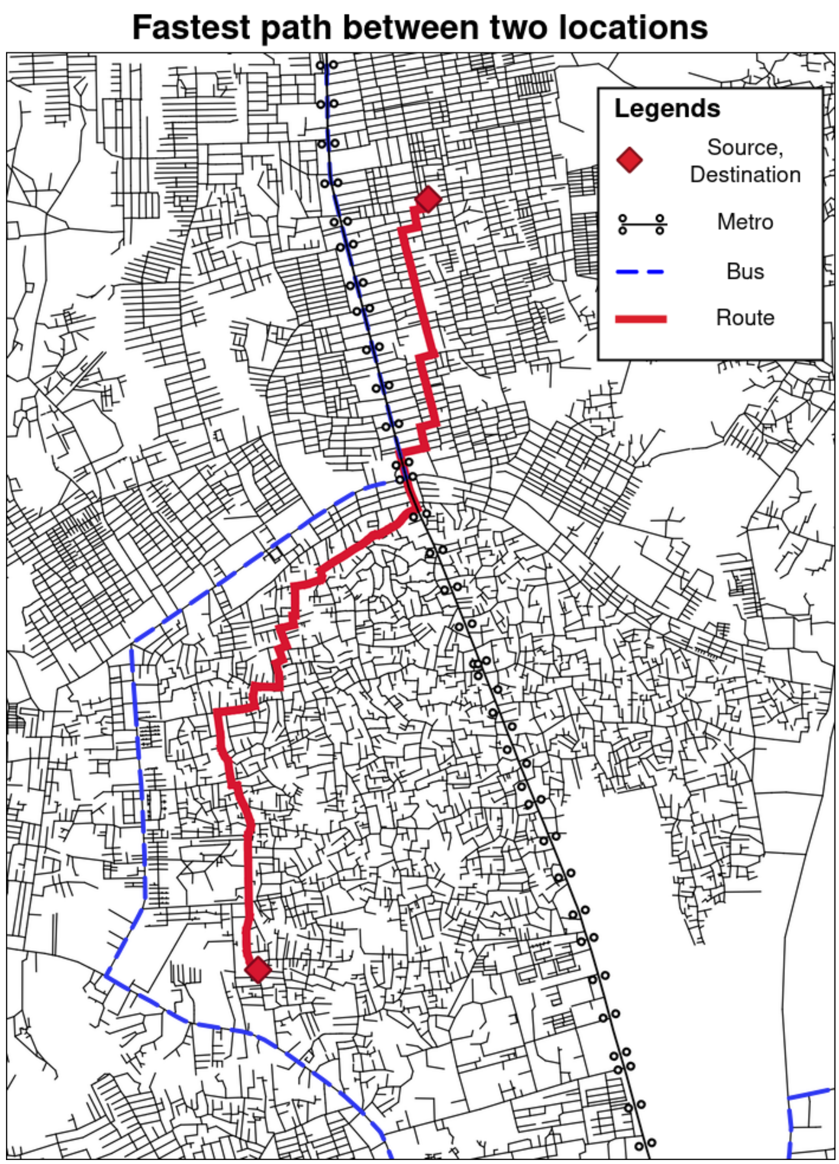

Figure 7: Fastest path between two locations (shown in red color)

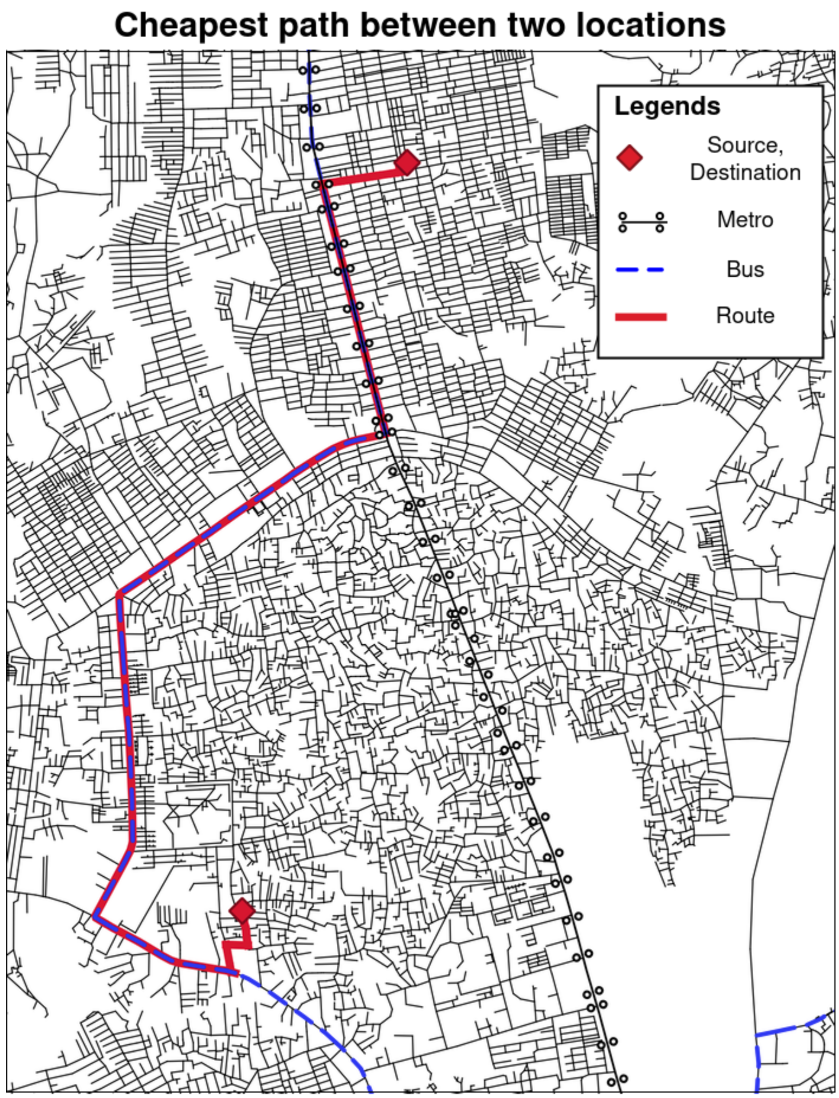

Figure 8: Cheapest path between two locations (shown in red color)

9. Acknowledgement

First of all, the authors express their sincere appreciation and heartfelt thanks to everyone who has inspired them for this paper's contributions. The authors would also like to thank all the people who helped them locate and collect the traffic datasets for the analysis. Finally, the authors want to thank this paper's reviewers and the researchers in the field of Geo-science for contributing to modern world knowledge.

10. Computer Code Availability

All the implementation code used in the research are open source and can be found on https://github.com/rahatzamancse/gis-analysis-of-dhaka-city-roads which is under user name of one of the authors Md. Rahat-uz-Zaman. The code can be run on any machine with proper python environment and qgis application as mentioned in the readme of the repository. Also, the dataset along with its proper explanation can be found in the published dataset Md. Rahat-uz Zaman [45] on Mendeley. Instructions to reproduce and analyze of data are also available in the public code repository.

References

[1] M. M. Hoque, The road to road safety: issues and initiatives in bangladesh, in: Regional Health Forum, volume 8, pp. 39–51.

[2] M. M. Hoque, B. Jobair, Strategies for safer and sustainable urban transport in bangladesh, Proceedings of CODATU X (2002).

[3] M. M. P. Rana, Urbanization and sustainability: challenges and strategies for sustainable urban development in bangladesh, Environment, Development and Sustainability 13 (2011) 237–256.

[4] R. Shamsher, M. N. Abdullah, Traffic congestion in bangladesh-causes and solutions: a study of chittagong metropolitan city, Asian Business Review 2 (2015) 13–18.

[5] A. Chowdhury, K. Kishi, K. Satoh, A study on highway logistics problem from dhaka to chittagong in bangladesh, in: Japan Society of Civil Engineers.

[6] E. Jenelius, T. Petersen, L.-G. Mattsson, Importance and exposure in road network vulnerability analysis, Transportation Research Part A: Policy and Practice 40 (2006) 537–560.

[7] W. Rees, Least-cost paths in mountainous terrain, Computers & Geosciences 30 (2004) 203–209.

[8] S. Ahmed, R. F. Ibrahim, H. A. Hefny, Gis-based network analysis for the roads network of the greater cairo area, in: Proc. of 2nd International Conference on Applied Research in Computer Science and Engineering.

[9] Y. Luping, Q. Dalin, Vulnerability analysis of road networks, Journal of Transportation Systems Engineering and Information Technology 12 (2012) 105–110.

[10] D. Das, A. K. Ojha, H. Kramsapi, P. P. Baruah, M. K. Dutta, Road network analysis of guwahati city using gis, SN Applied Sciences 1 (2019) 906.

[11] E. 2011, Arcgis desktop, ????

[12] A. Quium, S. Hoque, The completeness and vulnerability of road network in bangladesh, Engineering Concerns of Flood (2002) 59–75.

[13] Q. Li, Variable-scale representation of road networks on small mobile devices, Computers & geosciences 35 (2009) 2185–2190.

[14] A. A. Sufian, S. M. Khan, I. Ahmed, S. Islam, N. Saha, Safety analysis: observed deficiencies in existing transportation system of bangladesh, International Journal of Urban Planning and Transportation 27 (2016).

[15] R. Begum, S. N. Momotaz, Determinants of passengers'satisfaction with cng-run auto rickshaw services in bangladesh: An empirical study on dhaka city, DU Journal of Marketing 15 (2012) 159–174.

[16] H. M. Ahsan, et al., Study of mass transit in metropolitan dhaka (1990).

[17] B. Ahmed, The traditional four steps transportation modeling using a simplified transport network: A case study of dhaka city, bangladesh, International Journal of Advanced Scientific Engineering and Technological Research 1 (2012) 19–40.

[18] K. M. A. Waheed, Study on application of cut and cover method for the construction of metro rail tunnel in dhaka city (2008).

[19] M. S. Alam, Factors in deciding metro rail in developing countries: A study on the proposed metro rail system for dhaka, Journal of Bangladesh Institute of Planners ISSN 2075 (2010) 9363.

[20] M. M. Hoque, S. Barua, H. Ahsan, D. Alam, Brt in metro dhaka: towards achieving a sustainable urban public transport system, Proceedings of CODATU XV: The Role of Urban Mobility in (re) shaping Cities (2012) 1–11.

[21] M. Hossain, Y. O. Susilo, Rickshaw use and social impacts in dhaka, bangladesh, Transportation research record 2239 (2011) 74–83.

[22] S. Begum, B. Sen, Pulling rickshaws in the city of dhaka: a way out of poverty?, Environment and Urbanization 17 (2005) 11–25.

[23] S. I. Khan, A. Khan, M. N. I. Sarker, N. Huda, M. R. Zaman, A. Nurullah, M. Z. Rahman, Traffic congestion in dhaka city: Suffering for city dwellers and challenges for sustainable development, European Journal of Social Sciences 57 (2018) 116–127.

[24] K. R. Rahaman, N. Ohmori, N. Harata, Evaluation of the roadside walkway environment of dhaka city, in: Proceeding of the Eastern Asia Society for Transportation Studies, volume 5, pp. 1751–1766.

[25] M. S.-U. Rahman, K. Nahrin, Bus services in dhaka city-users' experiences and opinions, Journal of Bangladesh Institute of Planners ISSN 2075 (2012) 9363.

[26] J. W. Essam, M. E. Fisher, Some basic definitions in graph theory, Reviews of Modern Physics 42 (1970) 271.

[27] S. M. Khan, M. S. Hoque, Traffic flow interruptions in dhaka city: Is smooth traffic flow possible, Journal of Presidency University 2 (2013) 46–54.

[28] Q. Zhang, Road network generalization based on connection analysis, in: Developments in spatial data handling, Springer, 2005, pp. 343–353.

[29] Z. Zhang, S. Tan, W. Tang, A gis-based spatial analysis of housing price and road density in proximity to urban lakes in wuhan city, china, Chinese geographical science 25 (2015) 775–790.

[30] Y. Kamarianakis, P. Prastacos, Space–time modeling of traffic flow, Computers & Geosciences 31 (2005) 119–133.

[31] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1959) 269–271.

[32] P. Hart, N. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (1968) 100–107.

