

# Cairo Public Transport Route Finder

## A Pilot System

<https://doi.org/10.3991/ijes.v4i4.6542>

Khaled Ahmed Mohamed Hassan<sup>1</sup>, and Ghada Hassan<sup>1, 2</sup>

<sup>1</sup> British University in Egypt, Cairo, Egypt

<sup>2</sup> Ain Shams University, Cairo, Egypt

**Abstract**—Aiming to facilitate the choice of transport links leading from a starting location to a destination in greater Cairo, we propose in this work a public transportation mobile (android) application to assist users of public transport. The system is a pilot application that considers the public mini-buses network in three areas of Cairo, and builds the database of the mini-bus network verified on the ground. From this database, the transportation network graph consisting of nodes and possible links between them is constructed. Upon request, the system then identifies the series of public transport possible, calculates the shortest path between the two chosen locations, and displays the bus, or series of buses, and the routes to the user, ordered by distance. The specialized algorithm Dijkstra was implemented to find the shortest route.

**Index Terms**—Shortest path algorithm, Public transport networks, Android application.

### I. INTRODUCTION

Public transport commuters in Cairo suffer, in addition to congestion, their inability to control their time or money as they endeavor to find out the possible/ best route to reach their destination. No official routes are published either online or in print, and commuters have to rely on learning from other's experiences and advice. On the other hand, the internet use is growing massively, there is a huge number of smart phone users especially among the younger generation in Egypt, in addition to the reliable availability of 3G, and recently 4G, coverage.

A mobile application would be a suitable tool for discovering the best possible series of transports to arrive from one point to another. This work builds an Android based application for transportation links suggestion and optimization to assist users of public transport inside Cairo. The application user will choose the departure and destination locations from the drop down list, and the application will generate the suggested transport links to arrive to their destination based on optimal/shortest path.

#### A. Scope of work

The main objective of this project is to build an android mobile application to help users find out the most suitable public bus (or series of buses) to go from one location to another. The scope of work includes building up the database of locations in three regions of Cairo which are Nasr City, Masr El Gdeeda and Al Abassia, as well as the network of minibuses routes in these areas. Accordingly, we will build the transportation network constituting of nodes and possible links between them. Subsequently, we will investigate and implement algorithmic solutions, currently we consider one optimization objective: the shortest path

between the two nodes selected. The public transport considered will be the network of minibuses connecting these three areas. For the rest of this paper we use the word bus & minibus interchangeably.

To achieve this goal, these sub goals are to be achieved:

- Building database of the mini-bus transport network in the regions of Nasr City, Masr El-Gdeeda and Al-Abassia. The database consists of every bus passing in these areas (bus number), the route it takes, and the nodes it passes on the route.
- Implementation of shortest path algorithms. With possibly more than one available path to consider from the public transport network considered, we need to be able to choose the best among them. The algorithms that could be applied here are many, for example Dijkstra (Shortest Path Algorithm) [1] and A\* algorithm [2].
- Development of a user friendly mobile application (Android based), that allows the choice of departure and destination points, discovers the shortest path between them, and visually presents the results found.

#### B. Limitations

- The application works for nodes within three regions only: Nasr City, Masr El-Gdeeda and Al-Abassia.
- The application's database does not include all possible points (nodes) in each region. Since in many cases there are no officially identified bus stops, but rather community agreed on stops, we have only considered main points (nodes) on every route.
- In case of multiple routes between origin and destination, the routes are compared based on shortest distance only. Other possible objectives could be added, like number of transfers [3], road restrictions, or avoiding congestion for which more advanced algorithms are required [4].

### II. BACKGROUND AND RELATED WORK

Public transportation, also known as public transit or mass transit, is a service offered by the government or private sector for movement of people or goods in mass from and to various points around cities. Public transportation network in Cairo, Egypt is vast and varied. It includes more than one type of government buses, minibuses, trains, trams, and the underground, in addition to the private minibuses and microbuses. Transportation has a big impact on economic growth in general. On the individual level, having digital means that facilitate finding which transport(s) to ride to go from point A to point B saves time, and effort and positively impacts travelers and new comers to any city.

