The Shortest Path to the Health Services in Baghdad Using the Improved Dijkstra Algorithm on mobile device

Dena Kadhim Muhsen, Rihab Flaih Hassan
Department of Computer Sciences, University of Technology, Baghdad, Iraq

dena.kadhum.2015@gmail.com

Abstract

Shortest path problem is very significant in the study of transportation and communication network. Travelling is important part of daily life. Therefore the optimal shortest path (SP) to the particular destination becomes inevitable to reduce costs, losses in productivity, pollutions, and risks etc. An appropriate algorithm is used to find the shortest path between two known vertices is Dijkstra's algorithm which is submitted by the famous Dutch computer scientist Edsger W. Dijkstra, which was recognized as the optimal algorithm that can be applied to obtain the shortest path from a vertex (node) to any other vertex. In this paper design an android application based on improved Dijkstra algorithm which used constraint function and develop this algorithm by using priority queue and determine the source by GPS .This Application assist to find the shortest path to hospital from user location, time, distance, all hospital information related (like hospital name, contact number, e-mail, and specialization). It is helpful in terms of reducing transition time, effort, and cost. Android mobile's application considered as a guide for users toward the hospitals in Karada region from many locations within Baghdad city.

Keywords: Improved Dijkstra Algorithm, GPS, Mobile Application, Health services.

الطريق الاقصر للخدمات الصحيه في بغداد باستخدام خوارزميه دايكسترا المحسنه على جهاز الموبايل

دينا كاظم محسن , رحاب فليح حسن

قسم علوم الحاسوب , الجامعه التكنولوجيه , بغداد , العراق

الخلاصه

مشكلة المسار الاقصر جدا مهمه في دراسة التنقل وشبكة الاتصالات . التنقل هو جزء مهم في حياتنا اليوميه . لذلك السبب, اصبح المسار الاقصر الامثل لهدف معين لا مفر منه لتقليل الكلف , الخسائر بالانتاج , التلوث,والمخاطر ...الخ. الخوارزميه الملائمه لايجاد المسار الاقصر بين قمم او عقد معروفه هي خوارزميه الدايكسترا و التي قدمت من قبل عالم الحاسوب الهولندي المشهور اديسكير والذي عرفها كافضل خوارزميه التي من الممكن تطبيقها لايجاد المسار الاقصرمن عقده الى اي عقده اخرى . في هذا البحث تم تصميم تطبيق اندرويد يعتمد على خوارزميه دايكسترا المحسنه التي تستخدم الداله الشرطيه وتطوير هذه الخوارزميه باستخدام الطابور المعتمد على الاولويه وتحديد المصدر بواسطه GPS . التطبيق يساعد في ايجاد الطريق الاقصر الى المستشفى من موقع

المستخدم ،الوقت ,المسافة ، وجميع المعلومات المتعلقة بالمستشفى (مثل اسم المستشفى، ورقم الهاتف، البريد الألكتروني، والتخصصات)، مفيد من حيث التقليل من وقت التنقل، الجهد، والتكلفة. يعتبر هذا التطبيق دليلا للمستخدمين للتوجه نحو المستشفيات في منطقة الكراده من مواقع عديده داخل مدينه بغداد.

الكلمات المفتاحيه : خوارزميه دايكسترا المحسنه , نظام الموقع العام , تطبيق موبايل , الخدمات الصحيه .

1. Introduction

Nowadays the transportation is important issue to travel from one place to another in efficient and easy way, reducing in time, cost, and effort through obtain a system that can give decisions in one of necessary domains in our daily life named health services domain [1]. Finding the shortest path to hospitals is implemented through this system by using the famous algorithm in graph theory known Dijkstra's algorithm to provide health service to all people who need this service and guide them to correct choice for route. Everyone wants easy access for their desired locations including all the requested details of the location [2]. Many people, whether they are patients or general people may benefit from this system in addition to pharmaceutical companies that need information about the specialties of a particular hospital for the purpose of providing drugs and medical supplies. hospitals various in their types, all these hospitals provide services for general people, most of these hospitals are free supporting by government and some other are not. Many regions and streets around Karada region within Baghdad are taken (about 75 regions and street) to test any location within these regions and find the shortest path from this location to all hospitals in Karada.

