Improving delivery service applying shortest path algorithm for large Road Network

Aye Lai Lai Soe, Myo Kay Khaing
University of Computer Studies, Yangon
ayelailaisoe123@gmail.com ;

Abstract

Graphs and networks are collections of nodes and arcs.Numeric values on the links can represent the actual length of the link. This system presents the delivery service problem by means of shortest path algorithm. Suppliers have to find the path plan to the place where their products are being delivered, especially. in reference to the geographic position of the objects, its surroundings and the shortest path of the destination place from their current place. For large road network, the method is based on preprocessing and prepared for static graphs, i.e, graphs with fixed topology and edge costs.

This system solves the suppliers' Facing problem in finding the shortest path by using A Shortest Algorithm. A* is the most popular algorithm being used in finding the shortest path because of its flexibility and potentially search in a huge area of the map.*

1. Introduction

Shortest path algorithm is a program or set of directions that can be executed to provide the shortest path between the locations given. Delivery problem is the path planning problem while delivery service is the process of delivering an object to its destination aim to minimize the delivery time and save cost in the road network.

One of the class problem that was solved related to find the shortest path between two points (nodes) on a network. A* is a network searching algorithm that takes a 'distance-to-goal +path-cost' score into consideration. As it traverses the network searching all neighbours, it follows lowest score path keeping a sorted priority queue of alternate path segments along the way. If any point being followed has a higher score than other encountered path segments, the higher score path is abandoned and a lower score subpath traversed instead. This continues until the goal is reached.

This paper is organized as follows. Section 1 is the introduction, section 2 is related work. Route planning in road network is presented in section 3. In addition, section 4 presents Graph Theory and shortest path algorithms. A Star algorithm has been described in section 5. Section 6 is the proposed system design and section 7 is the system implementation and sample case study for A* algorithm. Section 8 is the conclusion and future work of the system.

2. Related Work

Finding the shortest path between two vertices is a common problem to be solved in many fields including the area of robotics, telecommunication, transportation and etc. There are different methods of shortest path algorithm. In [9], "On a Routing problem" has introduced one of the first algorithms in solving the single source problem. The algorithm is combined with the concept of approximation in policy space to solve the system of non-linear equations. The first formal shortest path algorithm was proposed by Dijkstra [3] which solves the single-source shortest path problem by using only required simultaneous storing data. In [4] Floyd Warshall algorithm has been proposed for all pairs shortest path problem.

Although various algorithms exist for finding the shortest path, their performance tends to deteriorate as the network size increases (Karimi,1996;Zhan & Noon,1998). These algorithm can ensure the optimal solutions and its efficiency has also been tested by practice. Dijkstra's algorithm [Dijkstra, 1959] is the most well-known algorithm for determining the shortest path from one location to all other locations in a road network. If the number of stations is large, the efficiency of Dijkstra algorithms decreases sharply. If only a shortest path between two locations has to be determined, Dijkstra's algorithm can be speeded up by taking an estimation of the cost from a location to the destination into account. This algorithm is called A* algorithm. The key of heuristic algorithms is to describe and define an evaluation function and this will play an important role to obtain the final results. If the used estimation satisfies a few conditions, then the A*algorithm can be used to plan the shortest path between two nodes in a road network.

3. Route Planning

In a process for determining a route from a predetermined starting point to a predetermined destination, firstly, weights are assigned to route segments interconnected at route nodes. At least a first weight relating to a first route segment is varied by querying at least a data source based on at least a second weight relating to a second route segment. The second route segment is prior to the first route segment in a consideration of the route from the starting point to the destination. Subsequently, the route is determined by using a route finding algorithm, the algorithm taking the weights into account.

Substantial effort has been exerted in the past and is continuing to be directed toward route planning between origins and destinations in a variety of networks. Such examples include directing traffic in a manner to facilitate travel by individuals between a wide variety of origin and destination combinations in an optimized amount of time, distance, cost or the combination of those measurements.

3.1. Route Planning Process

A process for determining an optimized route from a predetermined starting point to a predetermined destination using a route finding algorithm, the route including route segments and route nodes, the route segments interconnecting at route nodes, the route segments having weights assigned thereto, the process comprising the steps of: (a) determining an initial optimized route based on the weights assigned to the route segments between the predetermined starting point and the predetermined destination; and (b) predetermining the optimized route during execution of the algorithm during the initial optimized route, wherein at least one of the route segments is determined by the cumulative weights of the preceding route segments, and at least one of the route segments has a weight which is dynamically variable during the optimized route.

