USING A* ALGORITHM FOR PUBLIC TRANSPORTATION SYSTEM IN YANGON REGION

[1]MYAT THU ZAR, [2]MYINT MYINT SEIN

[1]University of Computer Studies, Yangon, [2]University of Computer Studies, Yangon
E-mail: myatthuzar86@gmail.com, myintucsy@gmail.com

Abstract- Building public route transportation system is the key technology for urban areas. Most of citizens in Yangon area make use of public transportation such as bus, taxi, rail and so on. In this system, A* algorithm is used to find the bus route information in Yangon downtown area. Bus routes, bus information, bus stops and their related position, latitude/longitude and road names are stored in spatial database. When the users want to know the bus route information, firstly they give their current location and desired destination to the system and then the related bus routes are retrieved from database. After getting the bus routes, A* algorithm is used to calculate the optimal path. Finally, all of the possible routes are display on the map. OpenStreetMap is used to create as base map in this paper.

Keywords- A* Algorithm, Openstreetmap, Road Network, Public Transportation System.

I. INTRODUCTION

The traffic problem of Yangon area has been directly affects the quality of the citizens of the area. The effects of the traffic of Yangon area are the pollution in every aspect such as air, sound, the time wasting on the road and also the hydro-carbon fuel. Even though the numbers of the roads are increasing, the traffic problem in Yangon area is still occurring. Advanced public transportation systems are an important component of smart transportation systems for reducing traffic demands. We report algorithms for providing bus information to tourists and commuter to encourage people to use public transportation systems. Providing transfer information is another effective measure for promoting the public transportation systems.

There are a number of algorithms that can be used to determine shortest route between two nodes in a network. Among them Dijkstra's algorithm and A*algorithm are more efficient. Dijkstra's algorithm determines the shortest route between the source node and every other node and A* algorithm much like Dijkstra's based on heuristics strategy.

An efficient public transport system, that integrates residential and employment hubs, plays an important role in moving skills, labor and knowledge within and between markets, stimulating productivity. Public transport corridors have shown to create activity centers that encourage and foster transit orientated development, and in turn through providing easy access to public transport increase adjacent property values.

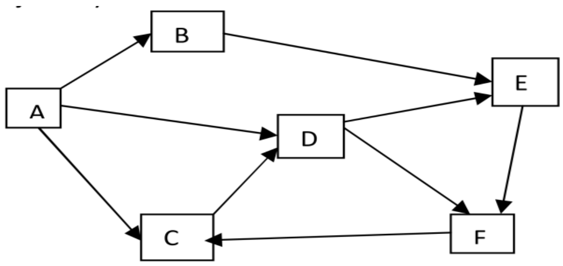
The remainder of the present paper is organized as follows. Section II describes the basic representative network structure. In Section III describes the applied method A* algorithm and related function for this system. Section IV briefly explains about system overview and data. Section V discuss about how to do experiment the proposed system on Yangon downtown region. Finally section VI summarizes present paper and describes future plan.

II. ROAD NETWORK REPRESENTATION

A public road network is composed by some nodes (bus stop) in this paper, the links connecting two nodes and bus lines. Define a public road network as G, G={N,E,R}, where N={$1 \leq i \leq n$} denotes the set of all nodes, and n is the number of nodes; the origin node and the destination node is O, D respectively. E={$1 \leq e \leq m$} is the set of all transit links, and m is the number of links; R={ $1 \leq r \leq u$ } is the set of all bus lines and u is the number of links. In this paper, algorithm employs two functions that relate bus stops and bus routes. We use the distance on the road network as the cost in this paper.

A. Adjacency Matrices

In this paper, we represent the connectivity among locations in transportation networks with adjacency matrices. We designate each location in the network a unique number. The value of a cell $M_{i,j}$ of an adjacency matrix M is set to the number of direct ways that we may travel from a location I to j. The following example, fig(1) illustrates how we represent a simple road network with an adjacency matrix T and distance matrix.



$$T= \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Adjacent matrix

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & \infty & \infty \\ \infty & 0 & \infty & \infty & 1 & \infty \\ \infty & \infty & 0 & 1 & \infty & \infty \\ \infty & \infty & \infty & 0 & 1 & 1 \\ \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & 1 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

Distance Matrix

Fig1. Example bus route and its adjacency matrix and distance matrix

B. Distance Matrices

Given a weighted graph that represents a street network, a problem instance consists of a number of sources and destinations located in the graph. For these sets of nodes we want to know the distances from all sources to all destinations. Hence the result of such an M×N query is a matrix of distances. An entry of this distance table denotes the distance from the source corresponding to the current column, to the target corresponding to the current row.