2. ndroid Platform and Mobile Applications

Now a day, with the large expansion in the Internet, many information tools and systems have been modified dramatically. With the growth of mobile technology, smart mobile equipment such as smart phones become not only people’s daily used tools, but also the lifestyle. A smart mobile equipment is promising being able to do almost everything that a personal computer can do from a common user perspective. Furthermore, mobile phones are offered with flexibility, portability and instantaneity. Android is an open source mobile operation system developed by Google. Because of rapid development of smartphones, there are available variety of free and commercials applications of shortest path and spatial decision support systems where they are using various solutions. It has been become easy and efficient to build mobile application in the current study to accomplish task and provide simple technique for user to be used. The origin of android name is Greek word, the first part andr means "man" and the second part oid means "of the species", in another words, it means "being human". The definition of android is comprehensive, open and free platform which is designed mainly for touch screen mobile phones, like smart phones and tablet computers. Android comprises operating system that is depending on the free linux kernel, the essential middle - ware, libraries, and key mobile applications [3]. Developers for Android are able to build their applications by using the software development kit (SDK) developed by Google. SDK contains APIs (Android Application Program Interfaces)) for android to provide a great services set of system in class files [4]. Android Mobile Application Development is depending on Java language codes. It permits developers to write codes in the Java language. The relational database SQLite is embedded into android to share information. The Android Development Tools (ADT) encompasses an emulator to run an Android system. Mostly the emulator is like the real android device which permits to test the application [5].

3. G eneral System Architecture

The proposed system was built to find the nearest hospital from user. Improved Dijkstra algorithm has been used as a part of proposed system to provide the shortest path. The architecture of the proposed system in Figure (1), the Spatialite – gui (a GUI tool for SQLite or Spatialite) and android studio are used to contribute in implementation With a specific programming language like java language and Xml language (Extensible Markup Language) are used.

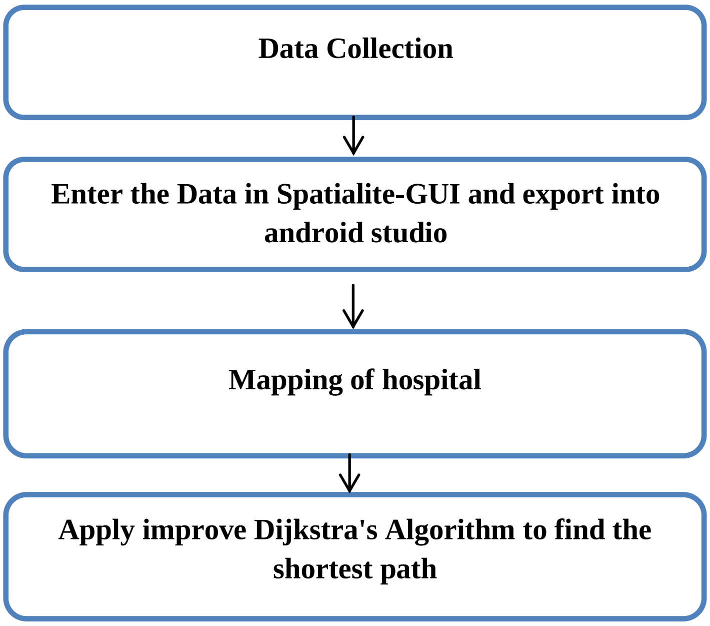

Data Collection
Enter the Data in Spatialite-GUI and export into android studio
Mapping of hospital
Apply improve Dijkstra's Algorithm to find the shortest path

Figure 1: The Block Diagram of Proposed System

3.1 Data Collection

The data required for this mobile application is in the form of spatial data like the locations for hospitals, road nodes, edges of road network and the This spatial data are obtained through the process of digitizing base map of specific area and provide the locations coordinates for the hospitals (longitude and latitude) by GPS (Table 1) and attribute data It collected by surveying of each hospital which contain name, departments, contact number, type of the health care provided and email id. All these information should be available for analysis. All the hospitals locate in Karada region.