4. Graph Theory

Graph theory developed a topological and mathematical representation of the nature and structure of transportation networks. However, graph theory can be expanded for the analysis of real-world transport networks by encoding them in an information system. In the process, a digital representation of the network is created, which can then be used for a variety of purposes such as managing deliveries or planning the construction of transport infrastructure.

The "standard" model of a road network is a weighted, directed graph model G=(V,E) (sometimes also called a network model) consisting of a set V of nodes and a set E of directed edges that connect pairs of nodes. In the graph G, the nodes in V are intersections in the road network, while we use an edge e=(u,v) to represent the road segment that connects the intersections u and v in the road network – going from u to v. We also assign a weight, w(e), on each edge e in the graph G. A route in the road network from a source point s to a destination point t will then be a path P(s,t) in the graph from the source node s to the destination node t. The "length" of the path P(s,t) is then the sum of the weight of the edges along the path P(s,t) from node s to node t. Then the single-mode route advisory problem is modeled as the well-known and well-studied and well-solved shortest path problem of finding a shortest path from node s to node t in the weighted, directed graph G=(V,E). Figure 1 presents weighted graph, where values on edge are weights or distances or costs between nodes.

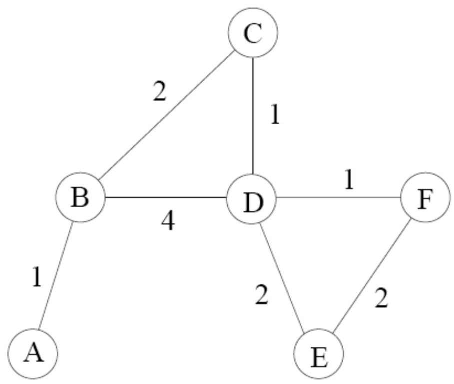

Figure 1. Weighted Graph in Road Network

4.1 Shortest Path Algorithm

Graphs are useful for representing many problems in computer science and in the real world. Applications of graph representations range from the seemingly simple, finding out whether a node is reachable from another node, to the extremely complex, such as finding a route that visits each node and minimizes the total time (the "travelling salesman" problem). A common, but solvable problem is that of problem of simple path finding. Generally, the task is determining the shortest path from a given node to any other node on the graph. In this paper, A* algorithm is used to find the optimized routing plan.

5. A* Algorithm

A* (pronounced "A star") is a computer algorithm that is widely used in pathfinding and graph traversal, the process of plotting an efficiently traversable path between points, called nodes. A* uses a best-first search and finds the least-cost path from a given initial node to one goal node.

It uses a distance-plus-cost heuristic function (usually denoted F(x)) to determine the order in which the search visits nodes in the tree. The distance-plus-cost heuristic is a sum of two functions:

- the path-cost function, which is the cost from the starting node to the current node (usually denoted G(x))
- and an admissible "heuristic estimate" of the distance to the goal (usually denoted H(x)).

The H(x) part of the F(x) function must be an admissible heuristic; that is, it must not overestimate the distance to the goal. Thus, for an application like routing, H(x) might represent the straight-line distance to the goal, since that is physically the smallest possible distance between any two points or nodes.

If the heuristic h satisfies the additional condition for every edge x, y of the graph (where d denotes the length of that edge), then h is called monotone, or consistent. In such a case, A* can be implemented more efficiently.The formula of A* algorithm is as follows:

F(n) = G(n) + H(n)

Where:
F (n) = total cost
G (n) = the cost of the path from the start point to any vertex n
H (n) = the heuristic estimate cost from vertex n to the goal.

Using this formula, a suitable shortest path route with less traveling time can be suggested to users. A* algorithm needs heuristic estimate distance to target node.

H can be estimated in a variety of ways. This paper uses Manhattan distance algorithm to compute the heuristic distance (H). In Manhattan method, the total number of squares moved horizontally and vertically is computed to reach the target node from the current node, ignoring diagonal movement. It is the most suitable for transportation (road) networks since nodes cannot be reached directly in diagonal movement. [2] According to Manhattan distance method, distance from P to Q can be computed as:

$$d(P, Q) = \sum_{i=1}^{2} |P_i - Q_i|$$

5.1. Process of A* Algorithm

Process of A* algorithm is shown in the following.

