Shortest Path Finding System by using Transitive Closure and Dijkstra's Algorithms

K-zin Phyo, Thandar Win
University of Computer Studies, Mawlamyine
kzinphyo09 @ gmail.com,thandarwin77@gamil.com

Abstract

Nowadays, optimization/minimization is a key factor in almost all topics of operation research, management science and economic. Road networks can be minimized within different constraints like time, distance, cost and traffic running on the road. This study is based on distance to calculate shortest path. This paper is intended to present shortest path finding system on Mandalay road map. In this system, Transitive Closure and Dijkstra's algorithms are used to find the shortest path. Transitive Closure algorithm is used to get all pairs of paths and the result is used to check whether there is a path between user chosen source and destination. If the pair of path exists, then Dijkstra's algorithm calculates shortest path of user desired source and destination. And then the result of shortest path is display to user. This system presents how it can be applied Transitive Closure and Dijkstra's Algorithms on Mandalay road map. In this system, the locations of Mandalay as vertices and associated distance between each location as weight of edges of a directed graph.

Keywords: **Transitive Closure, Dijkstra, Algorithm**

1. Introduction

The shortest path problem is one of the most important in the fields such as management science, computer science and artificial intelligent. An example is finding the quickest way to get from one location to another on a road map. In the shortest paths problem, a graph is presented with real weights on the edges and then finds the optimal path between two nodes that is the minimum weight over all paths between the nodes.

The primary application for this problem is finding an itinerary for a trip between two physical locations that gets you to your destination as early as possible [2]. In theoretical computer science, these problems have been studied alongside since the 1950s. These applications are also apply until now. Other applications of path problems include robot motion planning, highway and power line engineering, network connection routing, various scheduling problems sequence alignment in molecular biology and so on [4]. By using shortest path finding system can reduced the cost such as traffic, time and money.

The main objective of developing this system is to study the nature of Transitive Closure and Dijkstra's algorithms. By using these two algorithms, the system produces all possible paths and shortest path in a short time. In section 2 of this paper is described related work. Background theories of this system are discussed in section 3. The system design and implementation are described in section 4. In section 5, experimental result of system is reported and conclusion is discussed in section 6.

2. Related Work

The paper [3] assists travelers with planning, perception, analysis and decision making to improve the convenience, safety and efficiency of travel. The main purpose in [5] is to evaluate the computational efficiency of optimization shortest path algorithms. In [6], the nature of the system is based on optimatization of real road network by means of distances. The paper tried to refer the shortest path problem by using Dijkstra's algorithm [2]. Paper [9] is representing a general framework for shortest path algorithm including among others Dijkstra's algorithm. An algorithm is given for computing the transitive closure of a directed graph [8]. In this paper, Transitive Closure algorithm is used to know all possible paths and Dijkstra's algorithm is used to find shortest path in a short period of time.

3. Background Theories

3.1 Shortest Path Finding Problems

Let G be a directed graph. The graph G* which has the same vertex set as G, but has an edge from v to w if and only if there is a path (of length 0 or more) from v to w in G, is called the reflexive and transitive closure of G. A problem closely related to finding the transitive closure of a graph is the shortest path problem. Associated with each edge e of G, the cost, c (e), is nonnegative. The cost of a path is defined to be the sum of the costs of the vertices (v, w), the lowest cost of any path from v to w. It shows the idea behind the best algorithm known for both the transitive closure and shortest path problems of finding the (infinite) set of all paths between each pair of vertices [4].

3.2 Transitive Closure Algorithm

Transitive closure algorithm can be evaluated by a variety of many criteria. It is closely related to the all-pair shortest path problem. To compute the transitive closure of a graph, it is defined the labeling function

$$L(v, w) = \begin{cases} 1, & \text{if } (v, w) \text{ is an edge} \\ 0, & \text{if not} \end{cases}$$

Then c (v, w) =1 if and only if there is a path of length 0 or more from v to w. On the diagonal A[i] [j] =1 iff the component i has a least two vertices [1]. The procedure of Transitive Closure algorithm is as follow:

Input: Directed graph with nodes (e.g. v = 1 ,…, n) and edges (e.g. $v_i \rightarrow v_j$, $v_j \rightarrow v_k$,…).