Table 1: Karada hospitals and health services

ID	Names of hospitals	Longitude X	Latitude Y
1	Jenin Private Hospital	44.406140	33.293115
2	Al Khadria private hospital	44.408057	33.289154
3	Majeed private Hospital	44.412561	33.294193
4	AL-Jadiriah private Hospital	44.422773	33.293669
5	Albishara private Hospital	44.419922	33.301658
6	Al Hayat Al Rahibat private Hospital	44.438857	33.306123
7	Al Alwaiya Children Teaching Hospital	44.451155	33.306968
8	Hay Babil clinic	44.459057	33.301019
9	Hospital St. Raphael	44.422376	33.307685
10	Dar Al Najat Private Hospital	44.429691	33.309638
11	Istishari Eyes private Hospital	44.431768	33.310776
12	International medical center	44.431897	33.311680
13	Ibn Al-Bitar Hospital for heart surgery	44.425919	33.310750
14	Al Alwaiya Private Hospital	44.426725	33.310866
15	Al Amal Private Hospital	44.429944	33.312066
16	Ibn Al Haitham Teaching Eye Hospital	44.429333	33.313023
17	Baghdad Private Hospital	44.426641	33.313406
18	Al-Jarah Private Hospital	44.426098	33.314250
19	Al Alwaiya center for dentistry	44.422201	33.313837
20	Shaikh Zayed Hospital	44.428439	33.316067
21	Al-Amal National Hospital for tumor treatment	44.429914	33.316695
22	Al Wasity Hospital for orthopedic surgery	44.432055	33.318399
23	Al-Elwiya Educationl Hospital	44.433393	33.319477
24	Ibn AL Nafees Hospital	44.427956	33.318041
25	Al Rafidain Private Hospital	44.427748	33.318943
26	al taj Private hospital	44.426925	33.321367
27	Ibn Rushd Psychiatry Hospital	44.425534	33.318143
28	AL- Ferdows Private hospital for ophthalmology	44.422991	33.316381
29	Al-Saadoon Private hospital	44.421368	33.317785
30	AL- Ferdows Private hospital	44.419148	33.318000
31	Kamal Al Samerra'y hospital	44.425634	33.317259

3.2: Enter the Data in Spatialite-GUI and export into android studio

The Spatialite–GUI has been used to store all information about Karada hospitals, nodes and edges of roads network. It is an open source Graphical User Interface (GUI) tool supporting SpatiaLite or SQLite.

3.3 **Mapping of hospitals**

The coordinates of the thirty one hospitals and health centers were taken by using Global Positioning System (GPS), then these hospitals mapping on map as a point (Figure2).

