A Shortest Route Problem by Using Dijkstra's Algorithm and Its Applications

Yi Yi Win
Faculty of Computer Science
University of Computer Studies (Meiktila)
yiyipku941@gmail.com

Tin Tin Thein
Faculty of Computer Science
University of Computer Studies (Pakokku)
tttheinymb@gmail.com

Hnin Su Hlaing
Faculty of Computer Science
University of Computer Studies (Loikaw)
hninsuhlaing07@gmail.com

Abstract

The shortest path algorithm (SPA) is one of the most fundamental and important in combinational problems. SPA, a problem in graph theory, has commonly applied in communications, transportation, Artificial Intelligence, Geographic Information System and electronics problems. The problem of finding a path between two vertices (or nodes) in a graph is a key role of SPA tricky to minimize the weights of constituent edges. Thus we would like to implement and highlight the advantages of a shortest path algorithm, Dijkstra's Algorithm (DA), which has been widely used in engineering calculation such as in Google maps. In this paper, we approve the constancy of this algorithm, also known as label algorithm, using only on positive weights for travelling tracking process in Myanmar cities. The criterion values are referred from Google map data. This paper presents not only the implementation of applicable tracking system but also the minimum cost of this applied algorithm with graph data structure including finite set of vertices. Finally, we also present the review of implementation in these ages as stated by complexity and where we can apply this algorithm as a future work.

Keywords: Applications, Dijkstra's Algorithm, Directed Graph, Shortest Route

1. INTRODUCTION

At first, we would like to explain about what is shortest path in graph theory. In graph theory, the problem of finding a path between two vertices (nodes) is the shortest path problem such as the sum of the weights of its constituent edges is minimized [2]. Graph is a mathematical representation of a network including the relationship among a finite number of points (nodes) connected by lines (edges) [3]. This shortest path problem can be modeled in a graph, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment. Thus, graph theory is a key role for various modeling problems in real world such as communications, transportation, Artificial Intelligence, Geographic Information System, electronics problems, travelling problems and various computer applications [4].

Our country, Myanmar, has varieties of interested places concerned with culture, religion, villages, ecotourism, historical buildings, temples and other pleasure natural residences. They who want to visit or travel from one place to another needs to recognize the effective and efficient ways. Thus we would like to put on the points in graph theory as cities in Myanmar and the edges may be roads as the connected links using weight value assigned with distance between two points. The key tricky of this work is for finding the shortest path through the network or the best way in travelling processes.

This paper will provide the shortest paths from one place to another by using Dijkstra‘s Algorithm in graph theory [1] [5] [6]. Firstly, we will express the background of study in section 1.1. In section 2, the objectives of this paper is expressed and the methodology including the case of shortest path problem, result and discussion, how Dijkstra‘s Algorithm do in problem and analytical solution is also explained with a case study in section 3. Thus we can see the optimal paths for travelling to the designated city with shortest distance and minimum time as a case study. Finally, we organize how to achieve the implementation result and how we can go to future plan with these groups of theories such as other shortest path problems, graph theory section 4.

1.1. Background of Study

In paper [1], the author apply Dijkstra's algorithm to provide the market problem for a tourism company in Bali, one of many small island in Indonesia. There were top ten tourist destinations to be implemented in this paper. They use Java language for more reliable and easier to maintain manage memory automatically with garbage collection discovery how the JVM provides portability and security.

The evaluation and discussion of the general aspect of Dijkstra's Algorithm is also presented for demonstrating the functionality of this algorithm to minimize the cost of routing the computer network [6]. In this paper, there are four variables such as set of visited nodes, queue, graph and weight to be implemented with Dijkstra's Algorithm. In addition, the acceptable in terms of overall performance is the time complexity in this consequence by implementing the IP routings as nodes and routing protocols as links. However, it has till some major drawbacks about this application such as using negative edges, leading to acyclic graphs, not obtaining the right shortest path and highly centralized. But they said what these drawbacks may be taken care by some other shortest path algorithm like Bellman-Ford and FloydWarshall algorithms.

The comparison of time and space complexity about three basic algorithms for finding shortest paths, Dijkstra's algorithm, Bellman-Ford algorithm and FloydWarshall algorithm, is theoretically presented in paper [5]. They conclude that the Dijkstra's algorithm applies spares graph and is always optimized in actual application, like heap optimization. The time complexity cuts to n*logn [7]. The time complexity of Dijkstra's algorithm is presented with a binary heap by using non-negative weight in a directed graph [8]. Accordingly, we are interested to compile an optimized shortest path algorithm, Dijkstra's algorithm, to provide a tourism organization to be cost effective.

