Dijkstra's Algorithm for Effective Travelling to the Most Famous Destinations in Myanmar

San San Maw[#], Khin Saw Lin[*], Lin Lin Naing[**]

[#]*Department of Engineering Mathematics, Mandalay Technological University, Mandalay, Myanmar.*

[*]*Department of Student Affairs, University of Information Technology, Yangon, Myanmar.*

[**]*Faculty of Computing, University of Computer Studies, Hinthada, Ayeyarwady Division, Myanmar.*

Abstract—*In this paper,the well-known Dijkstra's Algorithm is applied to find the optimal paths for travelling to the most famous destinations in Myanmar. It is aSingle-Source Shortest Path (SSSP) iterative algorithm and the basic idea is searching a graph by findingpath, starting at a point, and exploring adjacent nodes from there until the destination node is reached. Generally, the goal is to obtain the shortest path from current city to the other destinations. This algorithm can bereviewed with values which come from weights on edges according to actual situations on the road, such as costs, distances, and travelling time. The key issue to be addressed in this work is to find the shortest paths from one source point (city/place) to others by comparing the weighted values* (*i.e., distances and time*) *between any two different paths with their edge lengths (roads) that assigned by actual values for saving cost and time for effective travelling. The values of distances and time between any two destinations are taken from Myanmar distance calculator website and compared the accuracy of the results using those values in Google Map.*

Keywords—*Path Finding, Weighted Graph, Routing, Shortest Paths, Dijkstra's Algorithm.*

I. INTRODUCTION

In many fields of applications, graphs theory [1]plays vital role for various modelling problems in real world such as travelling, transportation, traffic control, communications, and various computer applications and so on [2]. Graph is a mathematical representation of a network and it describes the relationship among a finite numberof points (nodes)connected bylines (edges) [3]. Depending on the problems we considered, the points may be cities or any other destinations and the lines may be roads or other connected links. The main problem in this work is to find the shortest path through the network or the best way in travelling processes. Myanmar, Golden Land, is full of interesting places and the one who wants to visit or travel from one place to another needs to know the effective and efficient ways. This paper will provide the shortest paths from one place to another by using Dijkstra's Algorithmingraph theory [4], [5]. Firstly, the general and formal descriptions of the algorithm are presented. Then the steps of the algorithm are explained. Finally the detail implementation of the algorithm is illustrated to find the optimal paths for travelling to the most famous destinations in Myanmar, with shortest distances and minimum time, as a case study.

II. RESOURCES AND METHOD

A graph is a series of points and lines that can beused to represent the connections that exist in varioussettings. The lines are called edges and the points are called vertices(or 'nodes'), with each edge joining a pair of vertices.Although edges are often drawn as straight lines, theydon't have to be.Nowadays travelling has becomes an important aspect for everyone and the one who visits to some cities or interesting places around the countryby car has to drive to *n* cities. Because of all the cities are linked by roads to each other, any city can be visited from any other city directly. In this case, it is very important to find the *optimal route*, that is, the route with the shortest total mileage or lowest costor minimal travel time for overall trip. In this paper we consider one-to-all shortest path problem for determining the shortest path from a start city to all other famous destinations in Myanmar.