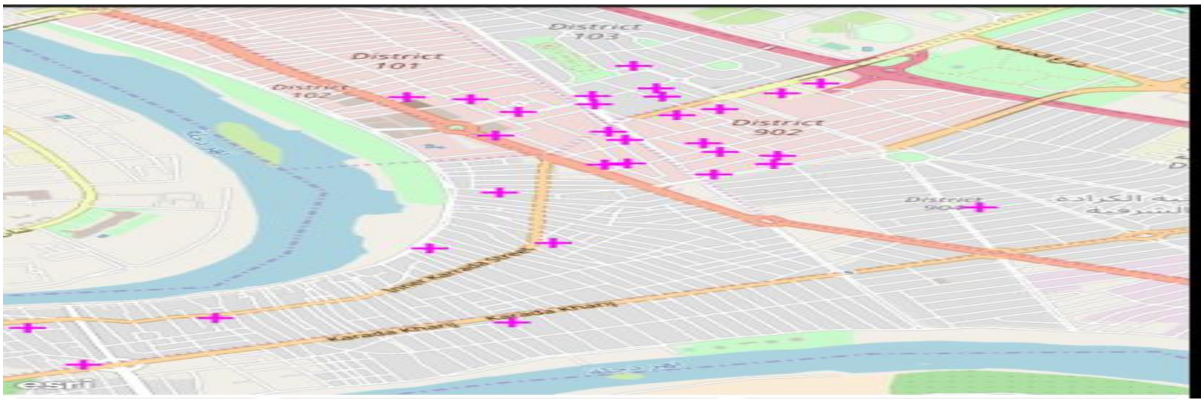


Figure 2: Mapping Hospitals on Study Area

3.4 Apply Improve Dijkstra's Algorithm to Find the Shortest Path

This stage is very important because it’s the core of the proposed system. Implementation of the Dijkstra algorithm done after finishing the preparation of the all roads edges (streets), roads nodes (intersections), the hospital nodes, and map of study area , then deciding the shortest path depend on this roads network. This search algorithm was implemented on the map after getting the start location of GPS and selects the desired hospital. The task of the algorithm is to find the shortest path to this hospital. Algorithm (1) illustrates the improved Dijkstra algorithm. The general sequence of processing the implementation of this algorithm explained in Figure (3).

Algorithm 1: Improved Dijkstra

```
Input: weighted Graph (G),          //  contains all node and edges for roads
       Source,                      //  start node in graph to begin the algorithm
       Multiple destination         //  the hospitals
Output: shortest path from [source]  to all multiple destinations in G
```

Process:

```
Step 1: Begin
        If GPS coordinates = node then        // node is one of road nodes
          Source = node
          dist [source]:=0,    else           // Distance from source to source
          nodes ∈{BBOx (location, r)}         // Bounding Box is an area defined by min longitude- min latitudes -max longitude - max latitude, r is the radius from current location
          for each node in nodes
            Distance [node, location]
          End for
          Mindistance [node] = minimum {Distance [node, location]}
          Source= mindistance [node]
        end
Step 2: for each vertex v in Graph            // Initializations
Step 3: dist [v]:=infinity                    // Unknown distance function from  source to v
Step 4: previous[v]:=undefined                // Previous node in optimal path  from source
Step 5: end for
Step 6: add source to Queue
Step 7: if source = target
Step 8: return;
Step 9: while Queue is not empty              // the main loop
Step 10: u: =vertex in Q with min dist[u]     // source node in first case
Step 11: remove u from Q
Step 12: for each adjacencies of u:
Step 13: calculate angle
Step 14: If $(-\frac{\pi}{2} \le \text{angle} \le \frac{\pi}{2})$
Step 15: continue
Step 16: improved=impact*Cos (angle (edge))
Step 17: distanceThroughU=dist [u] + edge.weight + improved
Step 18: if distanceThroughU<dist[v]          // A shorter path to v has been found
Step 19: dist[v] =distanceThroughU
Step 20: previous[v] = u
Step 21: end if
Step 22: end for
Step 23: end while
24   return dist [], previous []
25   end
```

Flowchart in **Figure (3)** explains how the algorithm works. The details definition of the elements used as explained below:

GPS: Current location (longitude (x), latitude (y)).

Nodes and Edges data base: The SQLite or Spatialite data base for nodes and edges of roads.

Intersection nodes ∈ **Bounding Box**: Contain the neighbors nodes from current location in bounding box when the current node not one of roads nodes.

Queue list: Using a priority queue instead of list because the queue is more efficient and it taking less memory storage.

Min {Di}: The nearest node from current location.

d: The shortest path weight value from start location to current node. The weights of roads edges available in meter when downloaded from OSM, it must convert to kilometer and then to decimal (decimal is equal 110 km on the earth) when insert it in equation of improved Dijkstra algorithm as follow:-

Weight = (weight/1000) / 110

The result is a weight in decimal degree, which can be handled in equation where also the impact factor in decimal degree.

r: Constraint function.

w: Impact factor, the value used is 0.05 decimal degree . The 1 decimal degree is equal about 110000m on the earth, so:

0.05*110000=5500m (impact factor in meter is 5500)

$\boldsymbol{Cos\theta}$**:** Is the angle between the vector that consist of nodes from starting point to current node and the vector that consist of nodes from starting point to the end point, the spatial reference of these points is (3857) for ESRI company maps[6].

The angle has been get between two points a and b as in equation (3.1) [7].