2. OBJECTIVES OF EMPLOYMENT

The following list gives the objectives of this research paper:

- To understand how to determine and identify the concepts of the shortest path problem in a same graph.
- To determine the representation of graphs in computer in order to solve the shortest path problem, as well as to understand the different basic terms of a graph.
- To explain the general concepts and the implementations of Dijkstra's Algorithm.
- To safe the transposition delay and cost.

3. METHODOLOGY

3.1. The Shortest Path Problem

This paper is to find the shortest path from a given city, called home, to another given city called destination by using Dijkstra's algorithm. Why we use this algorithm for implementation is that this is the most efficient algorithm compared with other types of shortest path algorithms in. [1, 7] The length of a path is assumed to be equal to the sum of the lengths of links between consecutive cities on the path. With no loss of generality we assume that city 1 is the home city and city n is the destination. [9] Some improvements on Dijkstra's algorithm are also done in terms of efficient implementation. [3]

At first, in this algorithm implementation, the Google maps shown in Figure 1 is used to set all vertices distances as shown in Figure 2 and source vertex in a min-priority queue in form (distance, times), as the comparison in the min-priority queue will be according to vertices distances. The minimum distance from the priority queue is popped; at first source may be the popped vertex. The distances of the connected vertices to this popped vertex is then updated in case of "current vertex distance + times < next vertex criteria" and then push the vertex with the new distance to the priority queue. After that, if the popped vertex is visited before, just continue without using it. Finally the same algorithm is applied again until the priority queue is empty. [8]

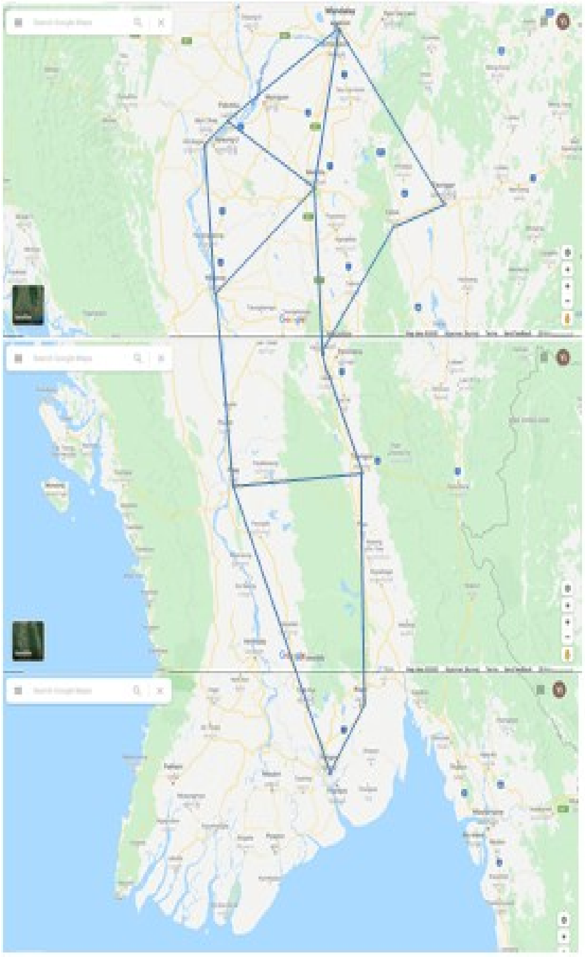

Figure 1. Total traffic ways between Yangon and Mandalay from Google Maps

3.2. Result and Discussion

The effectiveness of the application area is governed for some cities in our country. Now we like to express how we get the shortest path between the most famous two cities in Myanmar, Yangon as a source and Mandalay as a destination. The graph data structure of possible traffic ways between Yangon and Mandalay according to Google map [9] is shown in Figure 2.