[33] B. Government, GeoDASH, 2016 (accessed November 5, 2019). Url: https://geodash.gov.bd.

[34] E. ESRI, Shapefile technical description, An ESRI white paper 39 (1998).

[35] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub, et al., The geojson format, Internet Engineering Task Force (IETF) (2016).

[36] B. University of Dhaka, Code Samurai 2019, 2019. Url: https://codesamuraibd.net/.

[37] C. driven map, OSM streep map, ???? Url: https://www.openstreetmap.org/.

[38] F. Warmerdam, Coordinate systems: Proj. 4, epsg and ogc wkt, Presentation at the FOSS4G2006-Free And Open Source Software for Geoinformatics http://2006. foss4g. org/getFile. py/accessf802. pdf (2006).

[39] G. Boeing, Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks, Computers, Environment and Urban Systems 65 (2017) 126–139.

[40] K. Jordahl, Geopandas: Python tools for geographic data, URL: https://github.com/geopandas/geopandas (2014).

[41] M. Fleischmann, momepy: Urban morphology measuring toolkit, Journal of Open Source Software 4 (2019) 1807.

[42] S. Gillies, The shapely user manual, 2013.

[43] A. Hagberg, D. Schult, P. Swart, Networkx: Python software for the analysis of networks, Mathematical Modeling and Analysis, Los Alamos National Laboratory (2005).

[44] Q. D. Team, et al., Qgis geographic information system. open source geospatial foundation project, URL: http://qgis. osgeo. org (2015).

[45] S. H. Md. Rahat-uz Zaman, Shortest route analysis of dhaka city roads using various gis techniques (dataset and sample outputs) (2020).

Highlights

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

Md. Rahat-uz-Zaman, Shadmaan Hye

- Local transport costs in road network require algorithms to estimate accurate fare.
- Shortest path algorithms fail due to inconsistent fare of transportation rates.
- A* algorithm performs comparatively better for both cheap and shortest path analysis.
- Many transports can be merged with parameters to create finer direction support.

Highlights

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

Md. Rahat-uz-Zaman, Shadmaan Hye

- Local transport costs in road network require algorithms to estimate accurate fare.
- Shortest path algorithms fail due to inconsistent fare of transportation rates.
- A* algorithm performs comparatively better for both cheap and shortest path analysis.
- Many transports can be merged with parameters to create finer direction support.

Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques

Md. Rahat-uz-Zaman[1,*], Shadmaan Hye[2]

Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Abstract

History clearly shows from the former times to recent times that the development of a region is heavily contingent on the transport network. Nowadays, however, there are a number of network transport problems, including increased operational costs for vehicles, delay, road congestion, pollution, and road accidents, which are becoming the major issues in a developing country like Bangladesh. Digitized analysis and strategy formulation on constructing, maintaining and upgrading are crucial to get the greatest available solution for these problems. This research contributes towards this analysis by creation, evaluation and compilation maps from obtained information. The present study made an effort to prepare road networks and application of several Geographic Information System (GIS) techniques on the pedestrian, rickshaw, metro and public transit transports systems and receive the opti-

[*]Corresponding Author
Email addresses: rahatzamancse@gmail.com (Md. Rahat-uz-Zaman), praptishadmaan@gmail.com (Shadmaan Hye)

[1]Conceptualization of this study, Writing - Original draft preparation, Methodology, Software, Algorithm Design, Implementation

[2]Data curation, Writing - Review and Revision, Computational resource collection, Implementation

mum transport solution on a variety of cost metrics like traversing distance, velocity and time. Another important discovery made in this research is the relationship between regional road density to cost and travel time. This article demonstrated that in Dhaka city area, based on efficient shortest path algorithm design and a graph generated with 49869 nodes and 53878 edges, the shortest distance between two points may not always be cost-effective and conversely.

Keywords: GIS, Road Density Analysis, Shortest Path, A* algorithm, Uniform Cost Search, Fare Estimation,

---

1. Introduction

Geographic Information System (GIS) technology is one of the fastest-growing road network technologies and the most widely utilized method for science in the world. GIS can capture all the necessary data, store it, manipulate, analyze, present and quickly visualize for the required effective solution of the user.

In a densely populated city center such as Dhaka (DHK) Bangladesh, several traffic queues alongside high transportation costs are traffic congestion, limited utility of broad roadways, a complex travel network, a comparatively higher number of road accidents Hoque [1], rapid urbanization triggering unevenly wide roads [2, 3], traffic concerned problem [4] and the problem of highway logistics [5]. For all emergency transport, it is proved that the main criteria is its readiness to reach any destination as fast as possible or at least within a fixed period [6]. The key aim of this groundwork is to provide enhanced network analysis and algorithms focused on certain cost parameters such as time and money.

In the following sections, firstly related works done by other researchers are addresses in 2. Section 3 specifies initial assumptions for the method. Then the methodology itself is discussed from section 4 to 6. Then it is extended to some real collected data, which is covered by subsection 7.1. Finally, the results of the implementation of this research on the datasets are defined in subsection 7.3.

2. Literature Review

A well amount of work on the transportation system is done recently in different countries in the world. The works include shortest path analysis, cost analysis, algorithm development and suggestions of increasing efficiency of traffic systems.

W.G. Rees [7] has used Dijkstra's algorithm to find the least-cost path in mountainous terrain of Wales. They have built the cost function from the topography alone and only considered footpaths. They were successful to build a foot-path map of Wales and determined the divergence of their path with the true straight line. In this paper, Uniform Cost Search algorithm, which is very similar to Dijkstra has been used to solve very similar objective. But a greater number of means of transportation and cost is considered.

S. Ahmed et al. [8] has worked on the Greater Cairo Region (GCR) to find the correct destination during an emergency. He has also proposed a method based on GIS to detect the shortest path from an emergency situation to the required destination. He also has applied closest facilities analysis on the region and achieved impressive results. The lack of various transportation methods such as bus, metro was a major limitation for their research.

L. Yang and D. Qian [9] presented a method vulnerability evaluation which is helpful to examine network vulnerability through final travel time cost and the structure of the network. They have conducted a simple case study on urban traffic and found a method to recover the traffic damaged due to roadblocks. But their most effort was only to detect the vulnerability and find alternate solutions if a road is blocked for any reason.

D. Das, A. Ojha, H. Kramsapi et al. [10] has prepared road transportation network for Guwahati city and did a similar work like S. Ahmed et al. With proper shortest path analysis and digitization, ArcGIS software [11] is used to determine the shortest path from a preselected source to required destination. Their result shows an impressive collection of data and visualization like road network map, ward map, junction and edges map, density map of both major and minor roads, lastly lane density. The analysis was very specific to Guwahati city and some of those cannot be applied to other regions like Bangladesh.

A. Quium and S. Hoque [12] has analyzed the vulnerability of road network in Bangladesh with the help of some indexing such as alpha (α) and gamma (γ). They have also measured the density of national and regional road networks and found that Dhaka city has more than average density of road in Bangladesh compared to other districts. But most of their works are based on inter-district national roads instead of regional roads inside a city. They have neither done any network analysis nor shortest path analysis which is done in this paper.

Qingquan Li [13] has proposed a technique to represent a road network on tiny mobile devices. The interesting part of the adaptive multi-scaled road network representation is that, the method can automatically set a feasible scale to geographic scope and other parameters. This can be used as the representation of this proposals application for different parameter e.g. travel by vehicle type.

In all the above-mentioned works, an application-focused proposal was not present or only representation of network is proposed in some works. In this paper, the authors try to use their proposal on applications so that the users may wish to analyze and detect the cheapest or fastest route in Dhaka city. Also unlike others, this methodology can be used for more and different vehicles which made the research work novel.

3. Transportation System of Bangladesh

This paper has proposed a suitable route analysis from a source location to a destination location based on the transportation system of Dhaka city. Because there are a lot of deficiencies in the existing transportations system in Bangladesh [14], usual road transport analysis developed for other countries will not work well in this country. In this country, there are 2 unique vehicles which are rickshaw and auto-rickshaw (CNG). As they are similar in size, shape and availability [15], in this research, they are considered of the same class vehicle. Though the fair of these two classes are very different, only cost of rickshaw is surveyed and taken into account for the research. Further mention of rickshaw in this paper will indicate both rickshaw and CNG. The common means of transportation are discussed in subsection 3.1.

3.1. Common Means of Transportation

There are various transportation systems in Bangladesh. Among those transportation systems, the most popular means of transportation in Dhaka city [16, 17] are discussed below.

1. **Metro route** [18, 19]: Highly cost and time-efficient. But only one route available in the whole city. Passengers need to wait at the station for the arrival time of the metro.
2. **Bus route** [20]: Cost-efficient and fast transportation. Passengers need to wait less time than the metro for the arrival of the bus at the bus stoppage.
3. **Rickshaw** [21, 22]: Costly and takes time to travel far distance. But always available on almost all roads except VIP roads. Most people choose rickshaw for short-distance transportation [23].
4. **Walking** [24]: Preferred for feasibly short distance. It takes the highest time but expense-free and can walk almost everywhere.

4. Building Graph from Geo-spatial Data

For the analysis of transportation, a graph from geo-spatial data is necessary. Now to build the graph, the costs of transportations are estimated and then with the help of the costs, several graphs are created. The final graph on which the analysis is applied is found after merging all the graphs.

4.1. Costs of Transportation in Graph

The main costs that are considered in this research are time and expense. While building the graph, time taken for a path is dynamically estimated while traversing from the length of the road, the speed of the transport vehicle considered and the traffic time parameter. The expense function is dependent on the transport vehicle, source and destination location of path in the network by interpolating from a database of collected fares. While building the graph, both of these attributes discussed above are included in each edge. The attributes of each node and edge are discussed in subsection 4.2.

As metro transportation system is not complete during this research in Bangladesh, the fares of metro routes are not available. As the fare should be lesser than bus fares in Dhaka city, the authors have used a reasonable fare of BDT 5 per kilometer which satisfies the above condition and used it where needed during the research. The fare between two stations is also rounded during usage.

Another function is required for the graph building which is waiting time at a bus or metro station. Based on some research [25], it is assumed that each bus will start at a specific stoppage in the morning and will be at a certain time at each station with an interval time.

Sample of dataset collected for rickshaw, bus and metro are presented correspondingly on table 1, 2 and 3. All possible costs of bus routes can be collected with ease. But for the reason that there are numerous roads for rickshaw, it is almost impossible to collect all possible rickshaw fare for all roads. For this reason, an interpolation method is used to estimate fare of rickshaw in an unknown road which is shown in equation 1.

The working of the interpolation method is visualized in figure 1. Each edge directly connected to the edge e will contribute a weighted fare (W_i) for

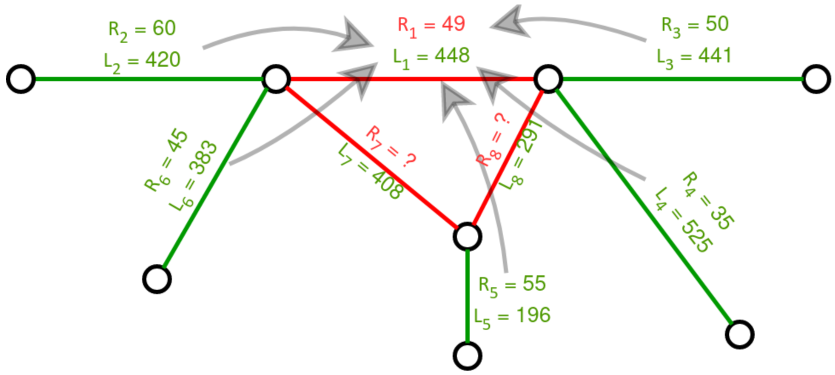

Figure 1: Demonstration of interpolation method to estimate rickshaw fare.

the estimation of fare of edge e. Here r_i is cost of rickshaw and R_i is cost of rickshaw per unit in road i with length L_i where i is a road directly connected via a node to edge e. This process is needed to apply several times as there might be some edges that are not directly connected to any edge with known fare. In these situations, the fare is estimated first for those adjacent edges of e and then are used for fare estimation of e.

$$
\begin{aligned}
S_{ei} &= \{\text{set of length of edges of shortest path} \\
&\qquad \text{from e to closest edge whose fare is known}\} \\
w_i &= \frac{1}{L_i + \sum_{l \in S_{ei}} l} \\
W_i &= \frac{w_i}{\sum_j w_j} \\
R_i &= \frac{r_i}{L_i} \\
R_e &= \sum_i W_i R_i
\end{aligned} \tag{1}
$$

Table 1: Sample of dataset of rickshaw cost survey.