Several web and mobile applications exist to solve the same problem all over the world. Many of these applications are established, mature, and cover the whole transport network of the city. Examples of mobile applications include: applications which are worldwide are Transit (Google App), and Moovit (<http://moovitapp.com>). Even more established are the web application for specific city route finders/ trip planning, an example of which is TFL in London, and similar applications in various cities around Europe. Efforts to develop these types of applications are continuous and are always evolving. Mobile application that provide tracking information of buses as well as trip planning are now also available [5], Transit is also an example of such applications. In Egypt, these applications are still in their infancy, and are still facing many problems due to lack of official data of the routes of public transport, the fragmentation of the data among many official entities and the lack of published schedules.

Algorithms to be used for these applications are many, and fall under the graph shortest path algorithms. Examples are: Dijkstra Algorithm Floyd Warshal Algorithm, heuristic shortest path algorithms, label-setting algorithm, Johnson's Algorithm, Brouvka's Algorithm, Prim's Algorithm, label correcting algorithm and label searching algorithm [3, 4, 6, 7, 8]. Dijkstra Algorithm was chosen at this stage for its simplicity [1].

Dijkstra works with weighted graphs and is a greedy algorithm. It repeatedly divides up a problem into smaller

ones then begins to solve one by one until all of the small problems are solved, and then the big problem is solved. In case of negative weights, the suitable algorithm will be Bellman Forman algorithm [8], but Bellman Forman algorithm has lower efficiency than the Dijkstra algorithm, and also, since the weights convey distance between two points, no negative weights are possible. Dijkstra's original algorithm does not use a min-priority queue and runs in time  $O(|V|^2)$  where  $V$  is the number of nodes. A more efficient implementation based on a min-priority queue implemented by a Fibonacci heap and running in  $O(|E| + |V| \log |V|)$ , where  $|E|$  is the number of edges. This is asymptotically the fastest known single-source shortest-path algorithm for arbitrary directed graphs with unbounded non-negative weights [8].

### III. CAIRO MINIBUSES NETWORK MODELLING

This work considers three specific areas of Cairo, which are Nasr City, Masr El-Gdeeda and Al Abassia and the public transport considered is the network of minibuses in these areas. The maps and roads of these areas were described geographically as illustrated below, and the nodes (popular stops) were decided upon. Next, data of the network of minibuses passing in these areas were gathered by collecting field data. Afterwards, graphs describing the nodes, existing paths (a mini bus passing through these two points), and the distance (weight) were created as illustrated in Figure 1 and Figure 2.