Output: Matrix to show whether edge is exists or not

```
Transitive-closure (A) {
   n = A. last
   for k = 1 to n
   for i = 1 to n
   for j = 1 to n
  A[i] [j] = A[i] [j] v (A[i] [k] ∧ A[k] [j])
 }
```

3.3 Dijkstra's Algorithm

The algorithm finds one vertex to all other vertices and it also called single source shortest path algorithm [8].The algorithm visits all nodes (connection points between links) starting at the start point of the route and determines the 'cost' (such as time, distance, money) to reach that node from the starting point. The nodes are visited by using the connectivity between links, starting at the start point supplied. In this way, each node need only visited once per run of the route calculation algorithm. Once the destination node has been reached, the shortest path can be read from the temporary cost data that has been calculated for each node. If all the nodes connected to the start point have been visited and the destination node hasn't yet been visited, then the route cannot be calculated since there is no connectivity between source and destination. In Dijkstra's algorithm weight of edges have one of three values [7]. The procedure of Dijkstra's algorithm is as follow:

$$W(u,v) = \begin{cases} 0, & \text{if } u=v \\ \infty, & \text{if } (u,v) \text{ is not an edge} \\ \text{The weight of edge } (u,v), & \text{Otherwise,} \end{cases}$$

```
for each u G:
      d[u] = infinity;
      parent[u] = NIL;
  end for
     d[s] = 0; // s is the start point
     H = {s}; // the heap
  while NotEmpty (H) and targetNotFound:
    u = Extract_Min(H);
   label u as examined;
     for each v adjacent to u:
        if d[v] > d[u] + w [u, v]:
          d[v] = d[u] + w [u, v];
           parent[v] = u;
     DecreaseKey [v, H];
```

4. System design and Implementation

4.1 System Design

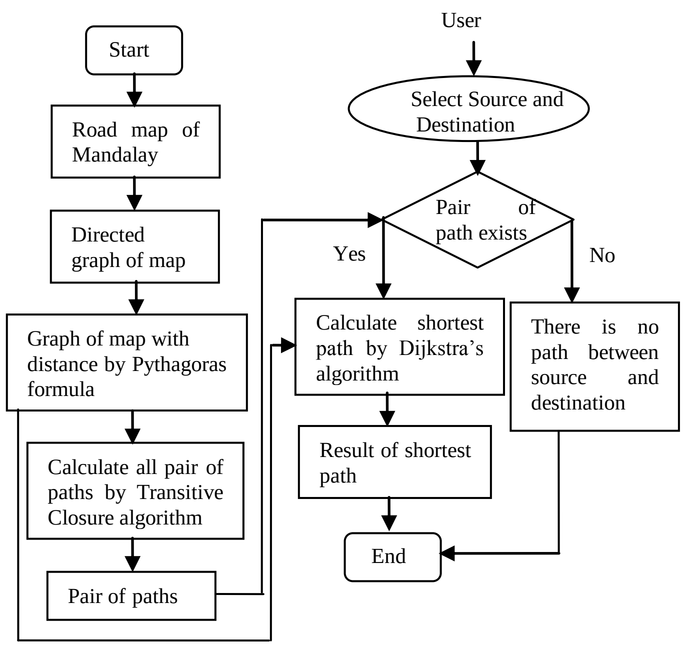

Figure 1: System Design

Mandalay road map is used to implement this system and the locations of Mandalay are used as nodes and the associated between those locations as edges. The over view of system design is shown in figure 1. The positions of nodes are defined on X, Y plane. The distances between nodes are calculated by Pythagoras formula. And then, all possible path of directed graph are calculated by Transitive Closure algorithm. The result of this algorithm is used to check whether there is the path between user desired locations. In this system user must choose source and destination. If pair of path between source and destination exists, the system calculates shortest path by Dijkstra's algorithm. And then the result of the path is display to user. Otherwise, the information that is no path between source and destination message is replied to user.

4.2 Implementation of the system

This system is a web-based system and is developed by C# language. To implement this system, Window 7 is used as the operating system and Mozilla Firefox is used as browser. ASP development server is used as web server and SQL Express is used for data storage. This system is intended to help visitors who want to know the shortest path between two places (i.e. his/her source/current and destination/target places). This system uses the Mandalay City as directed graph. There are two main processes in this system,

(1) Administrator process and

(2) User process.

These processes are shown in the following sub sections.

4.2.1 Administrator Process

In administrator process, the administrator can login with user name and password. If name and password are correct, user can modify the Map. The administrator input the locations of Mandalay and defines the path between these locations to construct directed graph. The example directed graph of the system is shown in figure 2. The directed graph of the system can vary because vertices and path are not predefined. The distances between the locations are calculated by Pythagoras Formula. The directed graph of map is applied by Transitive Closure algorithm to calculate all possible paths of Map. The result of Transitive Closure algorithm is used to check whether there is a path between source and destination. Administrator can easily add/remove nodes and paths because vertices and edges are not limited in this system. Furthermore, the administrator can find shortest path by choosing source and destination.