Source	Destination	Cost (BDT)	Straight Line Distance (km)	Graph Traversal Distance (km)	Time of asking
Bashundhara City	Banglamotor	50	0.50	0.71	10:00
Banglamotor	Mogbazar	45	1.59	1.88	13:30
BUET Quarter	Polashi	20	0.48	0.65	16:00
Azimpur	Nilkhet	25	0.42	0.44	14:30
Shimanto Square	Sonkor	60	1.70	1.70	21:00
Katabon	Motalab Plaza	30	0.76	1.00	20:00
...	...	...	...	...	...

Table 2: Bus fare survey for several source and destination locations.

Source	Destination	Cost (BDT)	Straight Line Distance (km)	Graph Traversal Distance (km)
Azimpur	Science Laboratory	10	0.90	0.95
Asad Gate	Kalyanpur	15	2.29	2.30
College gate	Gabtoli	20	3.11	3.40
Paltan	Shahbag	15	1.75	2.10
Banglamotor	Farmgate	10	1.22	1.30
Agargaon	Mirpur 10	30	3.48	3.49
...	...	...	...	...

Table 3: Metro fare survey for several source and destination locations.

Source	**Destination**	**Cost (BDT)**	**Graph Traversal Distance (km)**
Airport	Kamalapur	76	12.65km
Uttar Badda	Rampura	19	3.1km
Jhigatala	Golap Shah Mazar	26	4.4km
Gabtoli	Purba Dasherkandi	56	9.3km
Modhumoti	Mirpur 1	43	7.1km
Kochukhet	Gulshan 2	20	3.4km
...	...	...	...

4.2. Construction of Graph

The graph [26] for the transportation system in Bangladesh will be a directed graph with dynamic self-loops and parallel edges. A small preprocessed graph is displayed on figure 2. The graph must be directed because there are one-way roads in Dhaka city. Parallel edges are needed because two close locations may have more than one road. The self-loop edges will be used as waiting time in a station of bus or metro and will be dynamic depending on the requirement of waiting at the spot while traversing.

Primarily, an empty multi-edged directed graph is created. Then for all possible combinations of two paths in the input roads data, the intersection points are taken and added as nodes to the graph. A small sample of nodes data is shown on table 4 without the latitudes and longitudes. Each node must have the following attributes.

1. **Location name**: Location name in short form.
2. **Type**: Defined if the location is a bus stop, metro station or a junction of multiple roads.

3. **First arrival time**: This should be null if the type of the node is not a bus stand or a metro station. Else the first time when the bus or metro station will be in this attribute.
4. **Latitude and Longitude**: These are kept to calculate distance between two nodes.

While adding nodes to the graph, each node pair will also be joined together with an edge. A small sample of processed edges are represented in table 5. Each edge will have the following attributes.

1. **Type of route**: Defines if the route is available for rickshaw, bus or metro.
2. **Length of road in kilometers**: Length of the road calculated from the dataset.
3. **Cost of Rickshaw**: Usual cost of rickshaw to go from one node to another using this edge. The costs are calculated using interpolations from a dataset of a survey whose sample is in table 1. The interpolation is discussed in subsection 4.1.
4. **Traffic Time Parameter**: A parameter that will define how much traffic jams in the road represented by this edge. This is an important parameter for Dhaka city because almost all the roads in this city will have traffic jams [27]. This gives a value between 0 and 1 where 0 corresponds to very high traffic jam and 1 corresponds to no traffic jam. This can be fetched at times from online APIs like google maps.

After adding all the intersections of roads as nodes and the road itself as the edge linking between two intersections, the graph will be completely

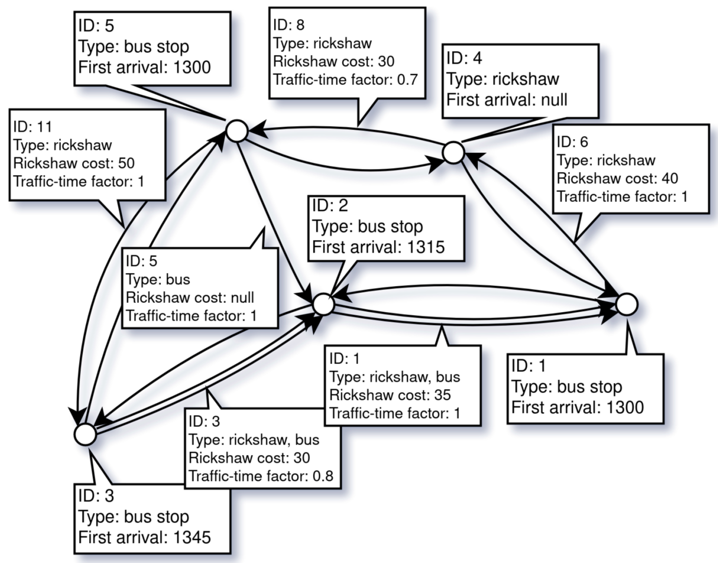

Figure 2: A sample small graph with the attributes of nodes and edges built from sample table 4 and 5.

Table 4: Sample list of nodes with their attributes.

Node ID	**Location Name**	**Type**	**First Arrival Time**
1	Banglamotor	bus stop	1300
2	Bashundhara City	bus stop	1315
3	Science Lab	bus stop	1345
4	Shat Moshjid	rickshaw	null
5	Dhaka University	bus stop	1300

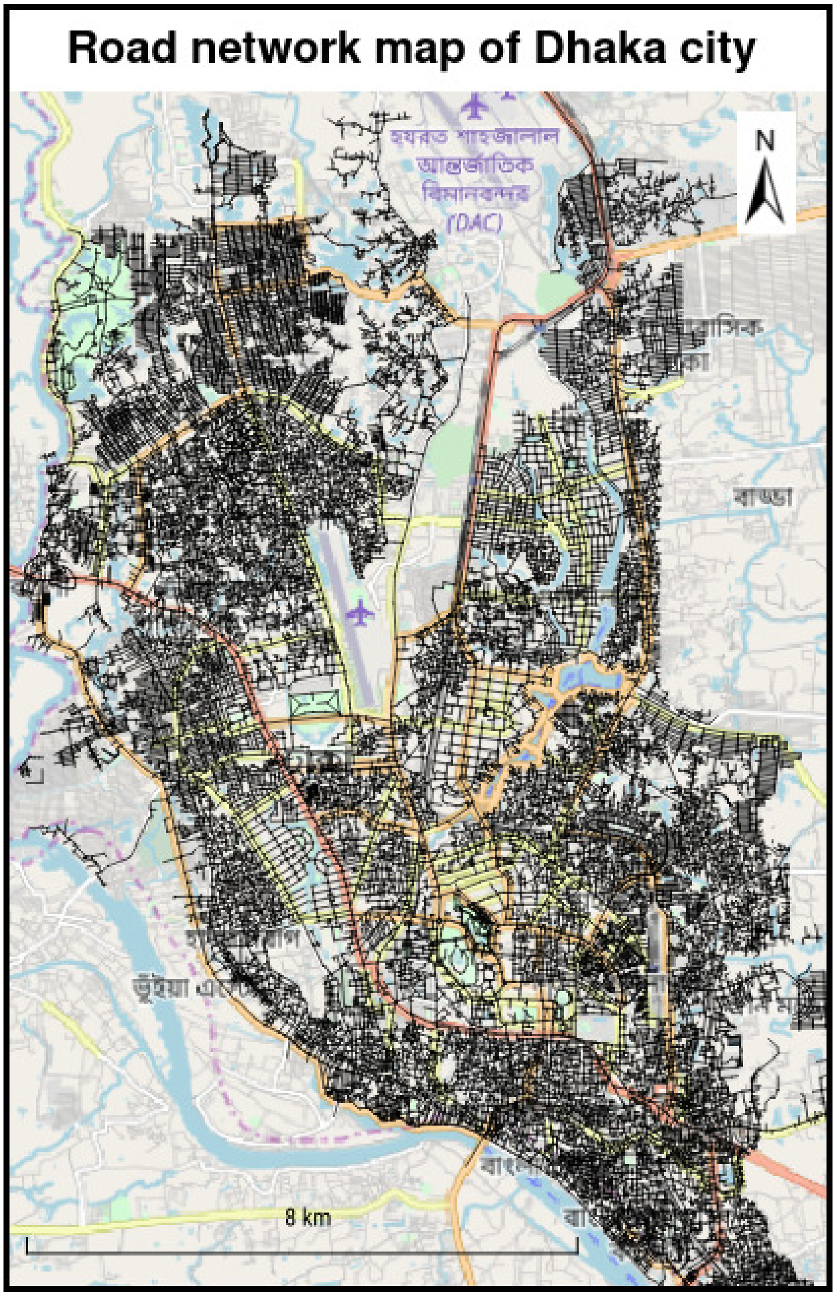

Figure 3: Complete transportation system of Dhaka city.

Table 5: Sample list of edges with their attributes.

Edge ID	Type of route	Length (km)	Rickshaw cost (BDT)	Traffic-Time Parameter
1	rickshaw	0.713	33.35	0.83
3	rickshaw	2.19	90.15	0.85
5	bus	2.52	96.27	0.91
6	rickshaw	3.11	124.03	0.76
8	rickshaw	3.70	129.32	0.81
11	rickshaw	1.45	61.17	0.89
...	...	...	...	...

usable for shortest path analysis for walking and rickshaw. The bus routes and metro routes come in a separate database which will be joined in the existing graph created for walking and rickshaw.

The bus and metro routes are similarly added to the graph to each other. The routes are firstly converted into nodes and edges just like the conversion of walking and rickshaw transportation graph. The nodes of the bus transportation graph must be a subset of the nodes of walking and rickshaw transportation graph because the roads where bus rides take place are also available for walking in Bangladesh. But this is not true for edges of the graphs. Because there might be a junction of roads which will turn into a node in walking transportation graph, but that node may not be a stoppage for a bus. The same reason also applies to metro transportation graph.

4.3. Merging of Graphs

The three newly created graphs are needed to be merged. The merging of the nodes of the transportation graphs will be edge-based. For each edge

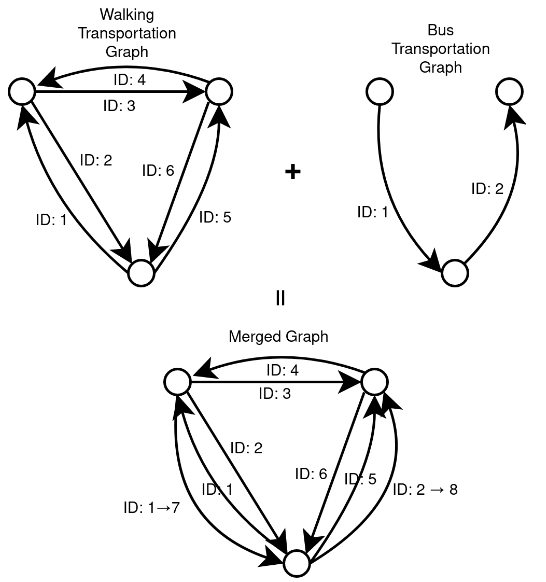

Figure 4: Graph construction through merging of two graphs.

of the bus transportation graph, a new edge will be added to the walking transportation graph. The existing edges of the walking transportation graph will not be used by modification as this is a multi-edged graph. The metro transportation graph will also be merged exactly in this manner. Figure 4 demonstrates the process of merging two seperate graphs.

5. Road Density Analysis

Road density analysis conveys important statistical and metric data at macro level road network generalization [28, 29]. There are numerous techniques to analyze the density of roads in a group of polygon areas. Firstly, the roads are cut into intersections with polygon's edges. Then in each polygon, the sum of lengths of major roads is calculated. In this paper, density analysis provides a very important relationship between cost, especially for rickshaw, and time in each administrative region of Dhaka city. In figure 5, the result of road density analysis is visualized.