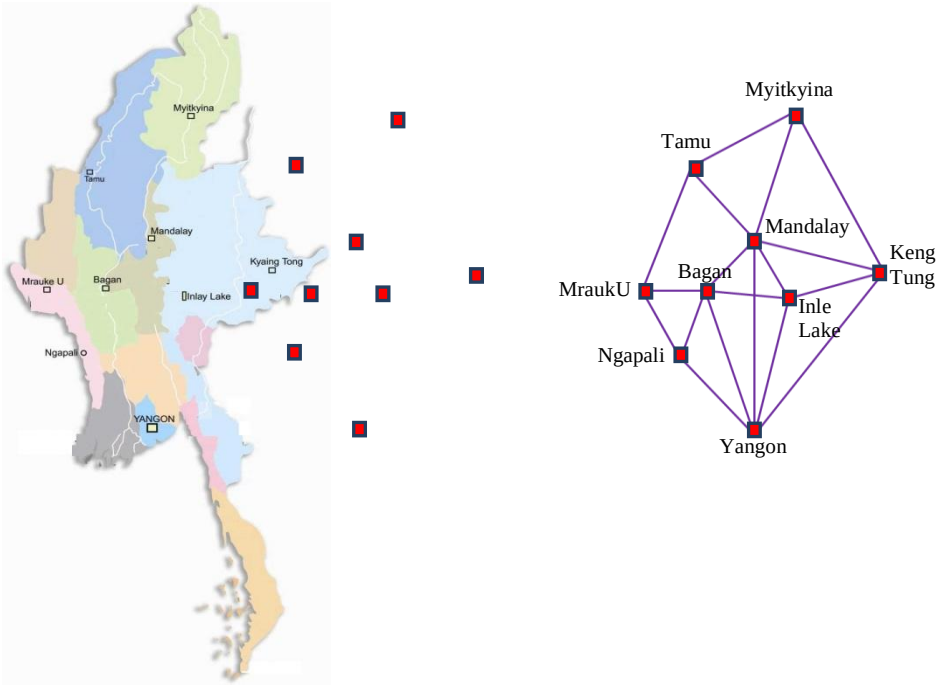

Fig. 1 Representation for the famous destinations in Myanmar by a connected graph [6]

Imagine that each node is a famous place and each edge is an existing road between two places. This means that anyone can drive from Myitkyina to Mandalay directly. However one cannot drive from Myitkyina to Mrauk U directly, as there is no direct road between those places. In this case, not every road is equal. Some of them are longer, some are not in good shape, and the capacity of some roads are lower than others. So you need to choose the optimal path to save your time and money. We may represent the time, money or distance as weighted values that we assign to the roads.

A. Problem Definition

The shortest path problem is a problem for finding the shortest path or route from a starting point to a final destination. The problem of computing shortest paths in a weighted graph frequently arises in practice. The lengths of the edges are often called weights, and the weights are normally used for calculating the shortest path from one point to another point [7]. When tourists came to visit in our country, they may not be familiar with all interesting destinations of our country. In this situation, an effective travelling plan is crucial for achieving their satisfactions during tourism.For a given source node in the graph, the algorithm finds the shortest path between that node and every other.Dijkstra's algorithm can be used to find the shortest path between one city and all other cities [8]. The input to the shortest-path algorithm is a graph G that consists of a set of vertices V, a set of edges $E$and theweight set D specifying explicit weights d_{ij} for the edges $(i, j) \in E$ [9]. The only restriction is that the weights are required to benonnegative. The graph is defined as $G = (V, E, D)$. The edges can be directedor undirected[10]. Although basic ideas and algorithms will be explained and illustrated by small graphs in this paper, the real-life problems may often involve many thousands or even millions of vertices and edges.

B. Dijkstra's Algorithm

Among several algorithms that have been proposed for the shortest path between vertex pair, perhaps the most efficient one is an algorithm due to Dijkstra[8]. His algorithm is used to efficiently find a shortest path from a given city s, called a starting node or initial node, to other cities in a network (one-to-all shortest path problem) with nonnegative weights. In this paper, it finds a route containing the shortest distance and time (from one city to another) among two or more places in Myanmar. The descriptions of algorithm are as follows [4]:

1) ***General Description:*** The general description of the algorithm is:
- to assign some initial values for the distances from node s and to every other node in the network,
- to operate in steps, where the algorithm improves the distance values at each step.
- to determine the shortest distance from node *s* to another node in each step.

2) ***Formal Description:*** The algorithm characterizes each node by its distance value and status label:
- Distance value of a node is a scalar representing an estimate of its distance from node *s*.