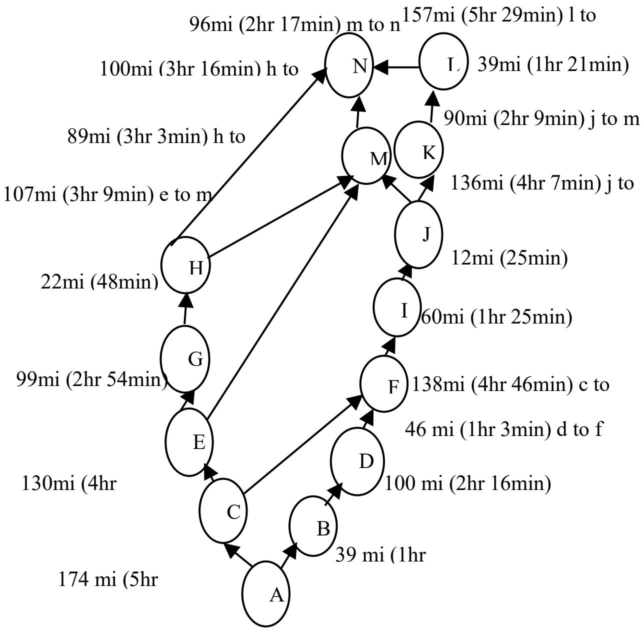

Figure 2. Total traffic ways between Yangon and Mandalay

In this paper, we present not only the implementation of Dijkstra algorithm as a National City traffic advisory practice which can satisfy the passengers' different demand of travel but also providing some optimal decision. The identifications of implemented figures for graphical representation, such as city names and related nodes are shown in Table 1.

Table1. Names of location and Nodes Identification

LOCATION	NODES
Yangon	A
Bago	B
Pyay	C
Phyu	D
Magway	E
Taungoo	F
Bagan	G
Pakokku	H
Lewe	I
Naypyitaw	J
Kalaw	K
Taungyi	L
Meiktila	M
Mandalay	N

3.3. Calculation of Dijkstra's algorithm

Edsger Dijkstra invented an algorithm for finding the shortest path through city to another. Thus, we use the following simple set of instructions along with the concept of DA in section 3.1.

Step 1: Label the start vertex as 0.
Step 2: Box this number (permanent label).
Step 3: Label each vertex that is connected to the start vertex with its distance (temporary label).
Step 4: Box the smallest number.
Step 5: From this vertex, consider the distance to each connected vertex.
Step 6: If a distance is less than a distance already at this vertex, cross out this distance and write in the new distance. If there was no distance at the vertex, write down the new distance.
Step 7: Repeat from step 4 until the destination vertex is boxed.

3.4. Analytical Solution to the Problem

As in the above section 3.3, the evaluation of shortest path between two famous cities according to the first three steps of DA as shown in Figure 3.
Step 1: Label A as 0
Step2: Box this number
Step3: Label value of 39mi (1hr 26min) at B, 174 mi (5hr 31min) at C

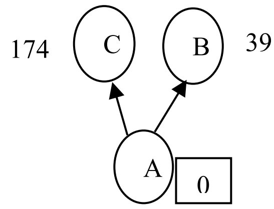

Figure 3. Graph representation of step1-3 for starting form source

In Figure 4, the node "B", Bago is chosen as a shortest path for starting from source according to the comparison of distance and time values with node "C", Pyay. According to the needed of updating and pushing the vertex with the new distance to the priority queue, the following step 4 and step 5 are done as follow:

Step 4: Box the 39 mi (1hr 26min) at B.
Step 5:From B, the connected vertices is D. The distance at these vertices are 139 mi (3hr 42min) at D (39+100) mi and (1hr 26min +2hr 16min).

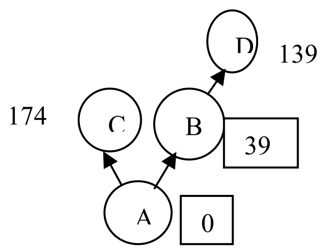

Figure 4. Graph representation of step 4-5 for choosing "B", Bago

After repetitive done on step 4 and step 5, we can get the new node "D", Phyu (139mi -3hr 42min) as a single one to be the shortest path as shown in Figure 5. An additional node "F", Taungoo (138mi-4hr 46min) has two input way from "C", Pyay and "D", Phyu to get the shortest one as said by the predefined events and values of each criterion as shown in Figure 6.

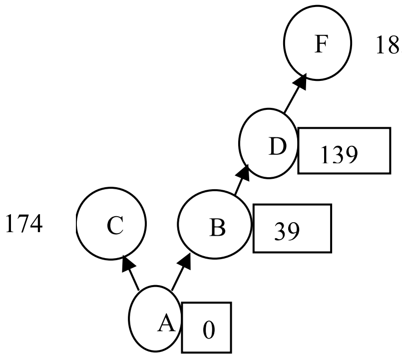

Figure 5. Graph representation of step 4-5 for getting a single one "D", Phyu

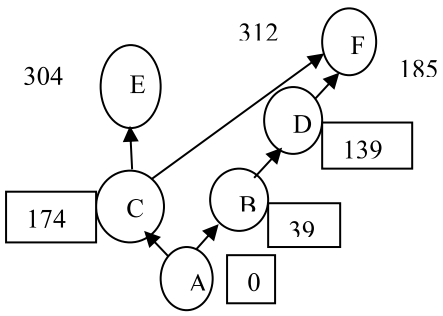

Figure 6. Graph representation of step 4-5 for two input miles to "F", Taungoo

As the comparison result of two input miles to "F", we choose "I", Lewe as a next city by removing the way from "C", Pyay because of larger miles as in Figure 7.