C. OpenStreetMap

The OpenStreetMap is used as a base map in this paper. OpenStreetMap can be used offline, without an internet connection, in a number of ways.

In fact some offline offering can be vastly superior to those of other map data providers because OpenStreetMap is free (zero cost) and we are free (as in freedom) to download data in bulk.

By contrast, commercial map provider will often deliberately use technical barriers to limit offline storage/distribution of their data.

II. SYSTEM METHODOLOGY

A. A* Algorithm

A* algorithm is a graph search algorithm that find a path from a given initial node to a given goal node in a mapped area. The A* algorithm by Hart and Nilsson formalized the concept of integrating a heuristic into a search procedure.

It employs a 'heuristic estimate' h(n) that gives an estimate of the best route that goes through the node. It visits the nodes in order of this heuristic estimate. It follows the approach of best first search.

A* algorithm are the building of a "closed list" to record areas already evaluated, an "OPEN list" to record areas adjacent to those already evaluated, and the calculation of distance traveled from the "start point" with estimated distance to the "goal point". The heuristic used to evaluate distances in A* is:

$$f(n)=g(n) + h(n) \qquad (1)$$

Where -g(n) is the cost of the path from the starting node to any node n, and

-h (n) is the heuristic estimated cost from any node n to the goal

B. A* Algorithm Steps

1. Create a search graph, G, consisting solely of the start node, n_0. Put n_0 on a list called OPEN.
2. 2. Create a list called CLOSED that is initially empty.
3. If OPEN is empty, exit with failure.
4. Select the first node on OPEN, remove it from OPEN, and put it on CLOSED. Called this node is n.
5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n_0 to n_n in G. (The pointers define a search tree are established in step 7).
6. Expand node n, generating the set, M, of its successors that are not already ancestors of n in G. Install these members of Max successors of n in G.
7. Establish a pointer to n from each of those members of M that were not already in G (i.e., not already on either OPEN or CLOSED). Add these members of M to OPEN. For each member, m, of M that was already on OPEN or CLOSED, redirect its pointer to n if the best path to m found so far is through n. For each member of M already on CLOSED, redirect the pointers of each of its descendants in G so that they point backward along the best path found so far to these descendants.
8. Record the list OPEN in order of increasing f(n) values
9. Go to step 3.

C. Heuristic Function

Heuristic search has been widely used in both deterministic and probilistic planning. The heuristic function can be used to control A*'s behaviour.

Euclidean distance is a common method for h(n).

$$Distance=sqrt(dx^2+dy^2) \qquad (2)$$

- x_2 = coordinate of the goal location
- x_1=coordinate of the current location
- y_2=coordinate of the goal location
- y_1=coordinate of the current location
- $dx=|x_2-x_1|$
- $dy=|y_2-y_1|$

If h (n) is always lower than to the cost of moving from n to the goal, then A* is guaranteed to find a shortest path. If h(n) is exactly equal to the cost of most of moving from n to the goal, then A* only follow the best path and never expand anything else, making it very fast. If h(n) is sometimes greater than the cost of moving from n to the goal, then A* is not guaranteed to find a shortest path, but it can run faster. At the other extreme, if h(n) is very high relative to g(n), then only h(n) plays a role and A* turns into Best-First-Search.

III. SYSTEM OVERVIEW

In this paper, the bus transportation information for Yangon downtown region such as bus stops name, bus routes name, bus type, latitude and longtitude position are stores in spatial database. Time complexity in A* algorithm is O(n log n), n is the number of node. This system can be used without internet connection.

In fig2, show the example data of bus stop and bus route. When the users enter the place name (bus stop name) in the search box and in return the map shows a marker at the place. In fig.3 show the sample result for this system. In fig 4, the process of system shows flow diagram. The process of associating a place name with coordinates on the map is called geocoding. In a spatial database, this data done as a point layer with the name of the place as an attribute to the location. This is one way of geocoding. For addresses, the associated coordinates are not saved in a database directly, but computed using a method called linear referencing. The start and end addresses along a long segment are saved and intermediate address are interpolated and the coordinates are calculated.