- Status label is an attribute specifying whether the distance value of a node is equal to the shortest distance to node s or not.

The status label of a node is *permanent* if its distance value is equal to the shortest distance from node *s* and otherwise, the status label of a node is *temporary*. The algorithm maintains and step-by-step updates the above states of the nodes by designing one node as a current node in each step.

3) ***Notation and Steps of the Algorithm:*** We use the symbold_l to denote the distance value of a node$l$. $p$and$t$ denotes the status labels of a node, where p stand for permanent and t stands for temporary.d_{ij} is the distance of traversing link(i,j) as given by the problem. The state of a node l is the ordered pair of its distance value d_l and its status label.

The steps of the algorithm are stated as follows:

Step 1. Initialization

- The starting point (node s) is (0, p) that label as Permanent and its distance value is 0.
- Every point (node) is (∞, t) that label as Temporary and its distance value is ∞.
- Designate the node s as the current node.

Step 2.Designation of update values of distance and current node

Let d_i be the distance value for the index of the current node. Search the nodes with temporary labels (∞, t) that can reach from the current node d_i by adjacent nodes(i,j). Update the distance values of these nodes.

- $newd_j = \min\{d_j, d_i + d_{ij}\}$

whered_{ij} is the distance values between i^{th}vertex and j^{th}vertex.

- Choose the smallest distance value d_j and use it as a current node.

Step 3.Termination

- If the set of all temporary nodesthat can be reached from starting node s is empty- we are done.
- If we cannot reach any temporary labelled node from the current node, then all the temporary labels become permanent - we are done.
- Otherwise, go to Step 2.

The flowchart illustrating this shortest path algorithm is as follows [11]:

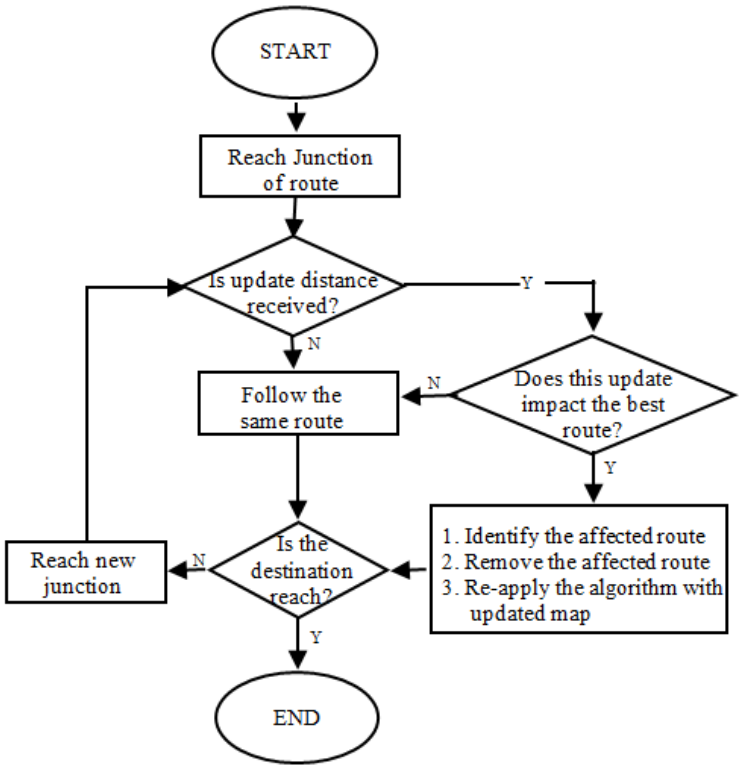

Fig. 2 Flowchart for illustrating the best route update during the journey

TABLE I

DIJKSTRA'S ALGORITHM FOR SHORTEST PATHS [1]

$[G = (V, E), V = \{1, \dots, n\}, l_{ij} > 0$ for all (i, j), in E]

INPUT: Number of vertices n, edges (i, j), and lengths l_{ij}

OUTPUT: Lengths L_j of shortest paths $1 \longrightarrow j, \ \ j = 2, \dots, n$

1. *Initial step*

Vertex 1 gets PL: $L_1 = 0$

Vertex $j(= 2, \dots, n)$ gets TL: $\tilde{L_j} = l_{ij}$ $(= \infty$ if there is no edge $(1, j)$ in G)

Set PL={1}, TL={2, 3, … , n}.

