Comparing Shortest Path Algorithms for Real Road Network

Moh Moh Wut Hmon, Thin Thin Htway
Computer University, (Hinthada)
mohmohwuthmon@gmail.com

Abstract

This system deals with the problem of finding shortest paths in traversing some location within Myanmar. In particular, it explores the use of three approaches, Dijkstra's shortest path algorithm, Bellman-Ford shortest path algorithm and A algorithm in constructing the minimum spanning tree considering the carriage ways in the road network of Myanmar. The result shows a remarkable reduction in the actual distances as compared with the ordinary routing. Comparing Dijkstra's shortest path algorithm, Bellman-Ford shortest path algorithm and A* algorithm which can produce the minimum calculating time to reach the goal. This result indicates, clearly the importance of these types of algorithm in optimization of network flows. This thesis is implemented by using Microsoft Visual Studio 2008 and Microsoft SQL Server 2005.*

Keywords: Dijkstra's algorithm, Bellman-Ford, A*

1. Introduction

Current development in geographic information system (GIS) technology ensures the network and transportation analyses within a GIS environment are become a common practice in many application areas. But, central problem in network and transportation analyses is the computation of shortest paths between different locations on a network. Sometimes these computations are to be done in real time. For example, traveler can go everywhere with shortest route by using GIS technology. Moreover, when large real road networks are involved, the determination of shortest paths on a large network can be computationally very intensive [11].

The main purpose of this study is to investigate the use of Graph algorithms in optimization given the changing phase of the town brought about physical developments and increase in the number of carriage ways coupled with the surge in the course of fuel and corresponding increase in the cost of transportation [2].

2. Related Work

Qiujin Wu and Joanna Hertley presented public transport travel using K-shortest path algorithms for finding one or several suitable route(s) according to user preferences from one place to another [1].

QU Rong et al. discussed the combination use of knowledge and algorithm for way-finding. In the combined method, the knowledge is used for retrieving the case and isolating the searching area while algorithm is used for finding out the best solution in the isolated areas [6].

Urs-Jakob Ruetschi et al. described network routing by landmarks. It is solved the key problems such as (i) how to attribute landmark information to the network and (ii) how to find an optimal route in this paper. And then Dijkstra's shortest path algorithm is used to find optimal routes [9].

Application of multi-objective Shortest Path and Allocation Analysis of Flood Prevention was discussed in [10]. This paper presented the use of the ArcGIS and mathematical programming, in accordance with the properties of a flood disaster, aiming pragmatically at the balance between the relief of a disaster and the shortest time for conveying the equipments, to construct the optimal model of the equipment's transportation and mobilization of the emergency [14].

3. Shortest Path Algorithm

3.1 Dijkstra's Algorithm

Procedure Dijkstra (G: weighted connected simple graph, with all weights positive){G has vertices a = $v_0, v_1, v_2, \ldots, v_n$ = z and weights w(v_i,v_j) where w(v_i,v_j) = infinity If { v_i,v_j }is not an edge in G}

```
for i:= 1 to n
            L (vi): = infinity
            L (a):  = 0
            S:      = empty set
{the labels are now initialized so that the label of a is 0 and all other labels are infinity, and S is the empty set }
while z not in S
begin       u: = a vertex not in S with L(u) minimal
            S: = S union {u}
            for all vertices v not in S
if L(u)+w(u,v) < L(v) then L(v) := L(u) + w(u,v)
```

{this adds a vertex to S with minimal label and updates the labels of vertices not in S}end {L(z) = length of a shortest path from a to z }[3]

3.2 Bellman-Ford Algorithm

Procedure Bellman-Ford
vector < pair<int,int> > EdgeList;
int graph [128][128];
int n , dist [128];
void bellman-ford (int s)
{//Initialize our solution to the BASE CASE S_0
    for (int i=0 ; i <n ; i++)
        dist [i] = IN_MAX;
    dist[s] = 0;
    for (int k = 0; k < n-1; k++)
{   // n-1 iterations
    //Builds a better solution S_{k+1} from S_k
    for (int j = 0; j < EdgeList.size( ) ; j++)
{   // Try for every edge
    int u = EdgeList[j] .first,
    v = EdgeList[j].second;
    if (dist[u] < INT_MAX && dist[v] > dist[u] + graph [u][v]) //relax
    dist[v] = dist [u] + graph [u][v];
}
}//Now we have the best solution after n-1 iterations
} [5]

3.3 A* Algorithm

Procedure A*
Euclidean distance as estimated distance to the destination. In the searching, the cost of a node V could be calculated as:

f (V)= distance from S to V + estimate of the distance to D.
= d (V) + h (V, D)
= d(V) +sqrt ((x(V) –x(D))^2 + (y(V) – y(D))^2)

Where x (V), x (D) and y (V), y (D) are the coordinates for node V and the destination node D.