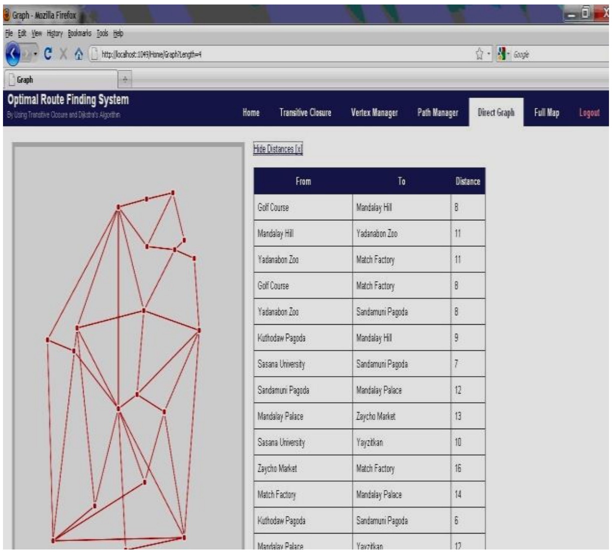

Figure 2: Example Directed graph

Example vertices and paths of the system are as follows:

(1)Mandalay Hill→ Mandalay Place→ Institute of Medicine→ Teacher Training School

(2)Kyauktaw Gyi Pagoda →Sandamuni Pagoda→ Kuthodaw Pagoda →Sasana University

(3)Thaungthaman Lake →Yadanabon University→ Myohaung Market

4.2.2 User Process

The user must choose the desired source and destination to find shortest path and check whether there is a path between his/her desired locations. If user chosen path exists, the system calculates shortest path by Dijkstra's algorithm and displays the result of shortest path with the line and also displays the total distance of the path. Otherwise, the information that is no path between source and destination is replied to user.

For example, the two shortest paths between Myanmar Electric Power Enterprise to Highway Bus Station and Highway Bus Station to Myanmar Electric Power Enterprise are illustrated with figures. Although these two locations are situated in 78^{th} street, the total distance and the locations on the path are different because 78^{th} street is assumed one way street in this system (i.e. one way street has one direction).

The shortest path of Myanmar Electric Power Enterprise to Highway Bus Station is shown in figure 3. In figure 3, the line red is the result of shortest path between Myanmar Electric Power Enterprise to highway Bus Station and the total distance is 39. The locations passed through the path are as follows:

Myanmar Electric Power Enterprise→ New Modern Yadanabon Market→ Mandalay University→ Chanmyatharzi Airport →Highway Bus Station

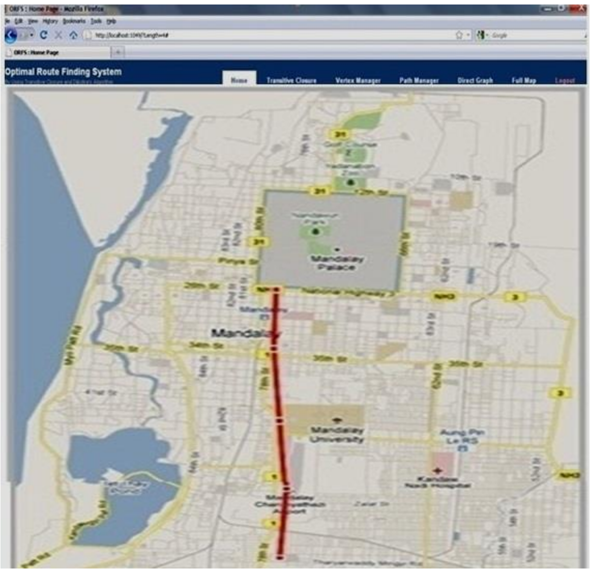

Figure 3: Result of Shortest Path from Myanmar Electric Power Enterprise to Highway Bus Station