1) Add the starting square (or node) to the open list.
2) Repeat the following:
   a) Look for the lowest F cost square on the open list. This is referred as the current node.
   b) Switch it to the closed list.
   c) For each of the adjacent nodes to this current node.
   If it is not walkable or if it is on the closed list, ignore it. Otherwise do the following.
   If it isn't on the open list, add it to the open list. Make the current square the parent of this square. Record the F, G, and H costs of the square.
   If it is on the open list already, check to see if this path to that node is better, using G cost as the measure. A lower G cost means that this is a better path. If so, change the parent to the current node, and recalculate the G and F scores of the node. Open list is sorted by F score.
   d) Calculation is Stopped:
   Add the target node to the closed list
3) Save the path. Working backwards from the target node, go from each node to its parent node until you reach the starting node. That is the shortest path.

6. Proposed System

This system presents solving the delivery problem in the road network for the transportation to different supermarkets in the city. A Star algorithm is used to find the shortest path between source and destinations. This system is implemented as one source and a set of destinations or nodes (supermarkets in the city) and junctions. Source, destinations and junctions are denoted as nodes in this system and compute the shortest path between them.

Weights on the graph are actual distance from one node to other node. Delivery process is performed according to the shortest path generated by the system. Process flow of the system is shown below:

- When user enters orders and amount into the system, this system processes the following to compute the shortest path.
- It draws graph of the destinations and nodes and set weights (distances) onto the graph edge.
- Destinations (nodes) and distances (weights) are stored in the database in prior.
- Then A* algorithm is applied to compute the shortest path.

- After computing the shortest path, delivery process is performed according to the shortest path.

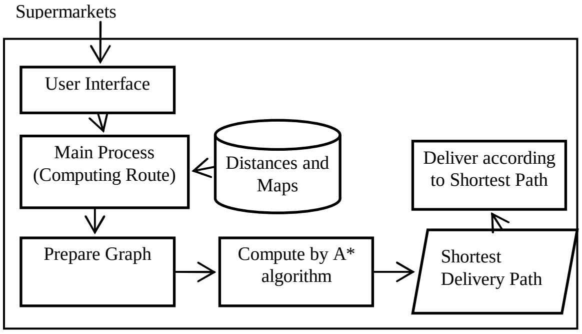

Figure 2: Process Flow of the System

7. System Implementation

The implementation represents the delivery process from source location to one or more destinations. Road networks are the graph model of source and a set of destinations. Nodes and distances are stored in the database. Graph is built according to the distances stored in the database. Then, A* algorithm computes the shortest distance. Delivery process is performed according to shortest path generated. This program is performed according to user request. For example, if user selects 7 targets (supermarkets) then, the computation is for those 7 targets. But if path to one target consists of other target as intermediate node then, the intermediate node is omitted in computing. Java programming with jdk version 1.6.0 is used to implement the system.

7.1. Data Collection

In our system, there are several nodes, including supermarkets and junctions and one source location. Mingaladon Industry is the source node in this system. Weights generated from this system are distances, scaled down according to Yangon Road Map. Bridges, companies, etc. are used as junctions in some of the junctions except supermarkets.

7.2. Database Design

In this paper, nodes (supermarkets, junctions and source) information is saved in the database. Their coordinates are also saved to compute the heuristics estimate distance from source to destination. There are three tables in the database, Nodes table for storing nodes information and Distances table for storing distances (weight in graph) between nodes, and Transaction table for order information. It is shown in Figure 3.

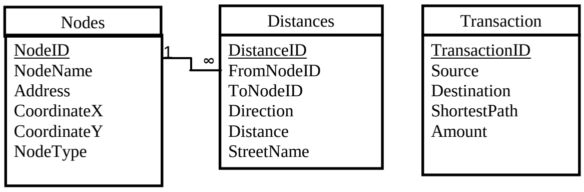

Figure 3: Database Table Design

7.3. Case Study

In this paper, sample case study to find shortest path from source and destination is demonstrated with road network structure shown in Figure 4.