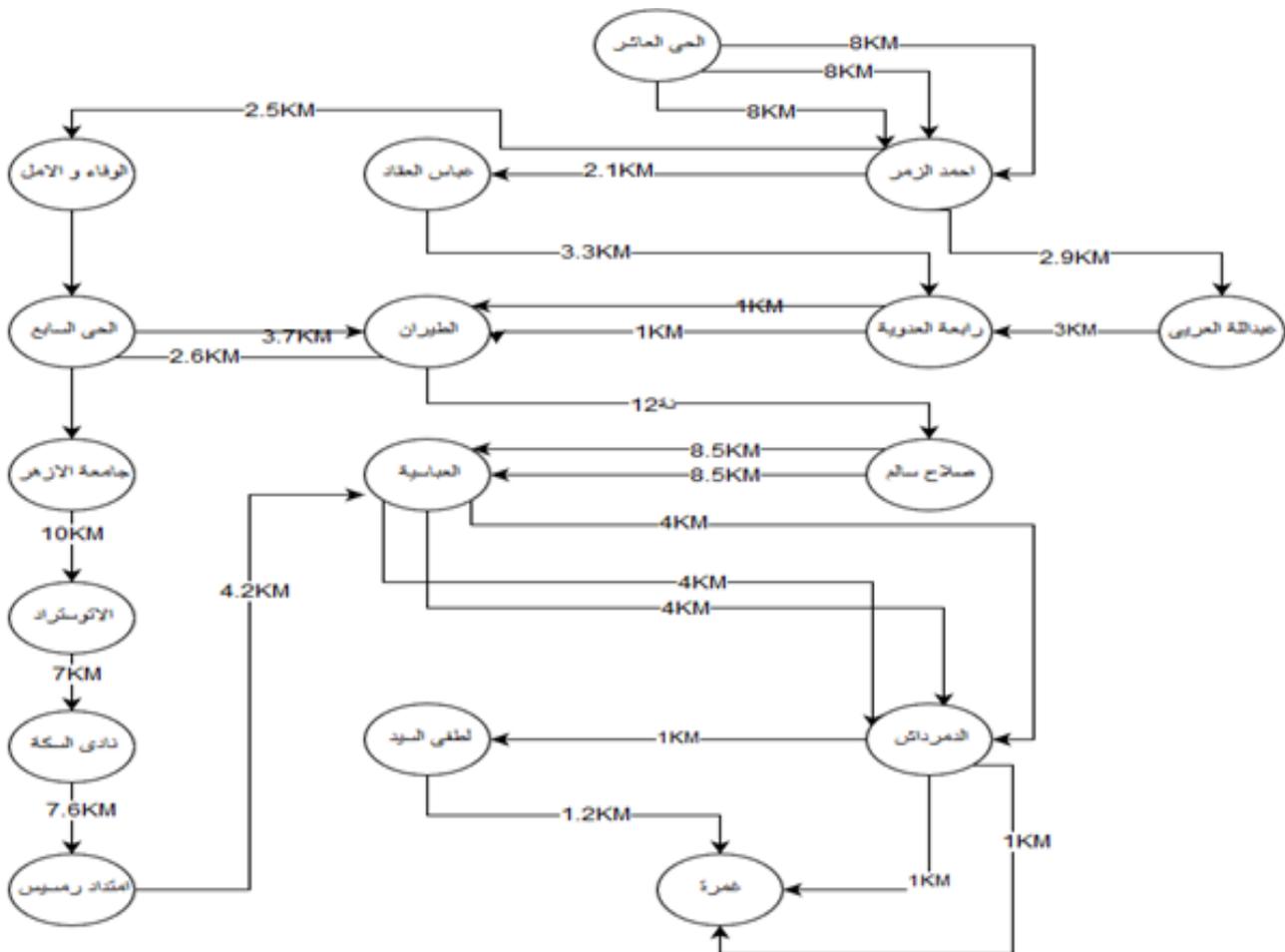


Figure 1. Sample graph I



### B. DataBase Tables

The following data base tables were created and populated with the data gathered:

#### 1) Stations Table:

This table is responsible for all the station of the buses and it has two attributes, first one is the “ID” this is the primary key of stations table, the second one is the “Name” which consists of all the points that each bus go through it and “Location ID” is for describing the belonging region for each station, if it is in Nasr City it will be 1, if it is in Masr El Gdeeda it will be 2 and if it is in Al Abassia it will be 3.

#### 2) Busses Table:

This table is responsible for the “Start Station” and “End Station” for each bus. It consists of 4 attributes which are, “ID”, “Number” which has the number of each bus, “Start ID” which has the start station of the bus and “End ID” which has the end station of the bus.

#### 3) Distances Table:

This table is responsible for the distances between the stations for each bus. It consists of four attributes which are, “ID”, “From ID” which is the station, “To ID” which is also another station and “Distance” which is the distance between these two stations. From ID and Go DI means the ID of the stations.

#### 4) Routes Table:

This table is a table formed due to the many to many relationship between Busses Table and Distances Table. This table consists of three attributes which are “ID”, Busses ID” which has the ID of number of the bus and “Distance ID” which has the distance between each two stations.

#### 5) Every Station Table:

This table consists of three attribute which are ID, Station ID, Bus ID, the main responsibility for this table is to consider and figure out the number of buses that goes to each station.

## IV. APPLICATION DESIGN

The application is implemented on the Android platform. Android have more than 2 million applications in the market which proves that is a suitable platform to build the application on it in order to spread out (<http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>). Android is a mobile operating system based on Linux and currently owned by Google and it was designed for smart phones with touch screens. Android is popular with companies that want a cheap operating system for suitable for a variety of technology devices. Its open nature encourage developers to use this open source, however android is less secure than the iPhone operating system. Overall, Android has succeeded to be one of the most famous mobile operating system due it flexibility and open source nature [9], [10].

### A. Functional Requirements

The application main page consists of two tabs, first tab for searching for the shortest buses route between two points, and the other for searching by a specific bus number. If the second tab is selected, the user enters a bus

number and the application outputs the route of that bus. If the first tab is selected, two input fields for “From” and “To” location open, each of these two locations has a three radio buttons for Nasr City, Masr El Gdeeda and Al Abassia. The user should selects one of these radio buttons, after that a drop down list will appear to the user has the main nodes (stops) in the selected region. The user choose the “From” and “To” locations and presses search. The Dijkstra algorithm begins to work to find the shortest path between the two points then outputs the path found on another page. The shortest path page shows the shortest path route between the two points, and contains the “Show Map” button which allows the user to see the path on Google Maps, “Show Buses” button which allows the user know which buses go through these two points. When the user clicks on show buses the application opens another page called “Available Buses” that has the available buses that go through these two points.