2. *Fixing a permanent label*

Find a k in TL for which $\tilde{L}_k$ is minimum, set $L_k = \tilde{L}_k$. Take the smallest k if there are several. Delete k from TL and include it in PL.

OUTPUT $L_2, \dots, L_n$. Stop

Else continue

3. *Updating temporary labels*

For all j in TL, set $\tilde{L}_j = min_k\{\tilde{L}_j, L_k + l_{kj}\}$

Go to Step 2.

END DIJKSTRA

III. RESULTS AND DISCUSSION

In this paper, the application of the Dijkstra's Single-Source Shortest Path (SSSP) algorithm has been usedto compute the shortest pathsfrom one place to another [10]. The travelling process starts from Yangon to other eight cities (places). Everyone who wants to travelneeds to optimize the route for saving time and money. Although the detail calculation for optimal result of the shortest distance has been focused, the other factor such astravelling timehas also been considered in this paper. Firstly, the weights on the links are referred as distance (miles) for corresponding route.

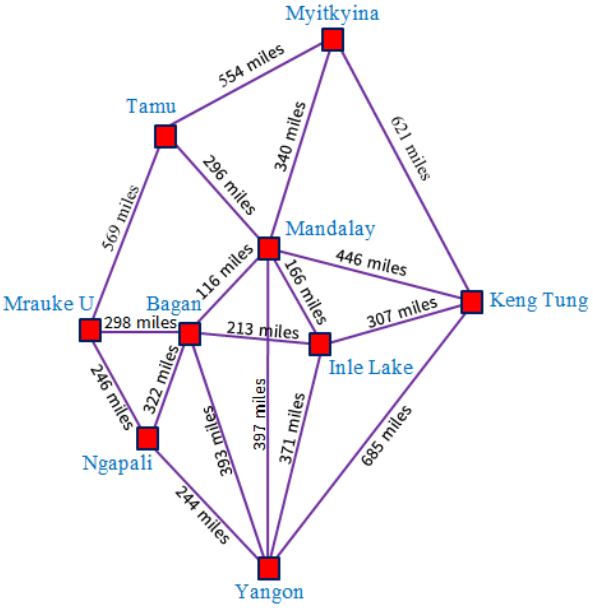

Fig. 3 Representation of cities by weighted distance (miles) connected graph

A. Implementation of the Algorithm

The implementation of the Dijkstra's shortest path algorithm is illustrated to find the optimal path from Yangon (YG)to other cities as follows:

Step 1.Yangon (YG) is designed as the current city and the state of the YG is (0,p). Every other city has state (∞, t) as shown in Fig. 4.

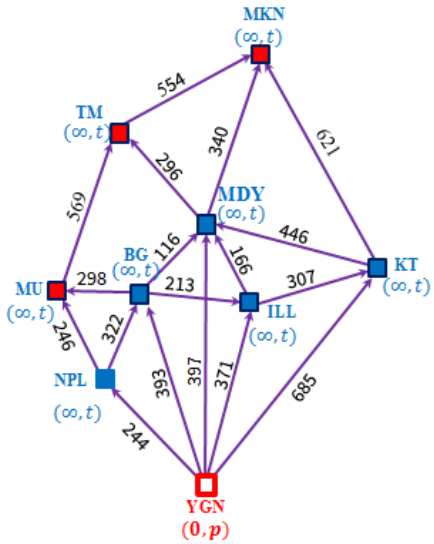

Fig.4 The states of current city (YGN) and its neighbouring cities

Step 2.The cities named Ngapali(NPL), Bagan(BG), Mandalay (MDY), Inle Lake (ILL) and Keng Tung (KT) can be reached from the current city YG. Update distance values for these cities;