In some online mapping service, the user may have seen satellite imagery. When these images are captured from a satellite or an airplane, they are just plain images, like photographs. But to display these images on a map, they need to be associated with map coordinates. This process is called GeoReferencing. Once the image is associated with the map coordinates it can be overlaid on top of street maps. For georeferencing, some user can use a GIS software such as ArcGIS or QGIS to georeference and otherwise un-referenced image or scanned map, and load them into Oracle spatial.

D. User Interface

Realistic bus-information provision system much have a good user interface do deal with naming problems of bus stops. Although we may provide a user interface, we cannot assume all users will locate their origins and destinations on a map. Any user-friendly system must prepare to accept queries that use text input. Such a demand requires our system to manage mapping between text inputs for location names and stop names gracefully.

No	BusName	Type
1	48	Specail
2	124	Special
3	202	Special
4	43	Mini Bus
5	51	Big
6	135	Mini Bus
7	141	Myawady
8	168	Dyna
9	220	Hilat
10	204	Minin Bus
11	34	Special
12	42	BM
13	38	Special
14	173	Paramie
15	333	Special
16	226	Red
17	231	Paramie
18	105	(Mini Bus)

No	BusStopNames	Latitude	Longtitude
1	pharlan	16 51 29.57	96 07 24.06
2	thamine	16 51 38.48	96 07 21.22
3	kalarkyaung	16 51 53.94	96 07 16.10
4	khawelchan	16 52 10.94	96 07 10.74
5	bpi	16 52 31.28	96 06 58.78
6	gyo gone	16 52 43.67	96 06 46.02
7	boc	16 53 03.84	96 06 37.23
8	pyitawthar	16 54 03.78	96 05 49.58
9	danyingone	16 56 01.94	96 06 06.91
10	kantharyar	16 57 01.40	96 06 02.62
11	kyaunglan	16 57 10.53	96 05 58.87
12	sittat	16 57 28.72	96 05 51.13
13	shwepyithargwe	16 57 41.35	96 05 48.24
14	gatehong	16 57 53.13	96 05 46.70
15	thanphyusine	16 58 19.86	96 05 39.72
16	oakhpo	16 58 27.99	96 05 39.72
17	thardu	16 59 00.89	96 05 36.14
18	computer	16 59 55.52	96 05 17.45
19	okkyin	16 51 18.40	96 07 25.92
20	bartar	16 51 02.68	96 07 28.68
21	thanlan	16 50 43.04	96 07 33.19
22	thukha	16 50 25.00	96 07 37.54
23	butaryonelan	16 50 13.94	96 07 39.75
24	sinyaytwin	16 50 04.29	96 07 42.63
25	sanveiknvein	16 49 51.93	96 07 46.76