The result shows that the southern regions have more roads than the eastern regions. Also, the northern regions are bigger than the western regions. Because of the small sizes of wards in the west of Dhaka city, the road density per ward is low.

Rickshaw's expense shows a negative relationship with road density. The wards with more road density tend to have a lower cost. During data collection, the authors have encountered that the bigger regions with less road density have less number of rickshaws and more costly. These insights can be used during interpolation of cost of rickshaw instead of just using simple interpolation to get estimated rickshaw cost of an unknown road that is left for future work.

6. Shortest Path Analysis

Before applying the path-finding algorithm, the graph is preprocessed in some phases. As the graph contains a very high number of nodes and edges, the preprocessing is required to lessen the pathfinding complexity.

Firstly, the graph is preprocessed irrespective of the source and destination location. This preprocessing contains the arrival time of bus or metro at each node starting from a given time. This depends on the start time and the first time the bus services start. After this preprocessing, each node will have a set of time, when the bus and metro arrive if it is a bus or metro station. Another preprocessing is the query of traffic time parameter from an API.

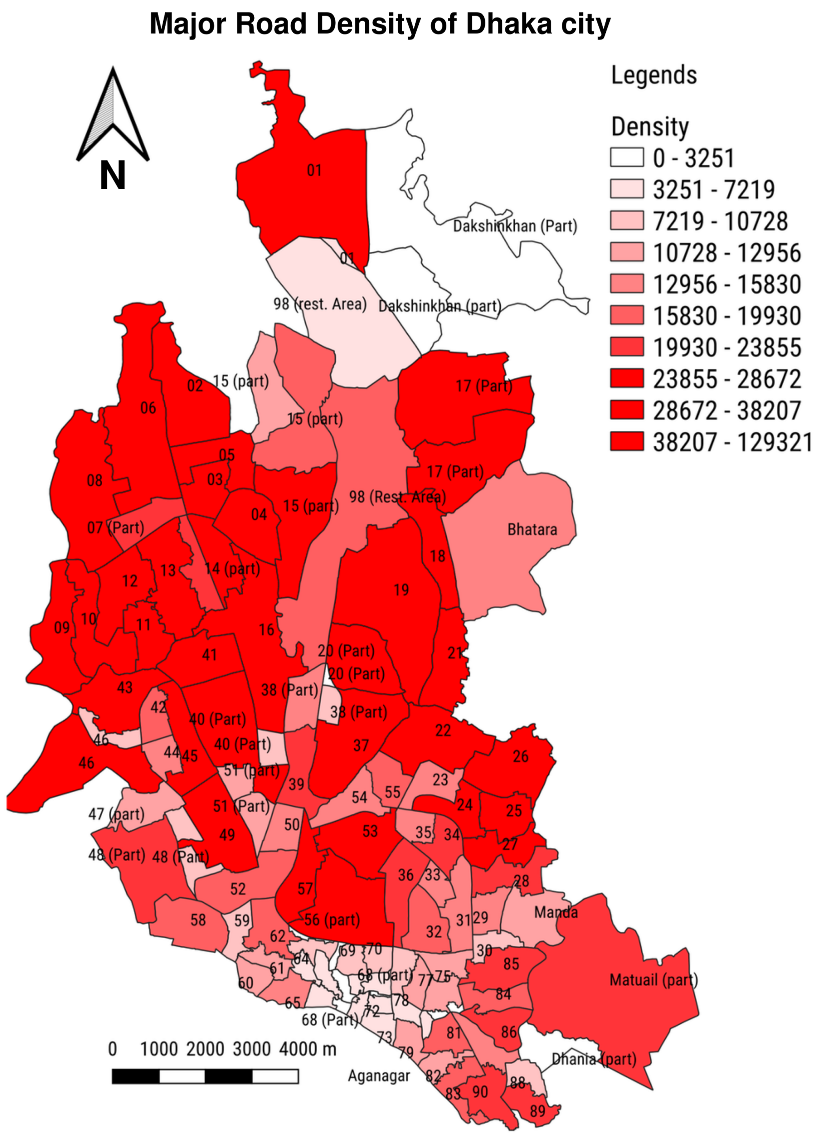

Figure 5: Road density on each ward regions represented by different shades of red color.

Secondly, the graph is augmented from the information on the source and destination location. Depending on the application of the algorithm in software, the source and destination might not be an already existing node in the graph. For example, the source or destination might be inside a house which is not a road, thus not a node or not on top of an edge in the graph. Even if the source or destination position is on an edge, it might not be on a node, instead, it is in the middle of an edge. For this reason, the source location and the destination location are converted to nodes with location values (latitude, longitude), and added to the graph. The closest edge to the source or destination point is determined. Then a point closest to the source or destination point on that edge is detected. This closest point will be considered as a node and added to the graph. This node and the source or destination node will be connected with a new edge. Finally, the closest node on the closest edge to the source or destination will be connected to the two nodes of the edge. All the parameters of this edge will be calculated from the existing data. A demonstration of this process is shown on figure 6.

Finally, real-time changing data is considered in the network. The only real-time changing data is time in the network. Time may vary depending on the traffic of the path and the arrival rate of the transports. The passenger may need to wait on a single place or node for a bus or metro, rather than moving by walking or with rickshaw for a cheaper cost. The graph can further be optimized or added more parameter to getting more effective and realistic space-time modeling of traffic flow and arrival time of bus or metro in each node with help of Yiannis Kamarianakis and Poulicos Prastacos [30]. Algorithm 1 is used to get the first arrival time of transport after a certain

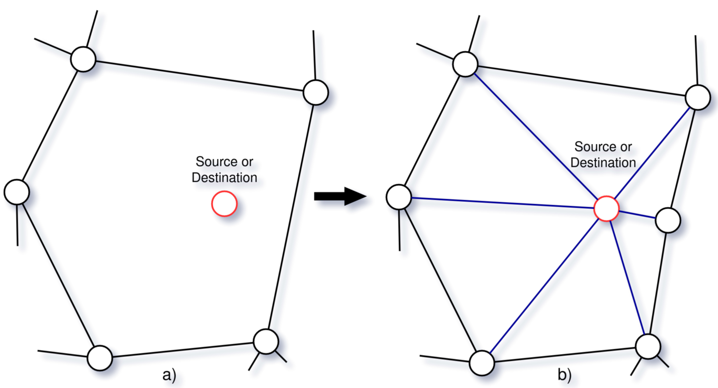

Figure 6: New node creation and graph augmentation if the source or destination location is not a node in the graph.

time at a specific place. The function assumes like Bangladesh that, buses or metro transports come from a starting node after N minutes interval, where N can be varied from 15 minutes to 45 minutes. By calculating the time taken due to traffic of the road with respective data, and speed of the transport, the function returns the next time a bus or train will arrive at a given node. The function is given on Algorithm 1. The function also considers a situation where one is at a starting point of a station, but the waiting time of the bus or metro transport is high. But a bus starting from another stoppage will come at the current node earlier. In that case, the earliest bus occurring time in the node is returned from the function.

Algorithm 1: Calculate next arrival time to a given node.

```
Data: Graph, Starting nodes of bus or metro, interval of bus leaving
      source node
Function nextarrival(node, current_time, busormetro):
    time[node] ← 0
    add node to priority queue q with highest priority value
    if node is starting node for busormetro then
        waiting_time ← calculate from current_time + t and interval
        t ← waiting_time
        add node to priority queue q with t as priority value
    end
    while q not empty do
        (p, t) ← pop most prior node from q and its priority
        if p is starting node of busormetro then
            return current_time + t
        end
        foreach incident edge e on p if type(e) = busormetro do
            n ← source node of edge e
            time[n] ← traffic time of e + t
            add n to q with t as priority value
            if n is starting node for busormetro then
                waiting_time ← calculate from (current_time + t and
                 interval)
                // Adding same node with waiting
                t ← t + waiting_time
                add n to q with t as priority value
            end
        end
    end
    return busormetro does not come at node
end
```

Two shortest path generating algorithms are used for the analysis. The first one is Uniform Cost Search (UCS) algorithm [31]. The second one is a modified version of the heuristic A* algorithm [32]. In the following subsections 6.1 and 6.2, the application of these two algorithms is explained.

6.1. Uniform Cost Search

Dijkstra algorithm works on a multi-edged directed graph. On this application, uniform cost searching, Dijkstra's modified algorithm has been applied. This similar algorithm was used to determine the shortest paths in Europe and the USA and it took less than one millisecond.

The algorithm used is presented on 2. The presented algorithm only applies to the cheapest pathfinding. From line 1 to 18, the initialization part is done. If there is no limit to walking distance in the traveling, the algorithm will always return cost 0 for the whole path and conclude to walk the whole path. That is why a walking distance limit is used. For each road, it is checked if the person has walked more than the limit on line 32. The algorithm will, by its nature, always check for the metro transport first, as it has the lowest cost and will be on top of the priority queue most of the time.

Algorithm 2: Cheapest Path Uniform Cost Search

Data:
Graph: Merged graph nodes and edges with all attributes
rate_b: Cost of bus per unit length
rate_m: Cost of metro per unit length
speed_w: Speed of walking
speed_r: Speed of rickshaw
speed_b: Speed of bus
speed_m: Speed of metro

Operators:
$A \sim_i B$: Length of ith edge from node A to B

Functions:
nextarrival(start, current[p], busormetro): Algorithm 1
rate_r(p, q, i): Rickshaw cost of ith edge from node p to q

Input:
start_place
max_w
current_time

Result: Cheapest path, Time at each place, Total Cost

foreach *node* $n \leftarrow Graph$ **do**
  $time[n][n] \leftarrow 0$
  $type[n][n] \leftarrow wait$
  $arrival[n] \leftarrow \infty$
  $parent[n] \leftarrow null$
  $cost[n] \leftarrow \infty$
  $current[n] \leftarrow \infty$
  $t\_walked[n] \leftarrow \infty$ // Total walked to reach this node
end
for *each node* $n1, n2$ *of edge in* $Graph$ **do**
  $time[n1][n2] \leftarrow \infty$
  $type[n1][n2] \leftarrow null$
end
$current[start\_place] \leftarrow current\_time$
$t\_walked[start\_place] \leftarrow 0$
if *start_place is incident node of edge type bus or metro* **then**
  $arrival[start\_place] \leftarrow$ `nextarrival`$(start\_place, current[start\_place], bus\ or\ metro)$
end
insert *start_place* into the priority queue with priority 1, q
if *start_place is out of graph* **then**
  throw error
end
// Else continue until the priority queue is empty

while *q not empty* **do**
  $p \leftarrow$ dequeue max prior element of q
  if *p is goal* **then**
    print path with transport type
    print total cost
    print time taken at each node
    exit
  end
  for *each incident c of all i edges from p* **do**
    if `type`$(p, c) = walking$ **and** $t\_walked[p] + (p \sim_i c) < min(max\_w, t\_walked[c])$ **and** $cost[p] < cost[c]$ **then**
      $time[p][c] \leftarrow (p \sim_i c) * speed\_w$
      $type[p][c] \leftarrow walk$
      $current[c] \leftarrow current[p] + time[p][c]$
      $t\_walked[c] \leftarrow t\_walked[p] + (p \sim_i c)$
      $parent[c] \leftarrow p$
      $cost[c] \leftarrow cost[p]$
      // walk later if it becomes cheaper
      add c to q with highest priority value
    end
    if `type`$(p, c) = rickshaw$ **and** $cost[p] + rate\_r(p, c, i) * (p \sim_i c) < cost[c]$ **then**
      $time[p][c] \leftarrow (p \sim_i c) * speed\_r * traffic\_param(p, c)$
      $type[p][c] \leftarrow rickshaw$
      $current[c] \leftarrow current[p] + time[p][c]$
      $parent[c] \leftarrow p$
      $cost[c] \leftarrow cost[p] + rate\_r(p, c, i) * (p \sim_i c)$
      add c to q with priority value $cost[c]$
    end
    if `type`$(p, c) = bus\ or\ metro$ **and** $cost[p] + rate\_b\ or\ rate\_b * (p \sim_i c) < cost[c]$ **then**
      $time[p][p] \leftarrow$ `nextarrival`$(start, current[p], busormetro)$
      $time[p][c] \leftarrow (p \sim_i c) * speed_b\ or\ speed_m * traffic\_param$
      $type[p][c] \leftarrow bus\ or\ metro$
      $current[c] \leftarrow current[p] + time[p][p] + time[p][c]$
      $parent[c] \leftarrow p$
      $cost[c] \leftarrow cost[p] + rate\_b\ or\ rate\_m * (p \sim_i c)$
      add c to q with priority value $cost[c]$
    end
  end
end

For the fastest route finding, algorithm 2 is slightly modified. Instead of pushing nodes with cost as priority on line 39, 47 and 56, the nodes will be pushed with time as priority. Then the nodes will be explored in order of how fast one can reach there. Besides some more slight modifications, the walking distance limit is not required for the fastest path searching. As walking is the slowest of all transportation, it will be explored at the last of all other edges.