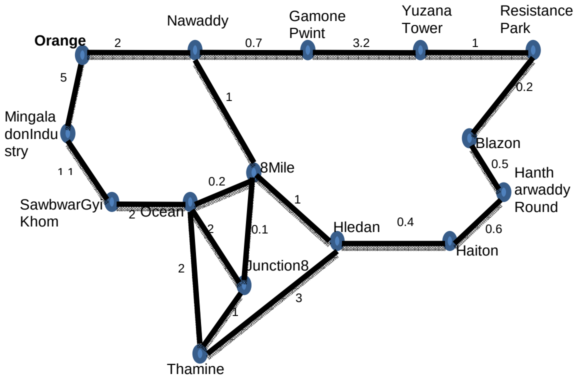

Figure 4. Sample Road Network

Source = MingaladonIndustry (Mingaladon)
Destination = Blazon
→f (Sawbwargyigone) = 1.1+1.9=3
f (Orange) = 5+2.5=7.5

→f (Ocean) = 3.1+1.1=4.2
f (Orange) =7.5

→f (J 8) = 5.1+0.8=5.9
f(8mile)=3.3+0.8=4.1
f(Thamine)=5.1+1.5=6.6
f (Orange) =7.5

→f (Hledan) = 4.3+0.7=5
f(J8)=5.9
f(Thamine)=6.6
f (Orange) = 7.5
→f(Haitai co.ltd)=4.7+0.6=5.3
f(J8)=5.9
f(Thamine)=6.6
f (Orange) = 7.5

→ f(Hanthawaddy Round)=5.3+0.4=5.7
f(J8)=5.9
f(Thamine)=6.6
f(Orange)=7.5
f(Blazon)=5.8

Shortest Path form Mingaladon Industry to Blazon= 5.8
Shortest Path
Mingaladon Industry SawbwarGyeKnom, Ocean, 8Mile, Hledan, Haitan Co.ltd, Hantarwaddy Round, Blazon

8. Conclusion

This system introduces the delivery problem and solves the problem using A* algorithm. It supports for delivering system in which the customer can get our product with a short time by using A* Algorithm. Delivering system is now widely being used in many business areas. So, it can be concluded that finding the shortest path in a road network using A* Algorithm, can optimize road finding process. Moreover, it can save traveling time and transportation costs for suppliers when they want to find shortest path for delivering customer's order by using A* Algorithm.

In this system, only static variables are computed and it can be further extended with the inclusion of dynamic variables such as road traffic condition, traveling time, etc.

9. References

[1] C. M. Flinsenberg, "Route Planning Algorithms for Car Navigation", Development Center of Siemens VDO Trading B.V, I.C.M. Flinsenberg 2004.

[2] E. P., Stoffel, "A Research Framework for Graph Theory in Routing Applications", 15. November 2005.

[3] E. W. Dijkstra, "A note on two problems in connexion with graphs", Numerishe Mathematik, 1959, pp. 269-271.

[4] Floyd, Robert W., "Algorithm 97: Shortest Path", Communications of the ACM, 1962.

[5] H. Xu, Z. Chen, "Solving a Practical Pickup and Delivery Problem", Department of Systems Engineering, University of Pennsylvania, January 2001.

[6] M. Idris, S. Bakar, E. Tamil, Z. Razak and N. Noor, "A Design of High-Speed Shortest Path Coprocessor", MASAUM Journal of Basic and Applied Sciences, Vol. 1, No. 3, October 2009

[7] P. Biswas, P. K. Mishra and N. C. Mahanti, "Computational Efficiency of Optimized Shortest Path Algorithms", International Journal of Computer Science & Applications, 2005.

[8] Q. Rong, W. Min, D. QuingYun and C. ZhongLiang, "COMBINING ALGORITHM WITH KNOWLEDGE FOR WAY-FINDING", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008.

[9] R.Bellman, "On a Routing Problem", Quarterly of Applied Mathematics", 1958, pp. 87-90.

[10] S. Knopp, P. Sanders, D. Schultes, D. Wanger and F. Schulz, "Computing Many-to-Many Shortest Paths Using Highway Hierarchies", Copyright by SIAM, 2008

[11] Z. Jianghua, L. Zhijui and Z. Xin, "Burning Algorithm for Route Planning of Multi-Mode Public Transport Networks", College of Resource and Environment Sciences, Xinjiang University, Urumqi 830046, Xinjiang, China, 2003.

[12] http://www.policyalmanac.org/games/heuristics.htm

[13]http://en.wikipedia.org/w/index.php?title=A*_search_algorithm