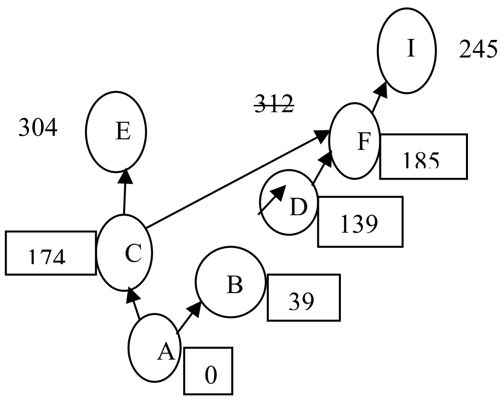

Figure 7. Graph representation of step 4-5 for choosing "I", Lewe

The following three steps are used to trace for getting the reasonable result of chosen city:

Step 6: As the distance at F is 185mi (4hr 45min), lower than the 312mi (10hr 17min) currently at F, cross out the 312 mi.

Step 4: Box the smallest number, which is the 185 mi (4hr 45 min) at F.

Step 5: From F, the connected vertices is I. The distance at these vertices is 245 mi (6hr 10 min) at I (185+60) mi and (4hr 45min +1hr 25min).

As a consequence of DA, the system gives the next city "J", Naypyitaw as a single way of shortest path and "M", Meikhtila as a shortest way compared with "K", Kalaw. When we go to "N", Mandalay through "M", Meikhtila followed by "J", Naypyitaw, we also use step 4 and step 5 only.

Although we got the distinction city, we still have to run other smaller ways from other cities. Thus another ways of "H", Pakokku to"M", Mandalay and "L", Taungyi to"M", Mandalay are also evaluated to compare each other. As a result, we are encouraged to go with the shortest path of "M", Meikhtila to "N", Mandalay consistent with the step 7 as shown in Figure 8.

Step 7: The final vertex, in this case N (goal), is not boxed. The boxed number at N is the shortest distance. The route corresponding to this distance of 443mi (11hr 1min) ABDFIJMN, but this is not immediately obvious from the network.

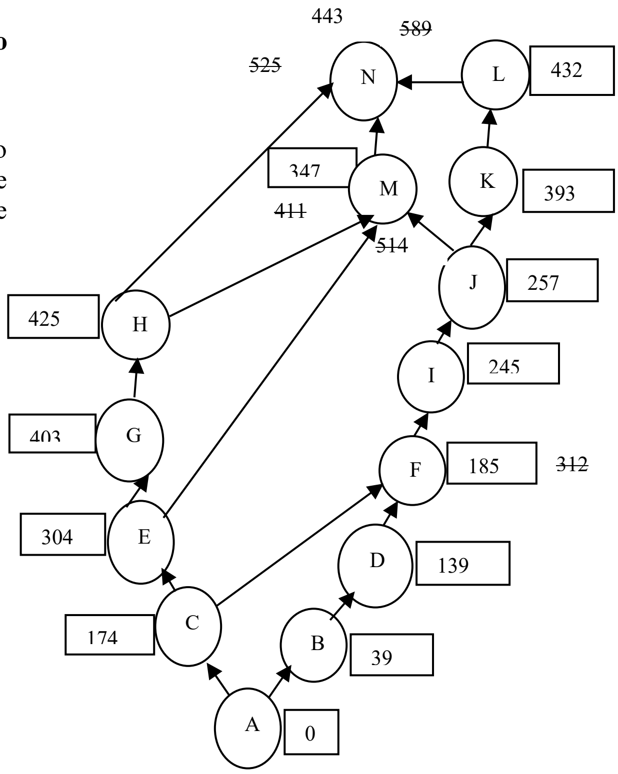

Figure 8. Graph presentation of shortest path evaluated result from Yangon to Mandalay

In conclude, the shortest path of Yangon to Mandalay is as the following pattern with the smallest distance of 443 mi (11hr 1min):

"A" – "B" – "D" – "F" – "I" – "J" – "M" – "N"

The traversed table of finding shortest path from Yangon to Mandalay and the referenced Google map are shown in Table 2 and Figure 8 for more thoughtful how to evaluate the applied algorithm, respectively. According to Table 2 and Figure 7, we can realize that the evaluation of the shortest distance between Yangon and Mandalay, consistent with the route of Google map, is 443 miles, and taking time is 11hr 1min. As a result, we can prove the implemented shortest path algorithm, Dijktra's algorithm, is a good one to be optimized in shortest path problem. We can also say that the space complexity of this algorithm is O (E); E is number of edges because of traversing 14 edges in this evaluation.