$$d_{NPL} = min\{\infty, 0 + 244\} = 244$$
$$d_{BG} = min\{\infty, 0 + 393\} = 393$$
$$d_{MDY} = min\{\infty, 0 + 397\} = 397$$
$$d_{ILL} = min\{\infty, 0 + 371\} = 371$$
$$d_{KT} = min\{\infty, 0 + 685\} = 685$$

Now among the cities NPL, BG, MDY,ILL, and KT, the Ngapali (NPL) has the minimum distance value.

- The state label at NPL changes to permanent so its state is (244, p) while the states of BG, MDY, ILL and KT remain temporary.
- NPL becomes the current cityas shown in Fig. 5.

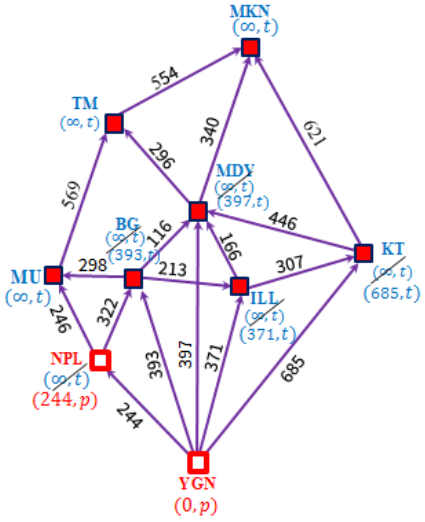

Fig. 5 Changing Ngapali (NPL) as permanent and current city

Step 3.Graph the end of step 2 as shown in Fig. 6.

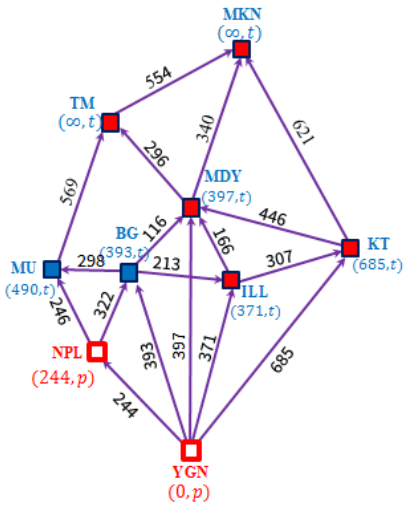

Fig. 6Current city Ngapali (NPL) and its neighbouring temporary cities

We are not done, not all cities have been reached from YG, so we perform another iteration step (back to step 2).

Another implementation of step 2

- Cities Mrauk U (MU) and Bagan (BG) can be reached from the current city NPL.
- Update distance value for these cities.

$$d_{MU} = min\{\infty, 244 + 246\} = 490$$

$$d_{BG} = min\{393, 244 + 322\} = 393$$

Now between the cities MU and BG, the distance of city BG is shorter than MU.

- The state label of BG changes to permanent, while the state of MU remains temporary.
- BG becomes the current city as shown in Fig. 7.

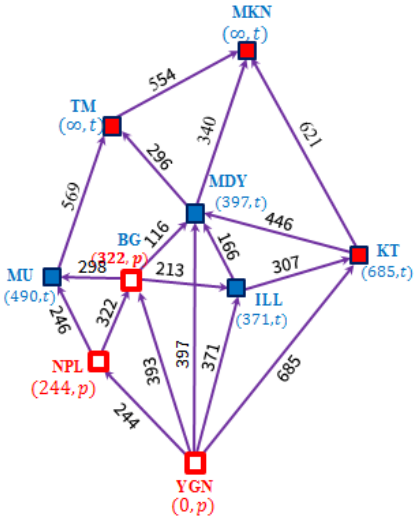

Fig. 7 ShowingBagan (BG)as current city and its neighbouring cities

Another Step 2

- Cities Mrauk U (MU), Mandalay (MDY) and Inle Lake(ILL) can be reached from the current city BG.
- Update distance value for these cities.