Angle = atan2 (b. y, b. x) – atan2 (a. y, a. x) …………………………………...……………... (1)

And this angle range is: **- pi/2<= angle<=pi/2**

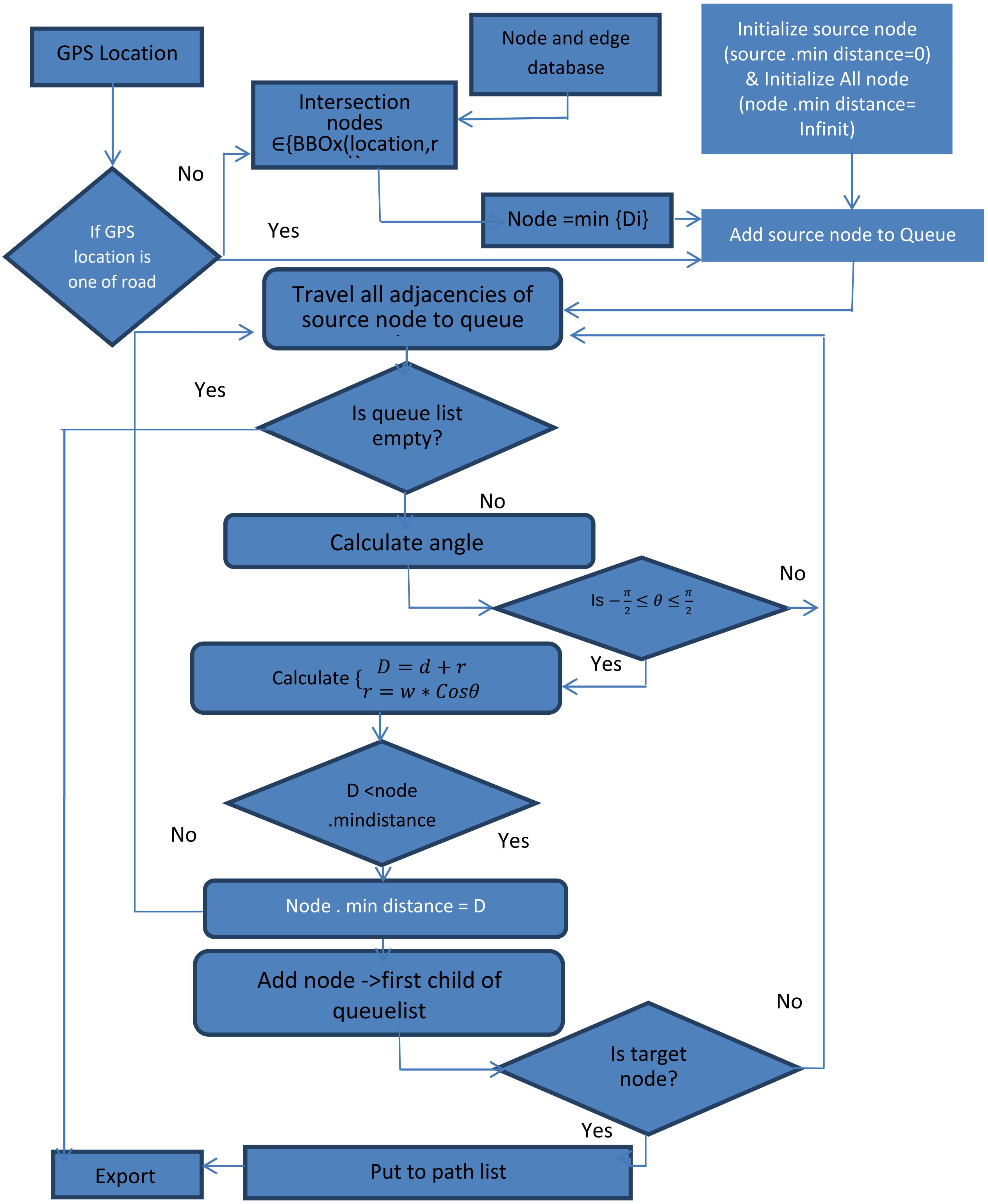

GPS Location
Node and edge database
Initialize source node (source .min distance=0) & Initialize All node (node .min distance= Infinit)
Intersection nodes ∈{BBOx(location,r
No
If GPS location is one of road
Yes
Node =min {Di}
Add source node to Queue
Travel all adjacencies of source node to queue
Yes
Is queue list empty?
No
Calculate angle
No
Is $-\frac{\pi}{2} \le \theta \le \frac{\pi}{2}$
Yes
Calculate $\begin{cases} D = d + r \\ r = w * Cos\theta \end{cases}$
D <node .mindistance
No
Yes
Node . min distance = D
Add node ->first child of queuelist
No
Is target node?
Yes
Export
Put to path list