Table 2. Traversed Table

Track	Source	No: of ways	Selected Destination
1	**A**	C (174mi, 5hr31min) B(39mi, 1hr26min)	B
2	B	D(139mi, 3hr42min)	D
3	D	F(185mi, 4hr45min)	F
4	F	C(174mi, 5hr31min)	C
5	C	E(304mi, 9hr49min) F(312mi, 10hr17mi)	E
6	E	I(245mi, 6hr 10min)	I
7	I	J(257mi, 6hr35min)	J
8	J	M(347mi, 8hr44min) K(393mi,10hr42mi)	M
9	K	L(432mi, 12hr3min)	L
10	M	E(304mi, 9hr49min)	E
11	E	G(403mi, 12hr43min) M(411mi,9hr58min)	M
12	M	N(443mi, 11hr1min)	**N**
13	G	H(425mi,13hr31mi)	H
14	H	M(514mi, 16hr24min) N(525mi, 16hr47min)	M

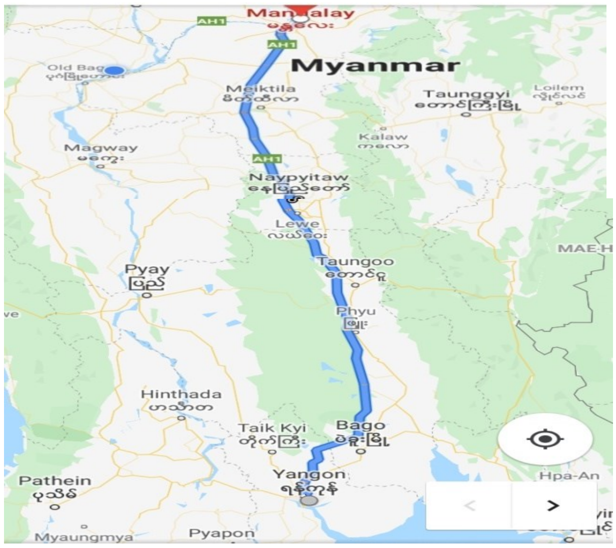

Figure 9. Referenced Google Map

4. CONCLUSION

This paper is proposed to implement the efficient shortest path algorithm, Dijkstra's algorithm, by using graph data structure with finite set of data about transportation for providing tourism organizations in Myanmar. In current, we just implement and give a chance for learning how to evaluate for finding shortest path and where they are used such as Google map, Telephone Networks. In our future, we will be going to learn and develop about maximum bandwidth path in Internet connection to reduce transmission delay and cost. In addition, we will also learn and express the different ways of shortest path algorithms.

References

[1]A Fitriansyah1, a, N W Parwati1, b, D R Wardhani1, c and N Kustian1,d
1Program, "Dijkstra's Algorithm to Find Shortest Path of Tourist Destination in Bali" PGRI, Jl. Raya Tengah No.80, Kel. Gedong, Kec. Pasar Rebo, Jakarta Timur, 13760, Indonesia.
[2]E.Kreyszig, Advanced Engineering Mathematics, 10th edition.John Wiley & Sons, Inc., 2011.
[3]E.Kreyszig, Advanced Engineering Mathematics, 10th edition. John Wiley & Sons, Inc., 2011.
(2) MeysamEffati, ―Analysis of Uncertainty Considerations in Path Finding Applications‖.5th SASTech 2011.
[4]Goldberg, A. V. (n.d.). Point - to - Point Shortest Path Algorithms with Pre-processing. Silicon Valley: Microsoft Research.
[5]Hebei, "The Comparison of Three Algorithms in Shortest Path Issue Xiao Zhu WANG" China First International Conference on Advanced Algorithms and

Control Engineering IOP Publishing IOP Conf. Series: Journal of Physics: Conf. Series 1087 (2018)

[6]Jogamohan Medak1 Partha Pratim Gogoi2 “Review and Analysis of Single-Source Shortest Path Problem Using Dijkstra's Algorithm”IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 20, Issue 2, Ver. II (Mar. - Apr. 2018), PP 10-15.

[7]Klein, P., Rao, S., Rauch, M., & Subramanian, S. (1994). “Faster shortest-path algorithms for planar graphs.” Twenty-Sixth ACM Symposium on Theory of Computing (Vol.55, pp.27-37). ACM

[8]Michael Barbehenn “A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices” https://www.researchgate.net/publication/3043930Article in IEEE Transactions on Computers · March1998

[9] https://www.google.com. mm. Maps