6.2. A Algorithm*

A* algorithm is a heuristic search algorithm. It is one of the most popular algorithms for graph traversing and pathfinding when a heuristic can be used. For geographical pathfinding or road analysis straight line distance can be defined easily. This makes A* algorithm perfect for application in the current scenario.

Almost in all the cases, A* algorithm will be equivalent to uniform cost search algorithm. The time consumed by the A* pathfinding algorithm will always be less than uniform cost search algorithm. The reason for this is, uniform cost search will expand in all directions from the source. Even if the path leads to opposing direction from source to destination location, uniform cost search will take it into account. But in A* algorithm, the straight line distance is used as a heuristic, which helps to provide more priority to the paths that go toward to destination.

A major problem in A* algorithm is its optimality. A* algorithm might not provide the optimal path whereas uniform cost search or Dijkstra algorithm will always provide the shortest path in the graph. For maps or geographical data, best heuristics are always straight line distance, which is used in the experiments. All the results obtained during the experiments were always the shortest path in both cases of cost and time.

7. Experiments and Results

Sevaral dataset are used for experiment with the developed algorithms. On subsection 7.1, these dataset are discussed. Then the implementation details are discussed on subsection 7.2. Finally the result along with the time taken by the algorithms to achieve the results are discussed in subsection 7.3.

7.1. Dataset Preparation

The dataset used for all the experiments is taken from geodash.gov.bd [33]. On the website, there are private datasets for all the bus stands. There are also bus routes and road maps of the country, especially detailed maps of Dhaka city. The datasets can be downloaded as various file formats like shapefile [34], geojson [35], CSV, kml etc. As shapefiles are standard for geographical data storage and processing, it is selected for the experiments.

The metro routes and stoppages dataset is obtained from a hackathon named Code Samurai 2019 [36]. In the hackathon, the authority has made public a road map of Dhaka city, a list of bus stoppages and metro stations. The road map of Dhaka city is almost identical to the road map in OSM street map [37] and the maps provided in geodash. The dataset was in CSV file format. So the data files are read and converted to shapefiles to insert in the experiment pipelines.

7.2. Implementation Details

Python programming language was used to implement all the algorithms. Fiona python library was used to manage the epsg code standards [38] and all the in-between conversions. osmnx package [39] was used to collect some roads and geographic information. Geopandas [40], momepy [41] and shapely [42] was used to manipulate, query and preprocess the collected dataset. Finally, graph implementation and algorithms were applied with the popular networkx python package [43]. QGIS application [44] was used to visualize the dataset and cross-validate the results.

The dataset was collected and stored in shapefile format. Then the graph was built with the help of networkx. The same package was used to preprocess the dataset and shapely was used to calculate the straight line distance to destination in case of A* algorithm. Finally, pre and post data analysis was done with the combination of python in jupyter notebook and QGIS application. The whole process was done on a custom-built PC having Intel Core i7, 3.5 GHz, 32 GB RAM and NVIDIA GeForce GTX 1080 Ti GPU.

7.3. Result Analysis

In section 2, it is seen that GIS application in Dhaka city is very rare. The authors have collected most of the datasets of rickshaw and CNG costs by surveying. The complete routes demonstrated on figure 3 has 49869 nodes and 53878 edges. There are two datasets collected for the bus transportation system, Uttara bus service, and Bikolpo bus service. The number of nodes and edges of the Uttara bus service is 18 and 17. For the Bikolpo bus service, it is 20 and 19. Lastly, the metro rail dataset consists of 15 nodes and 14 edges. The time it took to merge the walking transportation graph and bus transportation graph was 1.83 seconds. After this, the time taken to merge it with the metro transportation graph was 1.81 seconds.

The road density analysis provides an insight that regions that have more roads tend to have lower rickshaw costs. This result will be useful to estimate the cost of a rickshaw at an unknown place as it is tedious to inquire and get all the costs of all the roads.

The new proposed algorithm for the shortest route analysis has worked well on some test source and destination places. Figure 7 shows as example of the cheapest path using algorithm 2. Figure 8 demonstrates the fastest path applied in the same scenarios where the time of arrival of buses and metro is calculated with algorithm 1 with 15 minutes interval.

8. Discussion and Conclusion

The methodology proposed in this research has tried to create a traveling solution that will better fit in Dhaka city. Despite being accurate and optimized, the proposed system has still some limitations.

The bus routine is assumed to be starting in all assigned places at the same time. This is assumed due to a lack of data from different places and can be easily altered to accept different starting times in different places.

The cost of rickshaw is assumed to be fixed once calculated for all places. But it can be dependent on the time of year. This is left for the future work of this research.

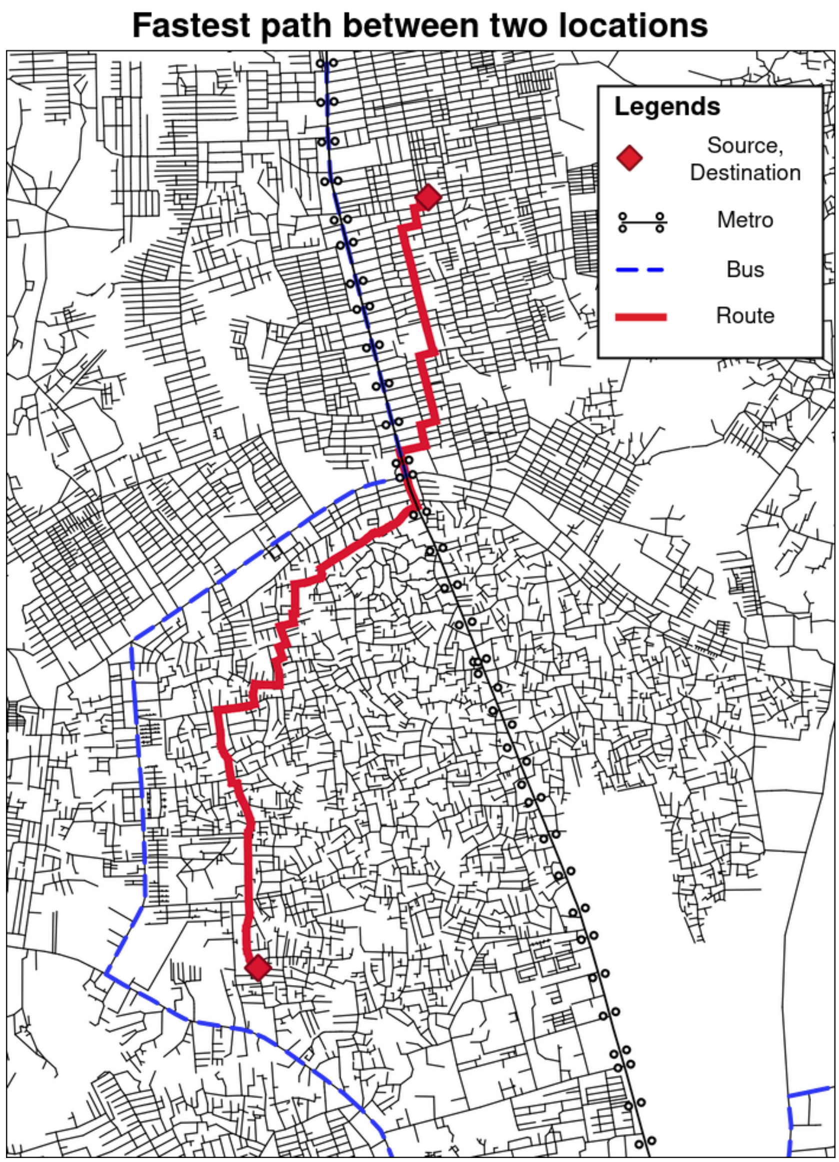

Figure 7: Fastest path between two locations (shown in red color)

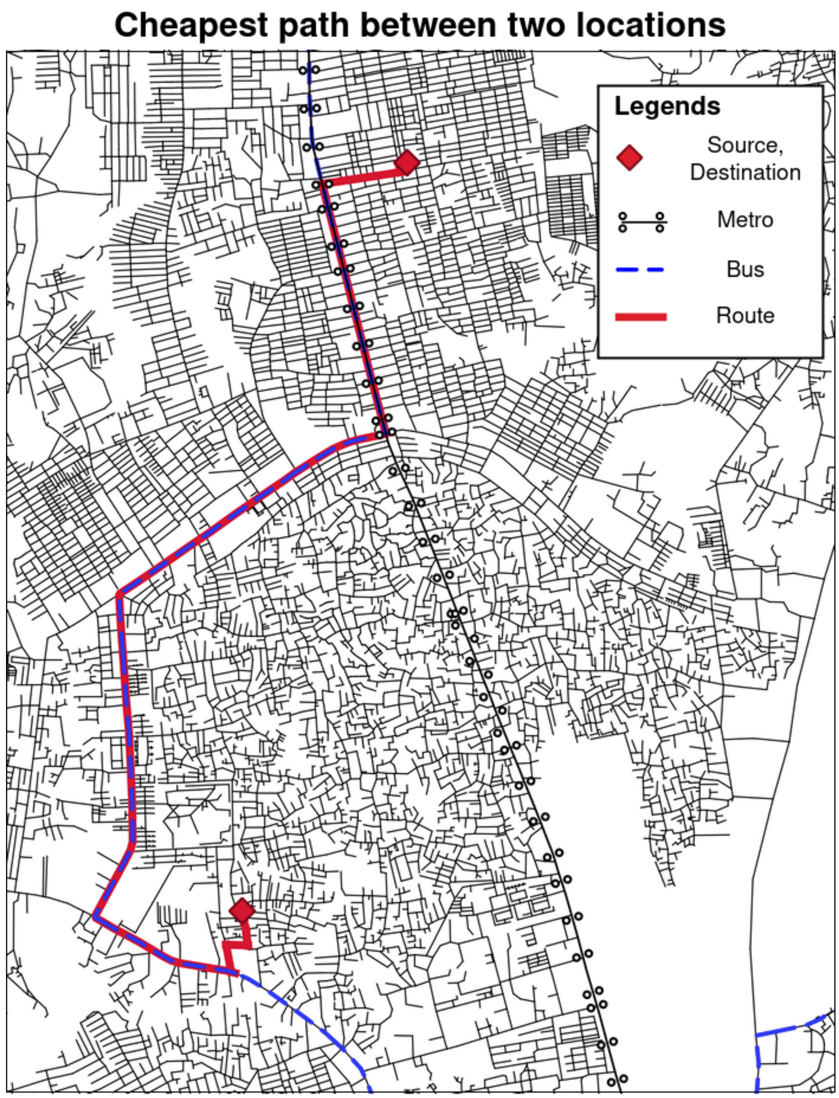

Figure 8: Cheapest path between two locations (shown in red color)

9. Acknowledgement

First of all, the authors express their sincere appreciation and heartfelt thanks to everyone who has inspired them for this paper's contributions. The authors would also like to thank all the people who helped them locate and collect the traffic datasets for the analysis. Finally, the authors want to thank this paper's reviewers and the researchers in the field of Geo-science for contributing to modern world knowledge.

10. Computer Code Availability

All the implementation code used in the research are open source and can be found on https://github.com/rahatzamancse/gis-analysis-of-dhaka-city-roads which is under user name of one of the authors Md. Rahat-uz-Zaman. The code can be run on any machine with proper python environment and qgis application as mentioned in the readme of the repository. Also, the dataset along with its proper explanation can be found in the published dataset Md. Rahat-uz Zaman [45] on Mendeley. Instructions to reproduce and analyze of data are also available in the public code repository.