### B. Application Interface

The design of the application is straight forward. There is a splash screen in the start-up of the application for several seconds, then the main page opens and consists of the two tabs for finding out the route of a specific bus and for searching for the shortest route between two points. In case of the second option, the user is prompted to select origin and destination areas, and then accordingly select a node in the region from a drop down list. Figures 3 to 7 show different screen shots illustrating the user interface.



Figure 3. Interface screenshot: Get bus route given a bus number



Figure 4. Select origin and destination points

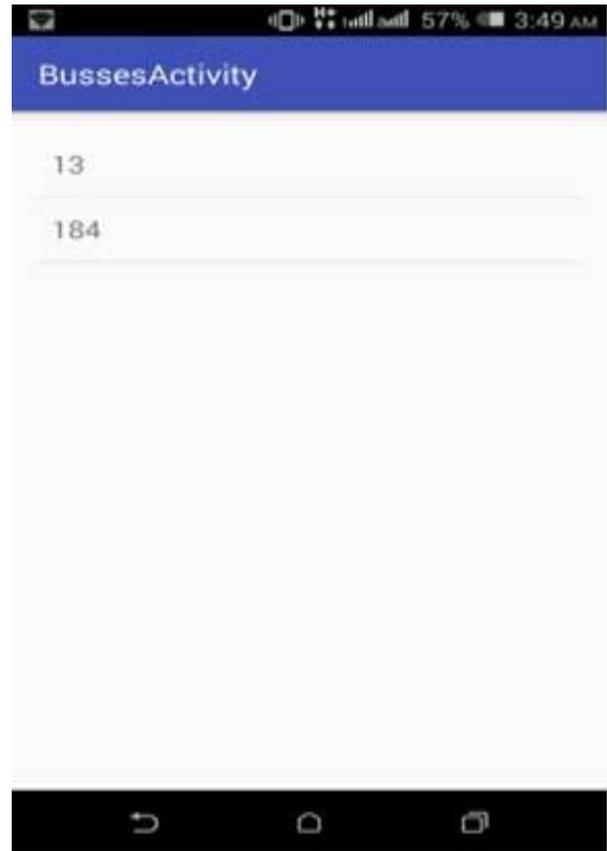


Figure 6. Result of clicking available buses



Figure 5. Shortest path resulting from the selection in Figure 4. Now the user has the choice of either finishing which bus to ride or to show a map of the route



Figure 7. Map displayed for the shortest route found (nodes displayed)

V. TESTING AND EVALUATION

Test cases for this application included testing:

- Shortest path results,
- Output of available buses,
- Points of the stations on the map, and
- Routes of the buses

Tables 1-4 demonstrate the design of the test cases.

VI. CONCLUSION AND FUTURE WORK

Cairo’s massive transport network has no published schedules and limited data is made available. Also there is no digitally enabled tool for finding out the possible transport links between two points. Recently, there has been some efforts to gather the data and implement digital tools to help commuters plan their journeys across the capital. In this project we aimed to build a pilot sample system that facilitates route identification for the mini-buses network in three specific areas of Cairo.

We devolved a mobile (android) application where the user chooses the departure and destination locations from the drop down list, and the application generates the suggested transport links to arrive at the required destination based on optimal/shortest path. We collected the data of the mini buses network operating in these three areas, and built the underlying database of viable nodes in the considered areas. From the data collected, the graphs of possible paths between the nodes were constructed. Finally, we investigated a number of algorithmic solutions for finding the optimal route, namely: Dijkstra Algorithm, A star Algorithm, and label searching algorithm, currently only the Dijkstra algorithms is implemented on the application. We presented the design and interface of the application, and the database built from on the ground information. This pilot system presents the ground work from which there is a huge area for improvement. For this pilot system, more nodes could be added to the system, and other shortest path algorithms for comparison of performance when the routes get more complicated. To fully automate the public transport route selection within Cairo, data collection from other public transport available needs to be collected, and the same effort of building up the databases and graphs is required to make the application more effective for Cairo public transport users.