The A* Search algorithm:
For each u Є G:
    d[u] = infinity;
    parent[u] = NIL;
End for
    d [s] = 0;
    f (V) = 0;
    H = {s};
    While NotEmpty (H) and targetNotFound:
    u = Extract_Min(H);
    label u as examined;
    for each v adjacent to u:
    p[v] = u;
    f(v) =d[v]+h(v,D);
    DecreaseKey [v,H]; [13]

3.4 Comparing three Algorithms

Dijkstra's algorithm is used as an underlying technique in this thesis because it finds the optimal shortest path between the single source and single destination [12].The Dijkstra's algorithm requires two or more nodes and searches possible routes for the shortest result based on specific weights of the edges that connect them. All nodes are connected to each other by these edges. The edges contain weigh which could be cost, distance, and or time [4].

The Bellman-Ford algorithm, a label correcting algorithm, computes single-source shortest paths in a weighted digraph (where some of the edge weights may be negative). Dijkstra's algorithm solves the same problem with a lower running time, but requires edge weights to be non-negative. If the graph does contain a cycle of negative weights, Bellman-Ford can only detect this; Bellman-Ford cannot find the shortest path that does not repeat any vertex in such a graph [7].

Like all informed search algorithms, it first searches the routes that appear to be most likely to lead towards the goal. What sets A * apart form a greedy best-first search is that it also takes the distance already traveled into account (the g (x) part of the heuristic is the cost from the start, not simply the local cost from the previously expanded node). The algorithm traverses various paths from start to goal. This algorithm is the best algorithm to find shortest path because it search the goal by using bidirectional searches [8].

4. The Proposed System Architecture

This system is comparing three (Dijkstra, Bellman-Ford and A*) shortest path algorithms of real road network within Myanmar Country. This system is dedicated for travelers who can use this system around Myanmar Country in shortest time. In this system, first of all, the user can choose desired town to start and the destination town. The system works with the database to find source and destination point and the adjacency matrix. By using these data, the program generates shortest path in road network map according to the algorithm. In the map, the paths between the towns describe black line and the shortest path will be described in blue color .This system provides the comparison of these algorithms with run time and the distance.

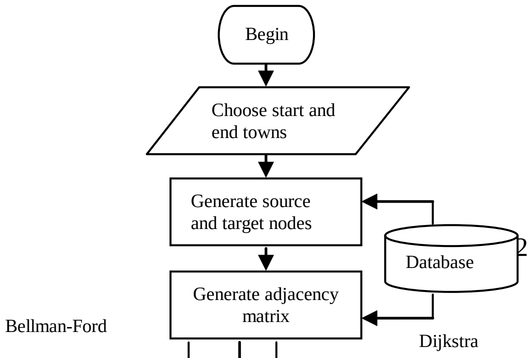

Figure 1: System design diagram

5. Case Study

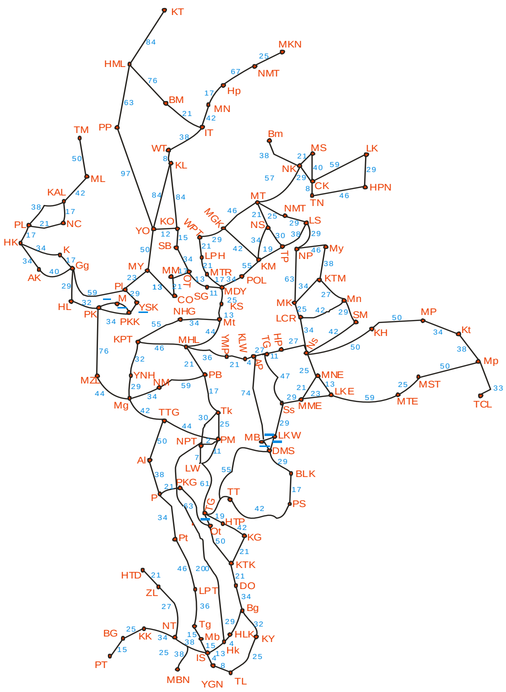

Figure 2: Case study of real road network within Myanmar country

Figure 2 illustrates the case study of real road network within Myanmar country. In this system there are 126 towns and cities in Myanmar country that can travel by car and described the distances with miles.

6. Experimental Result

In this form, the user can choose desired town to start and desired town to go in list boxes. Then the user can click Find button to calculate the shortest route between desired start and end town. The user can click Exit Button to exit the system. If the user wants to choose another state or division, Back Button can be used.