References

[1] M. M. Hoque, The road to road safety: issues and initiatives in bangladesh, in: Regional Health Forum, volume 8, pp. 39–51.

[2] M. M. Hoque, B. Jobair, Strategies for safer and sustainable urban transport in bangladesh, Proceedings of CODATU X (2002).

[3] M. M. P. Rana, Urbanization and sustainability: challenges and strategies for sustainable urban development in bangladesh, Environment, Development and Sustainability 13 (2011) 237–256.

[4] R. Shamsher, M. N. Abdullah, Traffic congestion in bangladesh-causes and solutions: a study of chittagong metropolitan city, Asian Business Review 2 (2015) 13–18.

[5] A. Chowdhury, K. Kishi, K. Satoh, A study on highway logistics problem from dhaka to chittagong in bangladesh, in: Japan Society of Civil Engineers.

[6] E. Jenelius, T. Petersen, L.-G. Mattsson, Importance and exposure in road network vulnerability analysis, Transportation Research Part A: Policy and Practice 40 (2006) 537–560.

[7] W. Rees, Least-cost paths in mountainous terrain, Computers & Geosciences 30 (2004) 203–209.

[8] S. Ahmed, R. F. Ibrahim, H. A. Hefny, Gis-based network analysis for the roads network of the greater cairo area, in: Proc. of 2nd International Conference on Applied Research in Computer Science and Engineering.

[9] Y. Luping, Q. Dalin, Vulnerability analysis of road networks, Journal of Transportation Systems Engineering and Information Technology 12 (2012) 105–110.

[10] D. Das, A. K. Ojha, H. Kramsapi, P. P. Baruah, M. K. Dutta, Road network analysis of guwahati city using gis, SN Applied Sciences 1 (2019) 906.

[11] E. 2011, Arcgis desktop, ????

[12] A. Quium, S. Hoque, The completeness and vulnerability of road network in bangladesh, Engineering Concerns of Flood (2002) 59–75.

[13] Q. Li, Variable-scale representation of road networks on small mobile devices, Computers & geosciences 35 (2009) 2185–2190.

[14] A. A. Sufian, S. M. Khan, I. Ahmed, S. Islam, N. Saha, Safety analysis: observed deficiencies in existing transportation system of bangladesh, International Journal of Urban Planning and Transportation 27 (2016).

[15] R. Begum, S. N. Momotaz, Determinants of passengers'satisfaction with cng-run auto rickshaw services in bangladesh: An empirical study on dhaka city, DU Journal of Marketing 15 (2012) 159–174.

[16] H. M. Ahsan, et al., Study of mass transit in metropolitan dhaka (1990).

[17] B. Ahmed, The traditional four steps transportation modeling using a simplified transport network: A case study of dhaka city, bangladesh, International Journal of Advanced Scientific Engineering and Technological Research 1 (2012) 19–40.

[18] K. M. A. Waheed, Study on application of cut and cover method for the construction of metro rail tunnel in dhaka city (2008).

[19] M. S. Alam, Factors in deciding metro rail in developing countries: A study on the proposed metro rail system for dhaka, Journal of Bangladesh Institute of Planners ISSN 2075 (2010) 9363.

[20] M. M. Hoque, S. Barua, H. Ahsan, D. Alam, Brt in metro dhaka: towards achieving a sustainable urban public transport system, Proceedings of CODATU XV: The Role of Urban Mobility in (re) shaping Cities (2012) 1–11.

[21] M. Hossain, Y. O. Susilo, Rickshaw use and social impacts in dhaka, bangladesh, Transportation research record 2239 (2011) 74–83.

[22] S. Begum, B. Sen, Pulling rickshaws in the city of dhaka: a way out of poverty?, Environment and Urbanization 17 (2005) 11–25.

[23] S. I. Khan, A. Khan, M. N. I. Sarker, N. Huda, M. R. Zaman, A. Nurullah, M. Z. Rahman, Traffic congestion in dhaka city: Suffering for city dwellers and challenges for sustainable development, European Journal of Social Sciences 57 (2018) 116–127.

[24] K. R. Rahaman, N. Ohmori, N. Harata, Evaluation of the roadside walkway environment of dhaka city, in: Proceeding of the Eastern Asia Society for Transportation Studies, volume 5, pp. 1751–1766.

[25] M. S.-U. Rahman, K. Nahrin, Bus services in dhaka city-users' experiences and opinions, Journal of Bangladesh Institute of Planners ISSN 2075 (2012) 9363.

[26] J. W. Essam, M. E. Fisher, Some basic definitions in graph theory, Reviews of Modern Physics 42 (1970) 271.

[27] S. M. Khan, M. S. Hoque, Traffic flow interruptions in dhaka city: Is smooth traffic flow possible, Journal of Presidency University 2 (2013) 46–54.

[28] Q. Zhang, Road network generalization based on connection analysis, in: Developments in spatial data handling, Springer, 2005, pp. 343–353.

[29] Z. Zhang, S. Tan, W. Tang, A gis-based spatial analysis of housing price and road density in proximity to urban lakes in wuhan city, china, Chinese geographical science 25 (2015) 775–790.

[30] Y. Kamarianakis, P. Prastacos, Space–time modeling of traffic flow, Computers & Geosciences 31 (2005) 119–133.

[31] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1959) 269–271.

[32] P. Hart, N. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (1968) 100–107.

[33] B. Government, GeoDASH, 2016 (accessed November 5, 2019). Url: https://geodash.gov.bd.

[34] E. ESRI, Shapefile technical description, An ESRI white paper 39 (1998).

[35] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub, et al., The geojson format, Internet Engineering Task Force (IETF) (2016).

[36] B. University of Dhaka, Code Samurai 2019, 2019. Url: https://codesamuraibd.net/.

[37] C. driven map, OSM streep map, ???? Url: https://www.openstreetmap.org/.

[38] F. Warmerdam, Coordinate systems: Proj. 4, epsg and ogc wkt, Presentation at the FOSS4G2006-Free And Open Source Software for Geoinformatics http://2006. foss4g. org/getFile. py/accessf802. pdf (2006).

[39] G. Boeing, Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks, Computers, Environment and Urban Systems 65 (2017) 126–139.

[40] K. Jordahl, Geopandas: Python tools for geographic data, URL: https://github.com/geopandas/geopandas (2014).

[41] M. Fleischmann, momepy: Urban morphology measuring toolkit, Journal of Open Source Software 4 (2019) 1807.

[42] S. Gillies, The shapely user manual, 2013.

[43] A. Hagberg, D. Schult, P. Swart, Networkx: Python software for the analysis of networks, Mathematical Modeling and Analysis, Los Alamos National Laboratory (2005).

[44] Q. D. Team, et al., Qgis geographic information system. open source geospatial foundation project, URL: http://qgis. osgeo. org (2015).