REFERENCES

- [1] Dijkstra, E. W.. "A Note on Two Problems in Connexion with Graphs". Numerische Mathematik Vol. 1: 269–271, 1959. <https://doi.org/10.1007/BF01386390>
- [2] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, Feb. 1968. <https://doi.org/10.1109/TSSC.1968.300136>
- [3] Daniel Delling, Thomas Pajor and Renato F. Werneck. "Round-Based Public Transit Routing". Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12). Society for Industrial and Applied Mathematics, 2012. <https://doi.org/10.1137/1.9781611972924.13>
- [4] Robert Geisberger. Advanced Route Planning in Transportation Networks. PhD Thesis, Universität Karlsruhe, Denmark, 2011.
- [5] Shuju Wu, Xiaobing Hou and Karen Coale Tracey. "Use of Mobile Application Development Technologies in Capstone Projects". Proceedings of the IAJC-ISAM International Conference ISBN 978-1-60643-379-9, 2014.

TABLE I.      SHORTEST PATH RESULTS

Test Case Name	Shortest Path Results
<i>Objective</i>	Ensure that the output path is the real shortest path.
<i>Test case Scenario</i>	Open the application and choose two points then click search.
<i>Input</i>	Choosing two points "From" and "To".
<i>Expected Output</i>	The shortest path between these two points.
<i>Actual Output</i>	The output path is matching with the expected output which is really the shortest path.
<i>Pass Fail</i>	Pass.
<i>Numbers of actual test cases</i> 4	

TABLE II.      OUTPUT OF AVAILABLE BUSES

Test Case Name	Output of Available Buses
<i>Objective</i>	Ensure that the output of buses is really right buses and they path through the two points entered by the user.
<i>Test case Scenario</i>	Open the application and choose two points then click search and find available buses.
<i>Input</i>	Choosing two points "From" and "To".
<i>Expected Output</i>	The available buses between these two points.
<i>Actual Output</i>	The output buses are really path with these points and matches with the expected output.
<i>Pass Fail</i>	Pass.
<i>Numbers of actual test cases</i> 4	

TABLE III.      POINTS OF THE STATIONS ON THE MAP

Test Case Name	Points of The Stations on The Map
<i>Objective</i>	Ensure that the points of stations are appear clearly on the map and appear in the right order.
<i>Test case Scenario</i>	Open the application and choose two points then click on show map.
<i>Input</i>	Choosing two points "From" and "To".
<i>Expected Output</i>	The points on the map are appear clearly and they are the same points as the output and in the right order.
<i>Actual Output</i>	The points appear clearly on the map and there is not any confusion.
<i>Pass Fail</i>	Pass
<i>Numbers of actual test cases</i> 4	

TABLE IV.      BUS ROUTE

Test Case Name	Route of The Bus
<i>Objective</i>	Ensure that the output of the route is the real and right output of the bus number.
<i>Test case Scenario</i>	Open the application and go to search tap, then enter the bus number you want to search for.
<i>Input</i>	Entering the number of bus.
<i>Expected Output</i>	The route of the bus is the right route.
<i>Actual Output</i>	The output route of the bus is the right route.
<i>Pass Fail</i>	Pass
<i>Numbers of actual test cases</i>	

PAPER  
CAIRO PUBLIC TRANSPORT ROUTE FINDER – A PILOT SYSTEM

- [6] L. Fu, D. Sun and L.R. Rilett. Heuristic shortest path algorithms for transportation applications: State of the Art. Computers & Operations Research, Volume 33, Issue 11, PP 3324–3343, November 2006.
- [7] James Munkres, Algorithms for Transportation Problems, journal of society for industrial and applied mathematics, Vol. 5, No.1, 2014.
- [8] Vaibhavi Patel, and ChitraBagga. “A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm for Finding Shortest Path in GIS Application”. International Journal of P2P Network Trends and Technology (IJPTT) – Volume 5 – February 2014 ISSN: 2249-2615.
- [9] Android API Guide: <http://developer.android.com/guide/index.html>. May 19, 2015.
- [10] Patrick Lebegue, Takatoshi SUENAGA, Android Development, 2013.

AUTHORS

**Ghada Hassan** is with the British University in Egypt, and Ain Shams University, Cairo, Egypt. (e-mail: [ghada.hassan@bue.edu.eg](mailto:ghada.hassan@bue.edu.eg)).

**Khaled Hassan**, was with the British University in Cairo and is with Ain Shams University, Cairo, Egypt.

Submitted 04 December 2016. Published as resubmitted by the authors 25 December 2016.