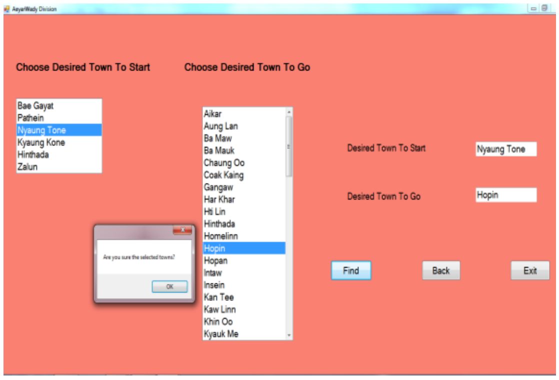

Figure 3: Input form

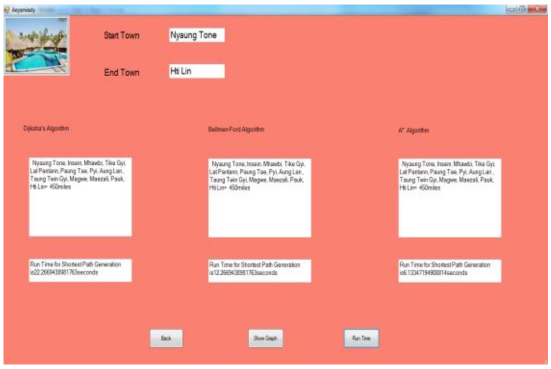

Figure 4: Shortest route using Dijkstra's, Bellman-Ford and A* algorithm

This form is shortest route using Dijkstra's algorithm, Bellmen-Ford algorithm and A* algorithm. In this form, there are three buttons; Back Button can choose another towns, Run Time Button shows the run time for corresponding algorithm and Show Graph Button load Shortest Path Road Network Graph form as shown in Figure 5.

In this form, the normal road is displayed in green color and the shortest path displays in blue color lines on the road network graph.

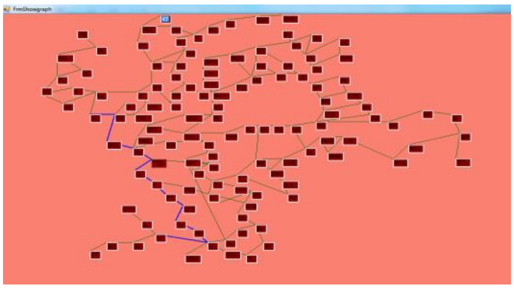

Figure 5: Shortest path road network graph

7. Runtime Result

The runtime performance of the different situations such as the source town, the destination town, the total miles, and the runtime for three algorithms are shown in Table 7.1. The optimal result is the same by using (Dijkstra's algorithm, Bellmen-For algorithm and A* algorithm). The performance of the runtime is different. In this table, A* algorithm is the best time to calculate the optimal shortest path because this algorithm search optimal route by using bi-directional search. Bellmen-Ford algorithm develop dijkstra's algorithm by using Sk times. This algorithm enhances the dijkstra's algorithm by using negative value for the cost. Dijkstra's algorithm is also best shortest path algorithm for positive value for the cost.

Table 7.1 Runtime for Various Situations

Source Town	Destination Town	Runtime (s) Dijkstra	Runtime (s) Bellmen-Ford	Runtime (s) A*	Total (miles)
Tamu	Mandalay	17.325	12.566	6.357	355
Nay Pyi Taw	Kaw Linn	16.991	11.986	5.772	274
In Taw	Lae Waii	17.834	13.045	6.768	357
Kan	Kyauk Me	16.217	12.704	5.232	263
Ohn Taw	Pyi	17.476	12.666	6.957	321
Hinthada	Yei Oo	19.411	14.333	9.205	531
Zalun	Yangon	15.009	10.873	5.055	69
Pathein	Meikhtilar	18.881	13.448	8.452	401
Hti Lin	Pauk Kaung	16.004	11.662	5.877	303

8. Conclusion

The driving guide system "Comparing Shortest Path Algorithms for Real Road Network" for cities and towns in Myanmar country can be established from that a fore mentioned that Dijkstra's algorithm, Bellman-Ford algorithm and A* algorithm are useful graph theoretical mechanism for optimization processes of network connectivity. Results have exposed the versatility of this theoretical tool in carrying out minimization process involving traveling, construction and for itinerancy in conveyance of goods and services in different locations of a town based on the existing road network.

8. References

[1] T. Alan "Applied Combinations "State University of New York at Stony Brook, ISBN 0-471-04766-X, 1980 by John Wiley & Sons, Inc.

[2] L.Dai "Fast Shortest Path Algorithm for Road Network and Implementation", Carleton University School of Computer Science CPMP 4905 Honours Project Fall Term, 2005.

[3] R.Diestal, "Graph Theory" 1997 by Springer-Verlag New York.

[4] E.W.Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematic.

[5] R.Diestal, "Graph Theory" 1997 by Springer-Verlag New York.

[6] A. Ibrahim "Graph Algorithm and Shortest Path Problems: A Case of Dijkstra's Algorithm and the Dual Carriage Ways in Sokoto Metropolis".

[7] G.A.Klunder, H.N Post, "The Shortest Path Problem on Large Scale Real Road Networks".

[8] W.Pijls, "Heuristic estimates in shortest paths", Statistical Neerlandica.

[9] I.Pohl,"Bi-Directional Search, Machine Intelligence".

[10] M . Scheu "Singel-Source Shortest Paths" (One-to-all shortest path).

[11] O. Sharma, D. Mioc, F. Anton and G.Dharmaraj "Traveling Salesperson Approximation Algorithm for Real Road Networks".

[12] D.Srini and L.Eric, "Graph Theory", Mathematics for Computer Science March 1, 2005.

[13] K.N.Swe, "Efficient Tour Information Based On Shortest Path Algorithm", October, 2002.

[14] P.Wim, P.Henk "A new bidirectional algorithm for shortest paths", Econometric Institute Report.