[45] S. H. Md. Rahat-uz Zaman, Shortest route analysis of dhaka city roads using various gis techniques (dataset and sample outputs) (2020).

```
This is pdfTeX, Version 3.14159265-2.6-1.40.19 (TeX Live 2018/W32TeX)
(preloaded format=pdflatex 2018.7.12)  24 JUL 2020 08:35
entering extended mode
 restricted \write18 enabled.
 %&-line parsing enabled.
**elsarticle-template.tex
(./elsarticle-template.tex
LaTeX2e <2018-04-01> patch level 5
(c:/TeXLive/texmf-local/tex/latex/aries/elsarticle.cls
Document Class: elsarticle 2008/10/09, 1.0.2: Elsevier Science
\@bls=\dimen102
(c:/TeXLive/2018/texmf-dist/tex/latex/base/article.cls
Document Class: article 2014/09/29 v1.4h Standard LaTeX document class
(c:/TeXLive/2018/texmf-dist/tex/latex/base/size12.clo
File: size12.clo 2014/09/29 v1.4h Standard LaTeX file (size option)
)
\c@part=\count80
\c@section=\count81
\c@subsection=\count82
\c@subsubsection=\count83
\c@paragraph=\count84
\c@subparagraph=\count85
\c@figure=\count86
\c@table=\count87
\abovecaptionskip=\skip41
\belowcaptionskip=\skip42
\bibindent=\dimen103
) (c:/TeXLive/2018/texmf-dist/tex/latex/graphics/graphicx.sty
Package: graphicx 2017/06/01 v1.1a Enhanced LaTeX Graphics (DPC,SPQR)
(c:/TeXLive/2018/texmf-dist/tex/latex/graphics/keyval.sty
Package: keyval 2014/10/28 v1.15 key=value parser (DPC)
\KV@toks@=\toks14
) (c:/TeXLive/2018/texmf-dist/tex/latex/graphics/graphics.sty
Package: graphics 2017/06/25 v1.2c Standard LaTeX Graphics (DPC,SPQR)
(c:/TeXLive/2018/texmf-dist/tex/latex/graphics/trig.sty
Package: trig 2016/01/03 v1.10 sin cos tan (DPC)
) (c:/TeXLive/2018/texmf-dist/tex/latex/graphics-cfg/graphics.cfg
File: graphics.cfg 2016/06/04 v1.11 sample graphics configuration
)
Package graphics Info: Driver file: pdftex.def on input line 99.
(c:/TeXLive/2018/texmf-dist/tex/latex/graphics-def/pdftex.def
File: pdftex.def 2018/01/08 v1.0l Graphics/color driver for pdftex
))
\Gin@req@height=\dimen104
\Gin@req@width=\dimen105
) (c:/TeXLive/2018/texmf-dist/tex/latex/psnfss/pifont.sty
Package: pifont 2005/04/12 PSNFSS-v9.2a Pi font support (SPQR)
LaTeX Font Info:    Try loading font information for U+pzd on input line
63.
(c:/TeXLive/2018/texmf-dist/tex/latex/psnfss/upzd.fd
File: upzd.fd 2001/06/04 font definitions for U/pzd.
)
LaTeX Font Info:    Try loading font information for U+psy on input line
64.
```

```
(c:/TeXLive/2018/texmf-dist/tex/latex/psnfss/upsy.fd
File: upsy.fd 2001/06/04 font definitions for U/psy.
))
\c@tnote=\count88
\c@fnote=\count89
\c@cnote=\count90
\c@ead=\count91
\c@author=\count92
\@eadauthor=\toks15
\c@affn=\count93
\absbox=\box26
\keybox=\box27
\Columnwidth=\dimen106
\space@left=\dimen107
\els@boxa=\box28
\els@boxb=\box29
\leftMargin=\dimen108
\@enLab=\toks16
\@sep=\skip43
\@@sep=\skip44
(./elsarticle-template.spl) (c:/TeXLive/2018/texmf-
dist/tex/latex/natbib/natbib
.sty
Package: natbib 2010/09/13 8.31b (PWD, AO)
\bibhang=\skip45
\bibsep=\skip46
LaTeX Info: Redefining \cite on input line 694.
\c@NAT@ctr=\count94
)
\splwrite=\write3
\openout3 = `elsarticle-template.spl'.

) (c:/TeXLive/2018/texmf-dist/tex/latex/mathtools/mathtools.sty
Package: mathtools 2018/01/08 v1.21 mathematical typesetting tools
(c:/TeXLive/2018/texmf-dist/tex/latex/tools/calc.sty
Package: calc 2017/05/25 v4.3 Infix arithmetic (KKT,FJ)
\calc@Acount=\count95
\calc@Bcount=\count96
\calc@Adimen=\dimen109
\calc@Bdimen=\dimen110
\calc@Askip=\skip47
\calc@Bskip=\skip48
LaTeX Info: Redefining \setlength on input line 80.
LaTeX Info: Redefining \addtolength on input line 81.
\calc@Ccount=\count97
\calc@Cskip=\skip49
) (c:/TeXLive/2018/texmf-dist/tex/latex/mathtools/mhsetup.sty
Package: mhsetup 2017/03/31 v1.3 programming setup (MH)
) (c:/TeXLive/2018/texmf-dist/tex/latex/amsmath/amsmath.sty
Package: amsmath 2017/09/02 v2.17a AMS math features
\@mathmargin=\skip50
For additional information on amsmath, use the `?' option.
(c:/TeXLive/2018/texmf-dist/tex/latex/amsmath/amstext.sty
Package: amstext 2000/06/29 v2.01 AMS text
```

```
(c:/TeXLive/2018/texmf-dist/tex/latex/amsmath/amsgen.sty
File: amsgen.sty 1999/11/30 v2.0 generic functions
\@emptytoks=\toks17
\ex@=\dimen111
)) (c:/TeXLive/2018/texmf-dist/tex/latex/amsmath/amsbsy.sty
Package: amsbsy 1999/11/29 v1.2d Bold Symbols
\pmbraise@=\dimen112
) (c:/TeXLive/2018/texmf-dist/tex/latex/amsmath/amsopn.sty
Package: amsopn 2016/03/08 v2.02 operator names
)
\inf@bad=\count98
LaTeX Info: Redefining \frac on input line 213.
\uproot@=\count99
\leftroot@=\count100
LaTeX Info: Redefining \overline on input line 375.
\classnum@=\count101
\DOTSCASE@=\count102
LaTeX Info: Redefining \ldots on input line 472.
LaTeX Info: Redefining \dots on input line 475.
LaTeX Info: Redefining \cdots on input line 596.
\Mathstrutbox@=\box30
\strutbox@=\box31
\big@size=\dimen113
LaTeX Font Info:    Redeclaring font encoding OML on input line 712.
LaTeX Font Info:    Redeclaring font encoding OMS on input line 713.
\macc@depth=\count103
\c@MaxMatrixCols=\count104
\dotsspace@=\muskip10
\c@parentequation=\count105
\dspbrk@lvl=\count106
\tag@help=\toks18
\row@=\count107
\column@=\count108
\maxfields@=\count109
\andhelp@=\toks19
\eqnshift@=\dimen114
\alignsep@=\dimen115
\tagshift@=\dimen116
\tagwidth@=\dimen117
\totwidth@=\dimen118
\lineht@=\dimen119
\@envbody=\toks20
\multlinegap=\skip51
\multlinetaggap=\skip52
\mathdisplay@stack=\toks21
LaTeX Info: Redefining \[ on input line 2817.
LaTeX Info: Redefining \] on input line 2818.
)
LaTeX Info: Thecontrolsequence`\('isalreadyrobust on input line 129.
LaTeX Info: Thecontrolsequence`\)'isalreadyrobust on input line 129.
LaTeX Info: Thecontrolsequence`\['isalreadyrobust on input line 129.
LaTeX Info: Thecontrolsequence`\]'isalreadyrobust on input line 129.
\g_MT_multlinerow_int=\count110
\l_MT_multwidth_dim=\dimen120
```

```
\origjot=\skip53
\l_MT_shortvdotswithinadjustabove_dim=\dimen121
\l_MT_shortvdotswithinadjustbelow_dim=\dimen122
\l_MT_above_intertext_sep=\dimen123
\l_MT_below_intertext_sep=\dimen124
\l_MT_above_shortintertext_sep=\dimen125
\l_MT_below_shortintertext_sep=\dimen126
) (c:/TeXLive/2018/texmf-dist/tex/latex/lineno/lineno.sty
Package: lineno 2005/11/02 line numbers on paragraphs v4.41
\linenopenalty=\count111
\output=\toks22
\linenoprevgraf=\count112
\linenumbersep=\dimen127
\linenumberwidth=\dimen128
\c@linenumber=\count113
\c@pagewiselinenumber=\count114
\c@LN@truepage=\count115
\c@internallinenumber=\count116
\c@internallinenumbers=\count117
\quotelinenumbersep=\dimen129
\bframerule=\dimen130
\bframesep=\dimen131
\bframebox=\box32
LaTeX Info: Redefining \\ on input line 3056.
) (c:/TeXLive/2018/texmf-dist/tex/latex/tools/multicol.sty
Package: multicol 2018/04/20 v1.8s multicolumn formatting (FMi)
\c@tracingmulticols=\count118
\mult@box=\box33
\multicol@leftmargin=\dimen132
\c@unbalance=\count119
\c@collectmore=\count120
\doublecol@number=\count121
\multicoltolerance=\count122
\multicolpretolerance=\count123
\full@width=\dimen133
\page@free=\dimen134
\premulticols=\dimen135
\postmulticols=\dimen136
\multicolsep=\skip54
\multicolbaselineskip=\skip55
\partial@page=\box34
\last@line=\box35
\maxbalancingoverflow=\dimen137
\mult@rightbox=\box36
\mult@grightbox=\box37
\mult@gfirstbox=\box38
\mult@firstbox=\box39
\@tempa=\box40
\@tempa=\box41
\@tempa=\box42
\@tempa=\box43
\@tempa=\box44
\@tempa=\box45
\@tempa=\box46
```

```
\@tempa=\box47
\@tempa=\box48
\@tempa=\box49
\@tempa=\box50
\@tempa=\box51
\@tempa=\box52
\@tempa=\box53
\@tempa=\box54
\@tempa=\box55
\@tempa=\box56
\c@columnbadness=\count124
\c@finalcolumnbadness=\count125
\last@try=\dimen138
\multicolovershoot=\dimen139
\multicolundershoot=\dimen140
\mult@nat@firstbox=\box57
\colbreak@box=\box58
\mc@col@check@num=\count126
) (c:/TeXLive/2018/texmf-dist/tex/latex/booktabs/booktabs.sty
Package: booktabs 2016/04/27 v1.618033 publication quality tables
\heavyrulewidth=\dimen141
\lightrulewidth=\dimen142
\cmidrulewidth=\dimen143
\belowrulesep=\dimen144
\belowbottomsep=\dimen145
\aboverulesep=\dimen146
\abovetopsep=\dimen147
\cmidrulesep=\dimen148
\cmidrulekern=\dimen149
\defaultaddspace=\dimen150
\@cmidla=\count127
\@cmidlb=\count128
\@aboverulesep=\dimen151
\@belowrulesep=\dimen152
\@thisruleclass=\count129
\@lastruleclass=\count130
\@thisrulewidth=\dimen153
) (c:/TeXLive/2018/texmf-dist/tex/latex/graphics/color.sty
Package: color 2016/07/10 v1.1e Standard LaTeX Color (DPC)
(c:/TeXLive/2018/texmf-dist/tex/latex/graphics-cfg/color.cfg
File: color.cfg 2016/01/02 v1.6 sample color configuration
)
Package color Info: Driver file: pdftex.def on input line 147.
) (c:/TeXLive/2018/texmf-dist/tex/latex/algorithm2e/algorithm2e.sty
Package: algorithm2e 2017/07/18 v5.2 algorithms environments
\c@AlgoLine=\count131
(c:/TeXLive/2018/texmf-dist/tex/latex/base/ifthen.sty
Package: ifthen 2014/09/29 v1.1c Standard LaTeX ifthen package (DPC)
)
\algocf@hangindent=\skip56
(c:/TeXLive/2018/texmf-dist/tex/latex/ifoddpage/ifoddpage.sty
Package: ifoddpage 2016/04/23 v1.1 Conditionals for odd/even page
detection
\c@checkoddpage=\count132
```

```
) (c:/TeXLive/2018/texmf-dist/tex/latex/tools/xspace.sty
Package: xspace 2014/10/28 v1.13 Space after command names (DPC,MH)
) (c:/TeXLive/2018/texmf-dist/tex/latex/relsize/relsize.sty
Package: relsize 2013/03/29 ver 4.1
)
\skiptotal=\skip57
\skiplinenumber=\skip58
\skiprule=\skip59
\skiphlne=\skip60
\skiptext=\skip61
\skiplength=\skip62
\algomargin=\skip63
\skipalgocfslide=\skip64
\algowidth=\dimen154
\inoutsize=\dimen155
\inoutindent=\dimen156
\interspacetitleruled=\dimen157
\interspacealgoruled=\dimen158
\interspacetitleboxruled=\dimen159
\algocf@ruledwidth=\skip65
\algocf@inoutbox=\box59
\algocf@inputbox=\box60
\AlCapSkip=\skip66
\AlCapHSkip=\skip67
\algoskipindent=\skip68
\algocf@nlbox=\box61
\algocf@hangingbox=\box62
\algocf@untilbox=\box63
\algocf@skipuntil=\skip69
\algocf@capbox=\box64
\algocf@lcaptionbox=\skip70
\algoheightruledefault=\skip71
\algoheightrule=\skip72
\algotitleheightruledefault=\skip73
\algotitleheightrule=\skip74
\c@algocfline=\count133
\c@algocfproc=\count134
\c@algocf=\count135
\algocf@algoframe=\box65
\algocf@algobox=\box66
) (./elsarticle-template.aux)
\openout1 = `elsarticle-template.aux'.

LaTeX Font Info:    Checking defaults for OML/cmm/m/it on input line 255.
LaTeX Font Info:    ... okay on input line 255.
LaTeX Font Info:    Checking defaults for T1/cmr/m/n on input line 255.
LaTeX Font Info:    ... okay on input line 255.
LaTeX Font Info:    Checking defaults for OT1/cmr/m/n on input line 255.
LaTeX Font Info:    ... okay on input line 255.
LaTeX Font Info:    Checking defaults for OMS/cmsy/m/n on input line 255.
LaTeX Font Info:    ... okay on input line 255.
LaTeX Font Info:    Checking defaults for OMX/cmex/m/n on input line 255.
LaTeX Font Info:    ... okay on input line 255.
LaTeX Font Info:    Checking defaults for U/cmr/m/n on input line 255.
```

```
LaTeX Font Info:    ... okay on input line 255.
(c:/TeXLive/2018/texmf-dist/tex/context/base/mkii/supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
\scratchcounter=\count136
\scratchdimen=\dimen160
\scratchbox=\box67
\nofMPsegments=\count137
\nofMParguments=\count138
\everyMPshowfont=\toks23
\MPscratchCnt=\count139
\MPscratchDim=\dimen161
\MPnumerator=\count140
\makeMPintoPDFobject=\count141
\everyMPtoPDFconversion=\toks24
) (c:/TeXLive/2018/texmf-dist/tex/latex/oberdiek/epstopdf-base.sty
Package: epstopdf-base 2016/05/15 v2.6 Base part for package epstopdf
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/infwarerr.sty
Package: infwarerr 2016/05/16 v1.4 Providing info/warning/error messages
(HO)
) (c:/TeXLive/2018/texmf-dist/tex/latex/oberdiek/grfext.sty
Package: grfext 2016/05/16 v1.2 Manage graphics extensions (HO)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/kvdefinekeys.sty
Package: kvdefinekeys 2016/05/16 v1.4 Define keys (HO)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/ltxcmds.sty
Package: ltxcmds 2016/05/16 v1.23 LaTeX kernel commands for general use
(HO)
))) (c:/TeXLive/2018/texmf-dist/tex/latex/oberdiek/kvoptions.sty
Package: kvoptions 2016/05/16 v3.12 Key value format for package options
(HO)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/kvsetkeys.sty
Package: kvsetkeys 2016/05/16 v1.17 Key value parser (HO)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/etexcmds.sty
Package: etexcmds 2016/05/16 v1.6 Avoid name clashes with e-TeX commands
(HO)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/ifluatex.sty
Package: ifluatex 2016/05/16 v1.4 Provides the ifluatex switch (HO)
Package ifluatex Info: LuaTeX not detected.
)
Package etexcmds Info: Could not find \expanded.
(etexcmds)             That can mean that you are not using pdfTeX 1.50
or
(etexcmds)             that some package has redefined \expanded.
(etexcmds)             In the latter case, load this package earlier.
))) (c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/pdftexcmds.sty
Package: pdftexcmds 2018/01/30 v0.27 Utility functions of pdfTeX for
LuaTeX (HO
)
(c:/TeXLive/2018/texmf-dist/tex/generic/oberdiek/ifpdf.sty
Package: ifpdf 2017/03/15 v3.2 Provides the ifpdf switch
)
Package pdftexcmds Info: LuaTeX not detected.
Package pdftexcmds Info: \pdf@primitive is available.
Package pdftexcmds Info: \pdf@ifprimitive is available.
Package pdftexcmds Info: \pdfdraftmode found.
```

```
)
Package epstopdf-base Info: Redefining graphics rule for `.eps' on input
line 4
38.
Package grfext Info: Graphics extension search list:
(grfext)
[.pdf,.png,.jpg,.mps,.jpeg,.jbig2,.jb2,.PDF,.PNG,.JPG,.JPE
G,.JBIG2,.JB2,.eps]
(grfext)             \AppendGraphicsExtensions on input line 456.
(c:/TeXLive/2018/texmf-dist/tex/latex/latexconfig/epstopdf-sys.cfg
File: epstopdf-sys.cfg 2010/07/13 v1.3 Configuration of (r)epstopdf for
TeX Liv
e
))

! LaTeX Error: Environment highlights undefined.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.299 \begin{highlights}

Your command was ignored.
Type  I <command> <return>  to replace it with another command,
or  <return>  to continue without it.


! LaTeX Error: Lonely \item--perhaps a missing list environment.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.300 \item L
             ocal transport costs in road network require algorithms to
esti...

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Underfull \hbox (badness 10000) detected at line 300
[][]
 []


! LaTeX Error: Lonely \item--perhaps a missing list environment.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.301 \item S
```

```
             hortest path algorithms fail due to inconsistent fare of
transp...

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Underfull \hbox (badness 10000) detected at line 301
[][]
 []


! LaTeX Error: Lonely \item--perhaps a missing list environment.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.302 \item A
             * algorithm performs comparatively better for both cheap and
sh...

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Underfull \hbox (badness 10000) detected at line 302
[][]
 []


! LaTeX Error: Lonely \item--perhaps a missing list environment.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.303 \item M
             any transports can be merged with parameters to create finer
di...

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Underfull \hbox (badness 10000) detected at line 303
[][]
 []


! LaTeX Error: \begin{frontmatter} on input line 257 ended by
\end{highlights}.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.304 \end{highlights}

Your command was ignored.
Type  I <command> <return>  to replace it with another command,
or  <return>  to continue without it.

[1

{c:/TeXLive/2018/texmf-var/fonts/map/pdftex/updmap/pdftex.map}] [2

]

Package natbib Warning: Citation `hoque2004road' on page 3 undefined on
input l
ine 330.


Package natbib Warning: Citation `hoque2002strategies' on page 3
undefined on i
nput line 330.


Package natbib Warning: Citation `rana2011urbanization' on page 3
undefined on
input line 330.


Package natbib Warning: Citation `shamsher2015traffic' on page 3
undefined on i
nput line 330.


Package natbib Warning: Citation `chowdhury2002study' on page 3 undefined
on in
put line 330.


Package natbib Warning: Citation `jenelius2006importance' on page 3
undefined o
n input line 330.

[3]

Package natbib Warning: Citation `rees2004least' on page 4 undefined on
input l
ine 345.


Package natbib Warning: Citation `ahmed2017gis' on page 4 undefined on
input li
```