$$d_{MU} = min\{490, 393 + 298\} = 490$$

$$d_{MDY} = min\{397, 393 + 116\} = 397$$

$$d_{ILL} = min\{371, 393 + 213\} = 371.$$

- Now among the cities MU, MDY, and ILL, the distance of Inle Lake (ILL) is the shortest one.
- The state label of ILL changes to permanent, while the state of MU and MDY remain temporary shown inFig. 8.
- ILL becomes the current place.

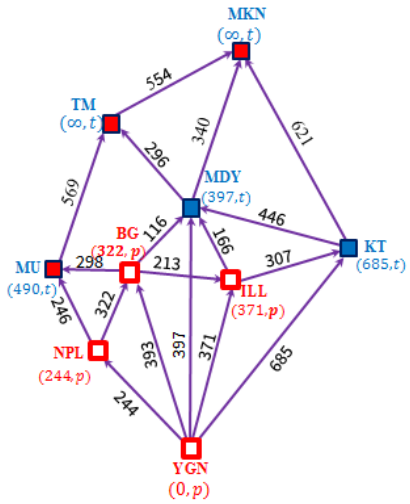

Fig. 8 Showing Inle Lake (ILL) as current place

Similar calculation can be done repeatedly for step 2 again and again until all cities have been changed to permanents. The final graph showing the shortest distance from start city (YGN) to other cities can be seen as follows:

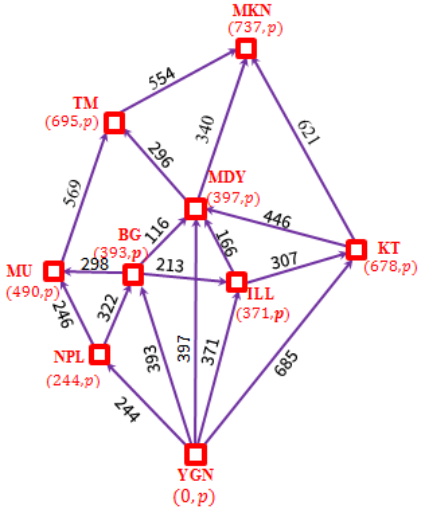

Fig. 9 The optimal path for shortest distance (mile) from Yangon(YGN) to other cities (or places) in Myanmar

The values written and assigned under each city with bracket shown in graph Fig. 9 are the shortest distances to travel from Yangon to respective cities. Likewise, we can start any cities (or place) and by applying the same algorithm to travel any other cities in effective way.

TABLE II
PROGRESSION OF THE SELECTED ROADS AND RESPECTIVE DISTANCES

City	YGN	NPL	MU	BG	ILL	MDY	KT	MKN	TM
Distance (mile)	0	∞	∞	∞	∞	∞	∞	∞	∞
		244	∞	393	371	397	685	∞	∞
			490	393	371	397	685	∞	∞
			490		371	397	685	∞	∞
			490			397	678	∞	∞
			490				678	737	693
			490					737	693
			490						693
			490						
Pred. (city)	NIL	YGN	NPL	YGN	YGN	YGN	ILL	MDY	MDY
Shortest Paths		YGN NPL	YGN NPL MU	YGN BG	YGN ILL	YGN MDY	YGN ILL KT	YGN MDY MKN	YGN MDY TM

We have been calculated the shortest distance from Yangon to other top eight famous destinations using the actual distances.Now minimum travelling time from Yangon to otherdestinationswill be computed using same procedure as above with actual time between two cities as weights and described on each road (edges) [13].