Fig 2. Examples data of bus stops and bus routes

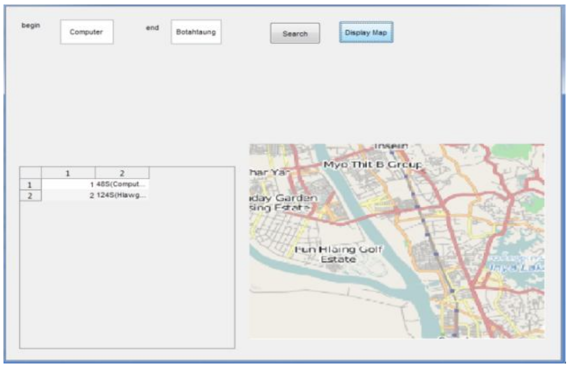


Fig3. Example data output GUI

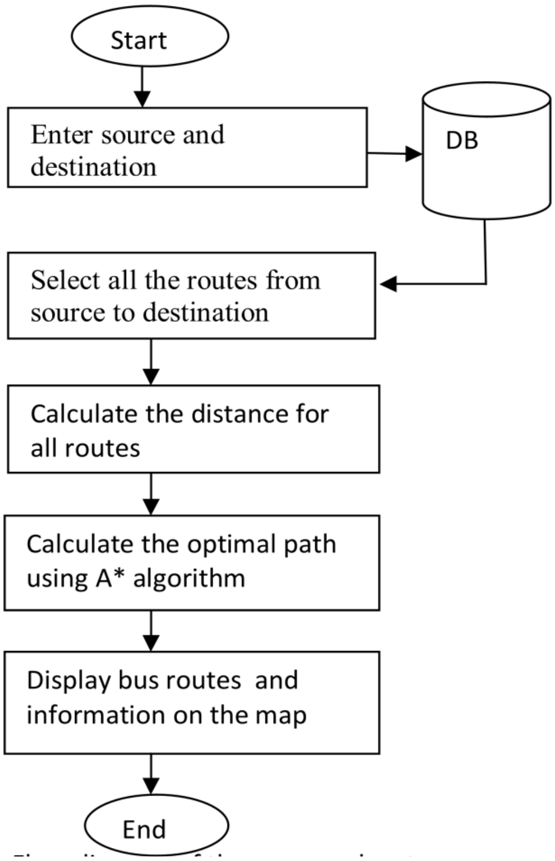


Fig4. Flow diagram of the proposed system

IV. EXPERIMENT

In this paper, the proposed system is tested on Yangon down town region, road network. Collect the bus routes information, bus stops, road name and their related geographical information data such as latitude/longtitude and position in Yangon downtown area are collected from Google earth and use GPS GARMIN etrex-10 device. Yangon down town region is limited from left latitude 17·03' 56.63" N to right longtitude 96· 19' 04.15"E.These collecting data are stored in spatial database which will use to develop the A-star algorithm of graph theory to calculate the shortest route. The system will be able to show the data to the user to select the optimal route and bus number with information of public bus transport for downtown region in Yangon.

CONCLUSION

Public bus transportation system by A* algorithm apply in Yangon downtown area will assist the bus transportation user to the information and data which is essential for them and also provide the planning system to plan the route and transportation mode with the fare. A* is better when we know both starting point and destination point.

A* is both complete (finds a path if one exists) and optimal (always finds the shortest path) if the user use an Admissible heuristic function. As a future work, the time or other factors such as waiting time at the bus stop, walking time to the bus stop will be used as weight for road network in the whole Yangon area.

REFERENCES

[1] Lillian S.C. Pun- Cheng,‖ An interactive web based public transport enquiry system with real time optimal route computation‖,IEEE Transactions on Intelligent Transportation Systems,Vol.13,No.2,June 2012.

[2] John Pucher and Nisha Korattyswaroopam, Rutgers University Neenu Ittyerah, ―The Crisis of Public Transport in India: Overwhelming Needs but Limited Resources‖, Volume 7, Number 3., 2004

[3] Hanan Samet, University of Maryland,USA-" Database and Geographical Information System(GIS)- 2002.

[4] Cho-L.L, Tun-W.P,Chun-T.C and Chang-M.H. "Path-Planning Algorithms for Public Transportation Systems", the 4th international IEEE conference on Intelligent Transportation System,USA,August 2001.

[5] Lang Dai, " Fast Shortest Path Algorithm for Road Network and Implementation", Carleton University if Computer

[6] Science COMP 4905 HONOURS PROJECT Fall Term, 2005. Subadra; M.Bhagavan Das; and C.Rama Seshagiri Rao," Directed Graph Algorithms for Tours – A Case Study", Scholarlink Research Institute Journals, (ISSN: 2141-7016),2011.

[7] Abhishek-G, Prateek-M, Rishabh.L and Ms. N.S "PATH FINDING: A* OR DIJKSTRA'S?", IJITE, Vol.02 Issue-01, (January, 2013)

[8] Andrew V. Goldberg, Chris Harrelson,"Computing the Shortest Path:A_ Search Meets Graph Theory", Technical Report MSR-TR-2004-24.

[9] Sebastian.K, Peter.S, Dominik.S and Frank.S, " Fast Computation of Distance Tables using Highway Hierarchies" –July 25,2006.

[10] http://wiki.openstreetmap.org/wiki/Offline_Openstreetmap

[11] http://gis.stackexchange.com/questions/28593/what-is-the-difference-between-geocoding-and-georeferencing

[12] Andrew V. Goldberg, Chris Harrelson,"Computing the Shortest Path:A_ Search Meets Graph Theory", Technical Report MSR-TR-2004-24.

★★★