ne 347.

[4]

Package natbib Warning: Citation `luping2012vulnerability' on page 5
undefined
on input line 349.

Package natbib Warning: Citation `das2019road' on page 5 undefined on
input lin
e 351.

Package natbib Warning: Citation `arcgis' on page 5 undefined on input
line 351
.

Package natbib Warning: Citation `quium2002completeness' on page 5
undefined on
input line 353.

Package natbib Warning: Citation `li2009variable' on page 5 undefined on
input
line 355.

[5]

Package natbib Warning: Citation `sufian2016safety' on page 6 undefined
on inpu
t line 361.

Package natbib Warning: Citation `begum2012determinants' on page 6
undefined on
input line 361.

Package natbib Warning: Citation `ahsan1990study' on page 6 undefined on
input
line 365.

Package natbib Warning: Citation `ahmed2012traditional' on page 6
undefined on
input line 365.

[6]

Package natbib Warning: Citation `waheed2008study' on page 7 undefined on
input
line 368.

```
Package natbib Warning: Citation `alam2010factors' on page 7 undefined on
input
 line 368.


Package natbib Warning: Citation `hoque2012brt' on page 7 undefined on
input li
ne 370.


Package natbib Warning: Citation `hossain2011rickshaw' on page 7
undefined on i
nput line 372.


Package natbib Warning: Citation `begum2005pulling' on page 7 undefined
on inpu
t line 372.


Package natbib Warning: Citation `khan2018traffic' on page 7 undefined on
input
 line 372.


Package natbib Warning: Citation `rahaman2005evaluation' on page 7
undefined on
 input line 374.

[7]

Package natbib Warning: Citation `rahman2012bus' on page 8 undefined on
input l
ine 386.

[8]

LaTeX Warning: File `figs/interpolation.png' not found on input line 394.


! Package pdftex.def Error: File `figs/interpolation.png' not found:
using draf
t setting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.394 ...th=0.8\linewidth]{figs/interpolation.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.
```

```
[9]

Package natbib Warning: Citation `essam1970some' on page 10 undefined on
input
line 473.

[10] [11] [12]

Package natbib Warning: Citation `khan2013traffic' on page 13 undefined
on inpu
t line 490.


LaTeX Warning: File `figs/samplegraph.png' not found on input line 495.


! Package pdftex.def Error: File `figs/samplegraph.png' not found: using
draft
setting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.495 ...idth=0.7\linewidth]{figs/samplegraph.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


LaTeX Warning: File `figs/wholemap.jpg' not found on input line 538.


! Package pdftex.def Error: File `figs/wholemap.jpg' not found: using
draft set
ting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.538 ...s[width=0.8\linewidth]{figs/wholemap.jpg}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.

[13] [14] [15]

LaTeX Warning: File `figs/mergegraph.png' not found on input line 553.


! Package pdftex.def Error: File `figs/mergegraph.png' not found: using
draft s
```

```
etting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.553 ...width=0.5\linewidth]{figs/mergegraph.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Package natbib Warning: Citation `zhang2005road' on page 16 undefined on
input
line 560.


Package natbib Warning: Citation `zhang2015gis' on page 16 undefined on
input l
ine 560.

[16]

LaTeX Warning: File `figs/Density.png' not found on input line 564.


! Package pdftex.def Error: File `figs/Density.png' not found: using
draft sett
ing.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.564 ...cs[width=0.8\linewidth]{figs/Density.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.

[17] [18] [19]

LaTeX Warning: File `figs/augmentation.png' not found on input line 583.


! Package pdftex.def Error: File `figs/augmentation.png' not found: using
draft
 setting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.583 ...dth=0.7\linewidth]{figs/augmentation.png}
```

```
Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


Package natbib Warning: Citation `kamarianakis2005space' on page 20
undefined o
n input line 588.

[20] [21]
Overfull \vbox (82.89095pt too high) has occurred while \output is active
[]

[22]

Package natbib Warning: Citation `dijkstra1959note' on page 23 undefined
on inp
ut line 596.


Package natbib Warning: Citation `Hart1968' on page 23 undefined on input
line
596.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.
```

```
Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.

Overfull \hbox (28.11765pt too wide) in paragraph at lines 606--606
[]\OT1/cmr/m/n/6 (\OML/cmm/m/it/6 start[]place; current\OT1/cmr/m/n/6
[\OML/cmm
/m/it/6 start[]place\OT1/cmr/m/n/6 ]\OML/cmm/m/it/6 ; bus or
metro\OT1/cmr/m/n/
6 )$
 []


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.
```

```
Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
```

```
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.
```

```
Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.

Overfull \hbox (7.94026pt too wide) in paragraph at lines 606--606
[]\OT1/cmr/m/n/6 (\OML/cmm/m/it/6 start; current\OT1/cmr/m/n/6
[\OML/cmm/m/it/6
 p\OT1/cmr/m/n/6 ]\OML/cmm/m/it/6 ; busormetro\OT1/cmr/m/n/6 )$
 []


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.
```

```
Package relsize Warning: Font size 4.16667pt is too small.
(relsize)                Using 6.0pt instead on input line 606.


Overfull \hbox (17.62482pt too wide) detected at line 606
[][]
 []

[23]
Overfull \vbox (5.41638pt too high) has occurred while \output is active
[]

[24] [25]

Package natbib Warning: Citation `geodash' on page 26 undefined on input
line 6
25.


Package natbib Warning: Citation `esri1998shapefile' on page 26 undefined
on in
put line 625.


Package natbib Warning: Citation `butler2016geojson' on page 26 undefined
on in
put line 625.


Package natbib Warning: Citation `codesamurai' on page 26 undefined on
input li
ne 627.


Package natbib Warning: Citation `osmmap' on page 26 undefined on input
line 62
7.


Package natbib Warning: Citation `warmerdam2006coordinate' on page 26
undefined
 on input line 631.


Package natbib Warning: Citation `boeing2017osmnx' on page 26 undefined
on inpu
t line 631.


Package natbib Warning: Citation `jordahl2014geopandas' on page 26
undefined on
 input line 631.
```

```
Package natbib Warning: Citation `fleischmann2019momepy' on page 26
undefined o
n input line 631.


Package natbib Warning: Citation `gillies2013shapely' on page 26
undefined on i
nput line 631.


Package natbib Warning: Citation `hagberg2005networkx' on page 26
undefined on
input line 631.


Package natbib Warning: Citation `qgis2015qgis' on page 26 undefined on
input l
ine 631.

[26] [27]

LaTeX Warning: File `figs/fastest_path_final.png' not found on input line
645.


! Package pdftex.def Error: File `figs/fastest_path_final.png' not found:
using
 draft setting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.645 ...8\linewidth]{figs/fastest_path_final.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.


LaTeX Warning: File `figs/cheapest_path_final.png' not found on input
line 652.



! Package pdftex.def Error: File `figs/cheapest_path_final.png' not
found: usin
g draft setting.

See the pdftex.def package documentation for explanation.
Type  H <return>  for immediate help.
 ...
```

```
l.652 ...\linewidth]{figs/cheapest_path_final.png}

Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.

[28] [29] [30]

Package natbib Warning: Citation `rahat-gis-dhaka' on page 31 undefined
on inpu
t line 670.

No file elsarticle-template.bbl.

Package natbib Warning: There were undefined citations.

[31] (./elsarticle-template.aux) )
Here is how much of TeX's memory you used:
 4694 strings out of 492646
 67085 string characters out of 6133325
 191547 words of memory out of 5000000
 8484 multiletter control sequences out of 15000+600000
 14096 words of font info for 51 fonts, out of 8000000 for 9000
 1141 hyphenation exceptions out of 8191
 41i,23n,38p,1535b,704s stack positions out of
5000i,500n,10000p,200000b,80000s
<c:/TeXLive/2018/texmf-dist/fonts/type1/publi
c/amsfonts/cm/cmbx10.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfont
s/cm/cmbx12.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmbx
6.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmbx8.pfb><c:/
TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmex10.pfb><c:/TeXLive/2
018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmmi10.pfb><c:/TeXLive/2018/texmf
-dist/fonts/type1/public/amsfonts/cm/cmmi12.pfb><c:/TeXLive/2018/texmf-
dist/fon
ts/type1/public/amsfonts/cm/cmmi5.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/p
ublic/amsfonts/cm/cmmi6.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsf
onts/cm/cmmi8.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cm
r10.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr12.pfb><c
:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr17.pfb><c:/TeXLive/
2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr6.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr7.pfb><c:/TeXLive/2018/texmf-
dist/fonts/
type1/public/amsfonts/cm/cmr8.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/publi
```

```
c/amsfonts/cm/cmss8.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts
/cm/cmsy10.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmsy6
.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmsy7.pfb><c:/T
eXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cmsy8.pfb><c:/TeXLive/201
8/texmf-
dist/fonts/type1/public/amsfonts/cm/cmti10.pfb><c:/TeXLive/2018/texmf-d
ist/fonts/type1/public/amsfonts/cm/cmti12.pfb><c:/TeXLive/2018/texmf-
dist/fonts
/type1/public/amsfonts/cm/cmti7.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/pub
lic/amsfonts/cm/cmtt10.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfo
nts/cm/cmtt12.pfb><c:/TeXLive/2018/texmf-
dist/fonts/type1/public/amsfonts/cm/cm
tt8.pfb>
Output written on elsarticle-template.pdf (31 pages, 347435 bytes).
PDF statistics:
 213 PDF objects out of 1000 (max. 8388607)
 151 compressed objects within 2 object streams
 0 named destinations out of 1000 (max. 500000)
 1 words of extra memory for PDF output out of 10000 (max. 10000000)
```

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Manuscript Title: **Shortest Route Analysis of Dhaka City Roads Using Various GIS Techniques**

Authors:
Md. Rahat-uz-Zaman
Shadmaan Hye