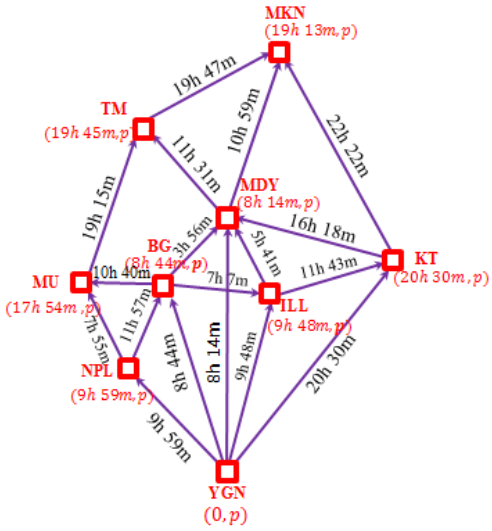

Fig. 10 The graph showing the shortest paths from YGN to other destinations using actual time as weights

TABLE III

PROGRESSION OF THE SELECTED ROADS AND RESPECTIVE MINIMUM TRAVELLING TIME

Cities	YGN	NPL	MU	BG	ILL	KT	MDY	TM	MKN
Time (hour, minutes)	0	∞ 9h 59m 9h 59m 9h 59m 9h 59m	∞ ∞ ∞ 19h 24m 17h 54m 17h 54m	∞ 8h 44m 8h 44m	∞ 9h 48m 9h 48m 9h 48m	∞ 20h 30m 20h 30m 20h 30m 20h 30m 20h 30m 20h 30m 20h 30m	∞ 8h 14m	∞ ∞ 19h 45m 19h 45m 19h 45m 19h 45m 19h 45m 19h 45m	∞ ∞ 19h 13m 19h 13m 19h 13m 19h 13m 19h 13m
Pred. (city)	NIL	YGN	NPL	YGN	YGN	YGN	YGN	MDY	MDY
Shortest Paths		YGN NPL	YGN NPL MU	YGN BG	YGN ILL	YGN KT	YGN MDY	YGN MDY TM	YGN MDY MKN

TABLE IV

COMPARISON FOR THE RESULT IN TABLE IV WITH ACTUAL DRIVING TIME FROM GOOGLE MAP

Start city	Destination cities	Driving time (hour: minute) (Dijkstra)	Driving time(hour: minute) (GoogleMap)	Difference (minutes)
Yangon	Ngapali	9h 59m	10h 07m	- 8
	Mrauk U	17h 54m	16h 33m	+81
	Bagan	8h 44m	8h 55m	-11
	Inle Lake	9h 48m	10h 08m	-20
	Keng Tung	21h 03m	21h 48m	-45
	Mandalay	8h 14m	8h 16m	-2
	Tamu	19h 45m	20h 02m	-17
	Myitkyina	19h 13m	19h 46m	-33

B. Comparison and Recommendation

There are many route search applications to find the shortest paths,among them the most popular algorithms are Dijkstra's algorithm, Bellman Ford algorithm, and Floyd Warshall algorithm. Although they all are algorithms to find shortest paths, the special features of them are different. Dijkstra's algorithm can be applied to find the shortest paths between any two nodes in a graph, whileBellman Ford algorithm and Floyd Warshall algorithm can be used for the networks of small number of nodes and edges respectively.The comparison for calculation times (time complexity)can be seen in the following table [14].

TABLE VI

COMPARISON FOR CALCULATION TIMES OF SHORTEST PATH ALGORITHMS

Shortest-path Algorithm	Time Complexity*
Dijkstra	$n^2 + m$
Bellman Ford	n^3
Floyd Warshall	nm

* *n* and *m* are number of nodes and edges respectively.

Dijkstra's algorithm is the most efficient one to find the shortest paths. It can be applied to both directed and undirected graphs, and calculation time is considerably faster than other algorithms. For these reasons and above comparison,we strongly recommendthat Dijkstra's algorithm is an algorithm to get the best immediate solution for finding shortest paths.