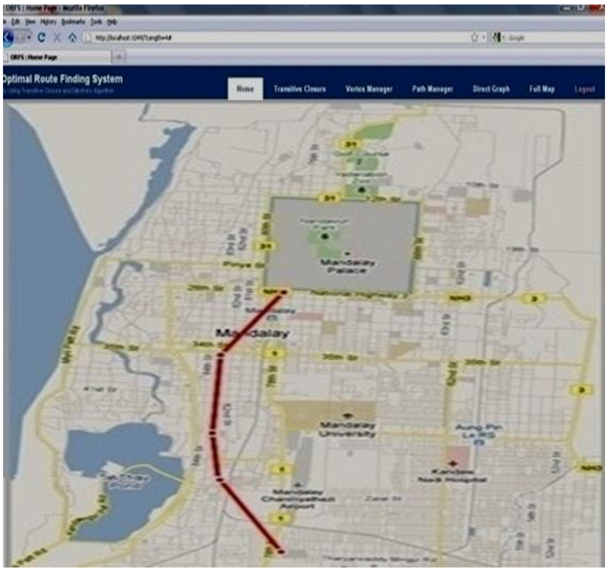

Figure 4: Result of Shortest Path from Highway Bus Station to Myanmar Electric Power Enterprise

The shortest path of Highway Bus Station to Myanmar Electric Power Enterprise is shown in figure 4. In figure 4, the red line is the result of shortest path between Highway Bus Station to Myanmar Electric Power Enterprise and total distance is 43. The locations along that path are as follow:

Highway Bus Station→ Mahamuni Pagoda→ Moyhaung Market→ Manyadana Market→ Myanmar Electric Power Enterprise

In this case, the distance between Myanmar Electric Power Enterprise to Highway Bus Station is shorter than Highway Bus Station to Myanmar Electric Power Enterprise. But each one is the shortest path for each source and destination.

5. Experimental Result

The result of the system is tested on Processor Duel Core 2.4 GHz and RAM 2 GB DDR2. In this testing, the system uses number of nodes as input and produces the performance result as time in second. The processing time for corresponding node is shown in table 1.

Table 1: The Performance of the System

Number of nodes	Time in seconds
10	0.002957
50	0.003504
100	0.290951
1000	4.284540

In table 1, the difference of resulting time between the number of nodes 10 and 50 is 0.000547. In this case, the result of time difference is not significant. The process of time consuming between number of nodes 100 and 1000 is 3.993589. In this case, the result of time difference is significant.

6. Conclusion

There have been developed many shortest path finding systems applying different approaches and methods for different application areas. Such as urban traffic planning, emergency services, transportation services, traveling and tourism service, etc. Researches are needed to focus on two main portions to obtain the efficient shortest path finding system. The *first portion* is concerned with how to transform the different form of map into graph representation. To automatically detect the vertex and edge from map, it will be required to use image processing techniques. The *second portion* is concerned to find the shortest path efficiently and effectively using algorithms.

This system design only emphasizes on second portion to provide the shortest path finding system. Therefore, the transformation step from map to vertex and edge of graph are performed by manually.

The remaining step of finding shortest path is based on two algorithms: Transitive Closure algorithm and Dijkstra's algorithm. The reason is that the Transitive Closure algorithm finds all possible paths from one node to other nodes. So, we can easily check whether a path exits that the user desired source and destination. Dijkstra's algorithm quickly finds the shortest path from chosen source to destination. The nature of shortest path problem is to search optimal solution from start point to the goal point.

A small prototype system is implemented inputting the Mandalay City Map to evaluate how effectively work of our proposed system design. By using this system, it will give the exact path to the users, and can reduce their time and cost in path finding. This system is useful not only for passengers but also for transporter and for the companies that spend a lot on road to travel in unfamiliar area and/or travel to unfamiliar destination in Mandalay.

References

[1] AHO, HOPCROFT, ULLMAN, "The Design and Analysis of Computer algorithm", Third printing, June 1976 by Addison Wesley.

[2] A.M.Kirson, "RF Data Communication considerations in Advanced Driver Information systems". In IEEE Trans on *Vehicular Technology*, 40(1):51--55, February 1991.

[3] A.S TNENBAUM, "Computer Network", Englewood Cliffs, N.J; Prentice-Hall, 1981.

[4] A.TUCKER,"APPLIED COMBINATORICS", State University of New York of Stony Brook.

[5] Emil Kelevedjiev, "Computational Approach For Assement Of Critical Infrastructure In Network System", Institute Of Mathematics And Information.

[6] Farrukh Shehzad, "Evaluation of Shortest Paths in Road Network", National College of Business Administration and Economics, Lahore, Pakistan.

[7] Marcus Jenkins, "Introduction to Route Calculation", NAVTEQ Principal Consultant.

[8] Paul w, Purdom, "Transitive Closure Algorithm", Computer Science Technical Report , 33 July 1968.

[9] P.Biswas, P.K.Mishraand N.CMahanti, "Computational Efficiency of Optimized Shortest Path Algorithms", Department of Applied Mathematics, Birla Institute of Technology, Mesra (India).