Figure 3: The Flowchart for Improved Dijkstra Algorithm

4. Proposed System Implementation

A virtual device called AVD (Android Virtual Device) is provided by Android studio to each application test. The device is independent in its running and run only one application at a time. This AVD is useless unless install the operating file for SDK, there is some factors that limit the virtual device such as speed. Generally the virtual device is much slower than a real device, it can use the Genymotion as a relatively fast Android emulator which suitable for application testing. Genymotion is definitely faster than the AVD manager, also contains the GPS where specific coordinates can be entered for testing the application.

The user is any person who wants to use this mobile application to benefit from the health services submitted by this application. The system of Android application begin from getting the current location of the user (longitude and latitude) from mobile GPS or can determine the specific coordinates of GPS in Genymotion emulator in windows system of computer. When the user chooses from menu as in Figure (4) and determine the options. The system starts to implement all the requirements. The user can select a particular hospital by Clicking the menu and select the desired hospital from the suggest hospitals (Figure 5).

The shortest path appear on map to hospital and all the details of this hospital also appear above it. User can zoom in and zoom out the map on screen.

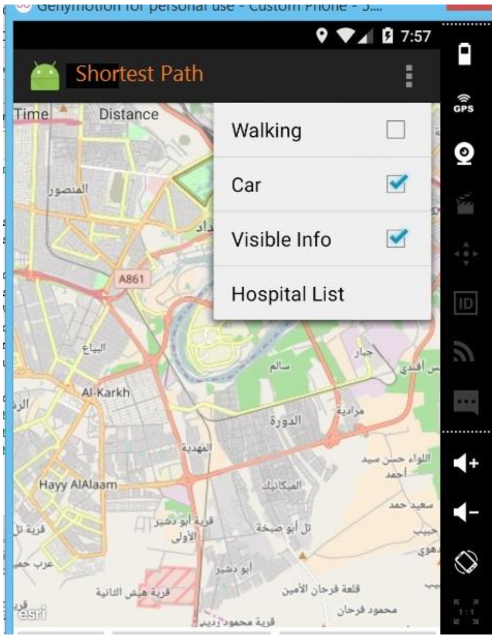


Figure 4: Interface Screen of the Shortest Path Application

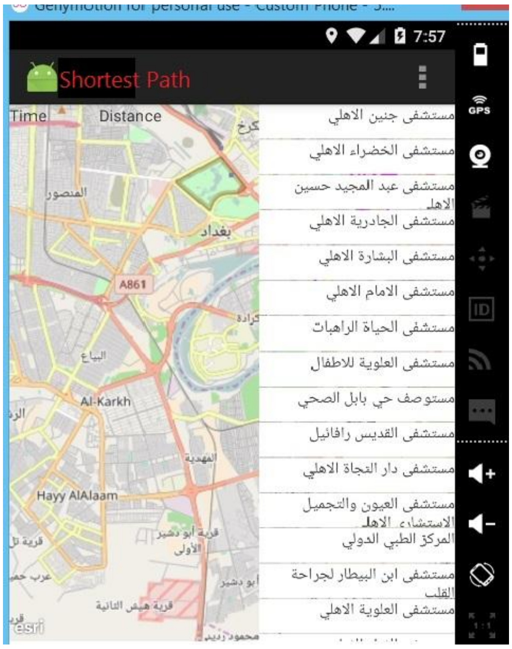
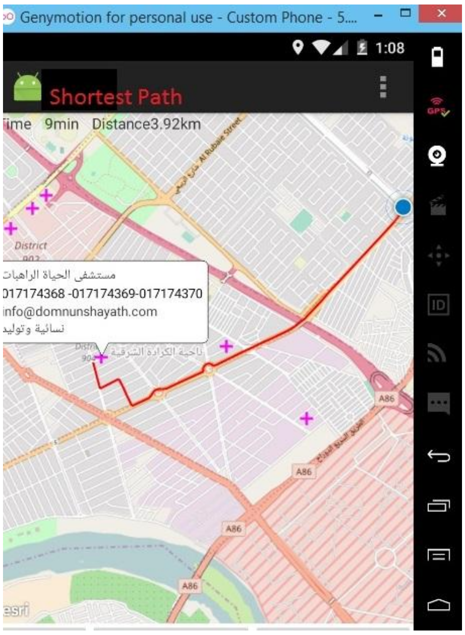