IV. CONCLUSIONS AND FUTURE WORKS

In this work, Dijkstra's Single-Source Shortest Path Algorithm is used to get the optimal results using actual distances, and time between any two nodes (cities). Detail descriptions and step by step procedure of the algorithm has been presented with a flowchart andillustrated by determining the shortest paths started from Yangon to other famous destinations in Myanmar using actual weighted values between any two cities. For further studies, this algorithm can also be applied to find the optimal results for traffic control, path finding in social networks, computer games, transportation systems, and operations research etc. Moreover, based on the flowchart and detail calculation procedure, one can create computer codes such as C/C++, or JAVA, running these codes by using various weighted valuesfromactualinformation and data to solve general Dijkstra's shortest-path problems.

ACKNOWLEDGEMENTS

The author would like to acknowledge her thank to Dr.Khin Saw Lin, Professor and Head, Department of Student Affairs, University of Information Technology, Yangon andDr. Lin Lin Naing, Professor and Head, Faculty of Computing, University of Computer Studies, Hinthada,Ayeyarwady region for their invaluable comments and perfect supervision throughout her research.

REFERENCES

[1] 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, Khavaran Higher-education Institute, Mashhad, Iran. May 12-14.
[3] MyintThanKyi, Lin Lin Naing, "Application of Ford-Fulkerson Algorithm to Maximum Flow in Water Distribution Pipeline Network", International Journal of Scientific and Research Publications (IJSRP), Vol. 8, Issue 12, pp-306-310, December, 2018.[Online]. Available: http://www.ijsrp.org/research-paper-1218/ijsrp-p8441.pdf
[4] P. Sharma, A.Planiya, "Shortest Path Finding of Wireless Optical Network using Dijkstra Algorithm and Analysis of Delay and Blocking Probability",International Journal of Research and Scientific Innovation (IJRSI).Vol. 3, Issue 4, April 2016.
[5] Solving Shortest Path Problem: Dijkstra'sAlgorithm, Lecture Note on Operations Research Methods. October 23, 2009. [Online]. Available: https://www.ifp.illinois.edu/~angelia/ge330fall09_ dijk- stra_l18.pdf
[6] (2012) Myanmar Map, Map No. 4168 Rev. 3 United Nation.[Online]: Available: http://www.un.org/Depts/Cartographic/map/profile/myanmar.pdf
[7] M. Muthulakshm, M.M. Shanmugapriya, "Shortest Path Algorithm and its implementation", International Journal of Mathematics Trends and Technology (IJMTT) – Vol. 36 No. 2- August 2016.
[8] (2019) Dijkstra'sAlgorothmWikipedia website.[Online]. Available: https://en.mwikipedia.org/ wiki/Dijkstar% 27s_algorithm
[9] Vladimir V. Riabov, "Exploring Algorithms for Effective Applicationof the Graph Theory".26th International Conference on Technology in Collegiate MathematicsRivier University, USA.
[10] A.Madkour, A Survey of Shortest-Path Algorithms, USA. 2017.[Online]. Available: https://arxiv.org/pdf/1705.02044.pdf
[11] Vi Ngoc-Nha Tran, Soufiene Djahel,John Murphy,"A comparative study of vehicles' routing algorithms for route planning in smartcities", Conference Paper · November 2012.[Online]. Available: https://www.researchgat e.net/figure/Flowchart-illustrating-the-best-route-update-during-the -vehicles-journey_fig3_271458021
[12] Dijkstra's Shortest Path Algorithm, CLRS 24.3, Lecture note: [Online]. Available: https://www.cse.ust.hk/~dekai/271/notes/L10.
[13] (2018) Myanmar Distance Calculator website. [Online]. Available: https://distance calculator.globefeed.com/Myanmar_Distance_Calculator.asp
[14] Zafar Ali, "Comparison of Dijkstra's Algorithm with other proposed algorithms",International Academic Journal of Science and Engineering. Vol. 3, No. 7, 2016, pp. 53-66.