Figure 5: The Hospitals Menu in Application

5. Experimental Results

The user should be selected options to activate the application. The shortest path application gives the user the shortest path to hospital with time and distance (Figure 6), in addition to the details about the hospital.



Genymotion for personal use - Custom Phone - 5....

Shortest Path

Time 9min Distance3.92km

مستشفى الحياة الراهبات
017174368 -017174369-017174370
info@domnunshayath.com
نسائية وتوليد

Figure 6: The Shortest Path to Hospital by Mobile Application

The current location (taken by GPS) is either one of the road nodes or close to the road nodes, in the lastly state the mobile application takes node that closest to the current location to start executing the algorithm, the representation for the two cases in the following points :-

1. The current location node not one of road nodes, as a result take the closest node to the current location and considered it the start of the algorithm to find the shortest path (Figure 7), the current location is indicated by blue circle (●) and the start of the red line is the shortest path begin at one of road node which closest to the current location.

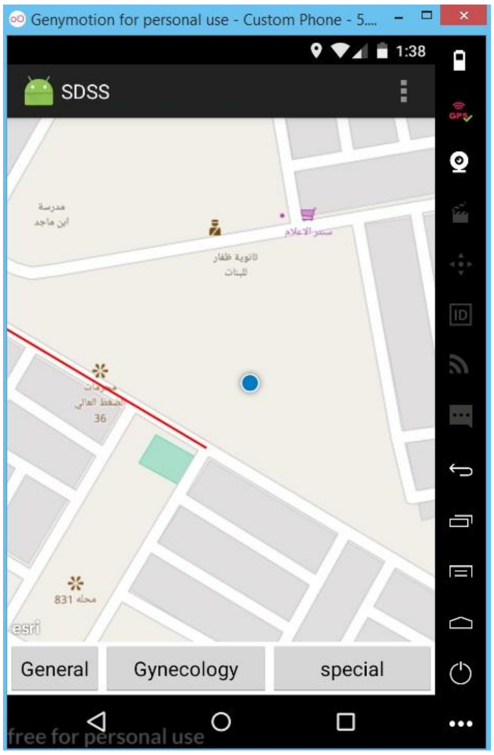
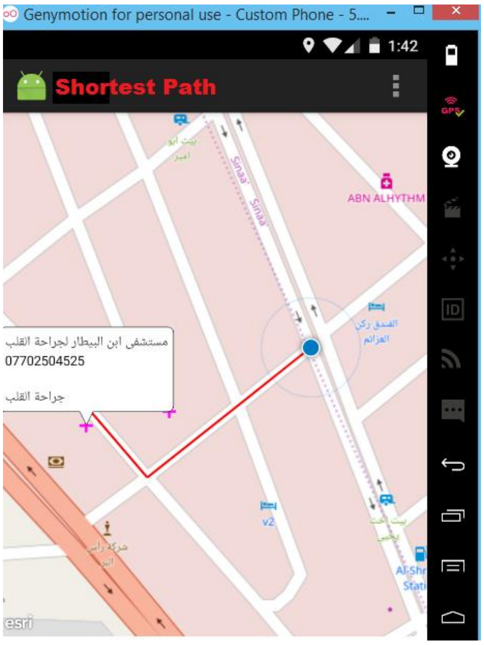


Figure 7: The Current Node not One of the Road Nodes

2. The current location node is one of road nodes, as a result take this node considered it the start of the algorithm to find the shortest path (Figure 8), the current location is indicated by blue circle (●) and the start of the red line is the shortest path begin at this road node.



مستشفى ابن البيطار لجراحة القلب
07702504525

جراحة القلب

Figure 12: The Current Node is One of the Road Nodes

Also to compare between improved Dijkstra algorithm in proposed system and traditional Dijkstra algorithm can get the results shown in Table (2)

Table 2: Comparison between Improved and Traditional Algorithm in both Search Nodes and Search Time

Start point	Target	Conventional algorithm		Improved algorithm		Weighted Value (Impact factor)
		N	**T**	**N**	**T**	**W**
Latitude 33.286671 Longitude 44.354340	1	18731	741	4239	238	800
		18731	741	4165	212	5500
	2	18731	995	4230	217	800
		18731	995	4148	143	5500
	3	18731	895	4240	174	800
		18731	895	4169	152	5500
Latitude 33.292424 Longitude 44.391555	1	18725	768	3463	129	800
		18725	768	3369	113	5500
	2	18725	967	3036	141	800
		18725	967	2890	152	5500
	3	18725	577	3473	143	800
		18725	577	3380	127	5500
Latitude 33.295412 Longitude 44.425212	1	18707	673	14057	610	800
		18707	673	13769	468	5500
	2	18707	1015	14010	450	800
		18707	1015	13846	732	5500
	3	18707	777	14057	703	800
		18707	777	13789	877	5500

N is referring to nodes, T is the time for search in milliseconds and W is a weighted value. From the results of the table, we can note that by taking a specific point and applying the improved Dijkstra algorithm based on the search strategy to find the distance between this point and the first three hospitals from the menu in the mobile application for sequence (1, 2, 3). The number of the search nodes and the search time decreases when using this algorithm compared with the conventional algorithm because in improved algorithm it ignores reverse nodes that not led to the goal. The number of nodes is reduced about 77% and the time about 73% than conventional algorithm. Besides it is possible to observe from experimental results the weighted value (impact factor) is flexibly changed where it can see that when using impact factor (5500), the number of nodes and time consuming are reduced than when using (800). The selection of impact factor is very important. When d(n) in equation of constraint function is large, it should set the impact factor to large value in order to the constraint function r(n) not much less than d(n) because if verified lastly case or the r(n) equals to 0, there is no constraint function and the algorithm is considered as conventional not improved. The constraint function in this improved algorithm directs the search towards the destination node in the complex network.

6. Conclusion

The large growth of technology in mobile applications assist people in daily life in various domains and especial in this study provide the health services to everyone need it. This application considered excellent guide for user or patient to benefit from its services by decrease time, cost, and effort by finding the shortest path to the hospitals from user location, most appropriate roads, time, distance, and all information related to this facilities. The improved Dijkstra algorithm its better than traditional algorithm because it reducing nodes number and consuming time.

References

[1] Amrapali Dabhade, K.V. Kale and Yogesh Geda, Network Analysis for Finding Shortest Path In Hospital Information System, International Journal of Advanced Research In Computer Science and Software Engineering, 5(7):618-623, 2015.

[2] Suneet Naithani, Abhishek Choudhry and Sandeep Chauhan, Decision Support System for Emergency Response, Scholarly

Research Journal for Interdisciplinary Studies, II/IX: 680-687, 2013.

[3] Semertzidis, K., Mobile Application Development to Enhance Higher Education Lectures, M.Sc. thesis, University of York, 2013.

[4] Poonam Bedarkar and Hemlata Meharkure, Literature Review on Android Application Developed on Eclipse Software, International Journal on Recent and Innovation Trends in Computing and Communication, 3(2):75-79, 2015.

[5] Bedarkar, P. and Meharkure, H., Literature Review on Android Application Developed on Eclipse Software, International Journal on Recent and Innovation Trends in Computing and Communication, 3(2):75-79, 2015.

[6] Srinath, D., Assorted GIS Utilites, M.Sc. thesis, Faculty of San Diego State University,2014.

[7]. Dragos Niculescu and Badri Nath, VOR Base Stations for Indoor 802.11 Positioning, MobiCom 4 Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, USA, 26 September - 1 October, 2004.